

**Carlos Eduardo de Andrade**  
**Flávio Lúcio Nogueira Batista**  
**Rodrigo Franco Toso**

**Modelo de Otimização para Transporte de Cargas em Ambientes Reduzidos**

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências para obtenção do título de Bacharel em Ciência da Computação.

Orientador  
Ricardo Martins de Abreu Silva

Lavras  
Minas Gerais - Brasil  
2004



**Carlos Eduardo de Andrade**  
**Flávio Lúcio Nogueira Batista**  
**Rodrigo Franco Toso**

**Modelo de Otimização para Transporte de Cargas em Ambientes Reduzidos**

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências para obtenção do título de Bacharel em Ciência da Computação.

*Aprovada em 24 de Junho de 2004*

---

Guilherme Bastos Alvarenga

---

Ricardo Martins de Abreu Silva  
(Orientador)

Lavras  
Minas Gerais - Brasil



## **Agradecimentos**

Gostaríamos de agradecer nossas namoradas Amanda, Paula e Rafaela, pela paciência com a distância e pelo amor concedido - tenham certeza que o recíproco é verdadeiro.

Agradecemos nosso professor, orientador e amigo Ricardo Martins de Abreu Silva, pela atenção e confiança depositada; e também aos professores do Departamento de Ciência da Computação da Universidade Federal de Lavras pelo curso proporcionado: seremos eternamente gratos pela oportunidade.

Aos nossos irmãos Bruno e Isis; Fabrina e Marcos Paulo; e Ana Silvia.

Agradecemos também nossos amigos por todos os momentos inesquecíveis que vivemos ao longo destes 4 anos de caminhada. Sem vocês o curso não teria a menor graça!

A Deus, pela plenitude da vida; pela maravilha do mundo.



*Dedicamos este trabalho aos nossos pais José Carlos e Nilza; Marcos e Janete;  
Rubens e Ana Maria, que nos proporcionaram a oportunidade de hoje estarmos  
aqui finalizando este curso.*





*A vida é um problema de otimização: um erro nem sempre traz o pior...*  
*...mas sim as escolhas que fazemos depois do erro!*  
**(Os autores)**



## **Resumo**

### **Modelo de Otimização para Transporte de Cargas em Ambientes Reduzidos**

O transporte de cargas é um problema muito custoso nos dias atuais. A dificuldade em se rotear uma frota de veículos para atender determinadas demandas dentro de um ambiente geograficamente reduzido tem incentivado um grande número de pesquisas a fim de otimizar estes tráfegos. Este trabalho tem por objetivo prover uma solução viável, de custo ótimo ou próximo a este, para o problema de coletas e entregas nestes ambientes reduzidos, tais como armazéns e minas de extração de minério. Para tal, o modelo apresentado utiliza um algoritmo baseado na heurística de colônias de formigas a fim de reduzir o tempo de fila nas extremidades do sistema.

**Palavras-chave:** problema de coletas e entregas, modelo de otimização, otimização por colônia de formigas.

## **Abstract**

### **Optimization Model for Transportation in Reduced Environments**

Nowadays, the transportation is a very expensive problem. The complexity in routing a fleet of vehicles to attend determined demands within a geographically reduced environment has incentivated a great number of researches in order to optimize these traffics. This work has for objective to find a feasible solution, with optimum cost or near this, to pickup and delivery problem in these environments, like warehouses and mines of ore extraction. For such, the model presented uses an algorithm based in the metaheuristic of ant colony optimization in order to reduce the queue time in the system extremities.

**Keywords:** pickup and delivery problem, optimization model, ant colony optimization.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Estado da Arte</b>	<b>5</b>
2.1	Otimização . . . . .	5
2.1.1	Definição . . . . .	5
2.2	Problema Geral de Coletas e Entregas . . . . .	6
2.2.1	Formulação . . . . .	8
2.3	Problema de Coletas e Entregas . . . . .	11
2.3.1	Notações . . . . .	12
2.3.2	Formulação do Problema . . . . .	12
2.3.3	Variantes do Problema de Coletas e Entregas . . . . .	13
2.4	Problema de Roteamento de Veículos . . . . .	14
2.5	Problema de Dar uma Volta . . . . .	15
2.6	Soluções . . . . .	16
2.6.1	Métodos Exatos . . . . .	17
2.6.2	Métodos Aproximativos . . . . .	18
2.6.3	Métodos Heurísticos . . . . .	19
<b>3</b>	<b>Otimização por Colônia de Formigas</b>	<b>21</b>
3.1	Preâmbulo: Agentes Inteligentes . . . . .	21
3.2	TSP por Ant System: a primeira implementação . . . . .	22
3.3	A Metaheurística ACO . . . . .	24
3.3.1	Introdução . . . . .	24
3.3.2	Diferenças e Semelhanças entre Formigas Reais e Artificiais	24

3.3.3	ACO: Detalhes e Peculiaridades . . . . .	27
<b>4</b>	<b>Proposta</b>	<b>31</b>
4.1	Descrição do problema . . . . .	31
4.1.1	Dimensões do Problema . . . . .	32
4.1.2	Parâmetros do Problema . . . . .	33
4.1.3	Variáveis de Decisão . . . . .	37
4.1.4	Restrições . . . . .	37
4.1.5	Função Objetivo . . . . .	38
4.2	Modelagem Baseada em Colônia de Formigas . . . . .	39
4.2.1	Relacionamento entre entidades do modelo . . . . .	39
4.2.2	Funcionamento do modelo . . . . .	40
4.2.3	Algoritmo . . . . .	41
4.3	Implementação . . . . .	49
4.3.1	Arquitetura do Software . . . . .	51
4.3.2	Tecnologia Utilizada . . . . .	53
<b>5</b>	<b>Resultados e Discussão</b>	<b>55</b>
5.1	Instâncias Propostas . . . . .	56
5.2	Gráficos e Análise dos Resultados Obtidos . . . . .	58
5.2.1	Apresentação dos Resultados . . . . .	58
5.3	Análise dos Resultados . . . . .	61
<b>6</b>	<b>Trabalhos Futuros</b>	<b>67</b>
<b>A</b>	<b>Feromônios</b>	<b>73</b>
<b>B</b>	<b>Algoritmo para a Metaheurística ACO</b>	<b>77</b>
<b>C</b>	<b>Instâncias</b>	<b>79</b>
C.1	Descrição do formato do arquivo das instâncias . . . . .	79
C.2	Instância 1 . . . . .	80
C.3	Instância 2 . . . . .	83
C.4	Instância 3 . . . . .	85
C.5	Instância 4 . . . . .	88

C.6	Instância 5 . . . . .	90
C.7	Instância 6 . . . . .	93
C.8	Instância 7 . . . . .	95





# Lista de Figuras

2.1	Problema Euclideano Unidimensional . . . . .	7
2.2	Exemplo de um Problema Dinâmico . . . . .	14
3.1	Experimento realizado para uma bifurcação no caminho das formi- gas em busca do alimento . . . . .	25
3.2	Valores obtidos em relação ao percentual de escolha entre o cami- nho superior e inferior . . . . .	26
4.1	Ciclos de Carga e Descarga . . . . .	36
4.2	Diagrama de Colaboração do Algoritmo . . . . .	43
4.3	Arquitetura Básica . . . . .	52
5.1	Desvio percentual das médias do tempo de fila para execução de- terminística . . . . .	59
5.2	Desvio percentual das médias da produção para execução deter- minística . . . . .	60
5.3	Desvio percentual das médias do tempo de fila para execução es- tocástica . . . . .	61
5.4	Desvio percentual das médias da produção para execução estocástica	62
5.5	Desvios de produção nas simulações estáticas, determinísticas e estocásticas. . . . .	63
5.6	Comparação dos tempos de fila nas instâncias ímpares. . . . .	64
5.7	Comparação dos tempos de fila nas instâncias pares. . . . .	65



# Lista de Tabelas

5.1	Descrição das Instâncias . . . . .	57
5.2	Blendagens das Instâncias 1 e 2 . . . . .	57
5.3	Blendagens das Instâncias 3 e 4 . . . . .	57
5.4	Blendagens das Instâncias 5 e 6 . . . . .	57
5.5	Blendagens da Instância 7 . . . . .	58
5.6	Tempos de fila e Desvios de Produção obtidos na simulação . . .	63

# Capítulo 1

## Introdução

Nos últimos anos, importantes trabalhos vêm sendo publicados no campo de Inteligência Computacional, especificamente na área de Heurísticas Naturais Evolutivas, uma área que procura se basear em sistemas naturais ou sociais para gerar métodos de solução não-determinísticos para problemas complexos (NP-difíceis) de otimização combinatória.

No presente contexto, surgiu em 1999 a metaheurística de Otimização por Colônia de Formigas (*Ant Colony Optimization*, ACO) ([DORIGO, 1999]), como uma formalização para aplicações já utilizadas em vários problemas combinatoriais complexos: Problema do Caixeiro Viajante (*Traveling Salesman Person*, TSP) ([DORIGO, 1991]), Problema de Atribuição Quadrática (*Quadratic Assignment Problem*) ([MANIEZZO, 1994]), Problema de Escalonamento (Job-shop) (*Job-shop Scheduling*) ([COLORNI, 1994]), Problema de Arranjo Sequencial (*Sequential Ordering Problem*) ([GAMBARDELLA, 1997]), Problema de Roteamento de Veículos (*Vehicle Routing Problem*, VRP) ([BULLNHEIMER, 1999]), dentre outros.

A metaheurística ACO derivou-se do trabalho de Dorigo ([DORIGO, 1991]) para sua tese de doutorado, onde foi proposto o estudo de sistemas artificiais baseados em colônia de formigas reais, método então denominado *Ant System* (uma heurística pré-ACO), com intuito de solucionar problemas de otimização combinatória, especificamente o problema do caixeiro viajante. A idéia desta metaheurística consiste na simulação dos processos de depósito e evaporação de fero-

mônio, substância excretada pelas formigas reais como mecanismo de aprendizagem de um sistema multiagente (formigas), proposta a fim de solucionar problemas combinatoriais.

Contudo, estudos recentes [SILVA, 2003] constataram que, na maior parte dos artigos publicados pela comunidade ACO, a análise experimental das heurísticas de otimização não seguia um método de avaliação, principalmente no que diz respeito à carência de estudos sobre a influência dos parâmetros utilizados pelas heurísticas na qualidade dos resultados observados. De maneira geral, os trabalhos se preocupavam mais com a identificação da melhor heurística para determinado problema de otimização que a verificação experimental de como as heurísticas realmente trabalham.

Uma vez feitos estudos de viabilidade em relação a metaheurística ACO, incluindo uma análise detalhada de suas potencialidades e seus pontos fracos, onde foi concluído que a metaheurística de colônia de formigas não possui comportamento eficiente em problemas envolvendo grafos com elevado número de conexões (grafos totalmente conectados, por exemplo) ([SILVA, 2003]), um estudo aplicado da metaheurística em seu ambiente de origem (problemas envolvendo rotas), não envolvendo grafos totalmente conectados, faz-se necessário.

Desta forma, o objetivo deste trabalho consiste em aplicar a metaheurística ACO para a resolução de um problema combinatorial que tenha foco na montagem de rotas para veículos, com análise de adaptabilidade da metaheurística ao problema abordado e enfoque analítico em função de seus parâmetros de otimização.

O problema de otimização combinatória envolvendo rotas adotado neste trabalho foi uma especialização do problema de coletas e entregas (*Pickup and Delivery Problem*, PDP). O PDP, classificado como sendo NP-difícil segundo Savelsbergh [SAVELSBERGH, 1985], foi modificado de acordo com as restrições observadas em ambientes reduzidos, a partir de um estudo de caso em uma mineradora de grande porte situada na capital mineira. Por ser um problema de grande complexidade, a expectativa de um desempenho satisfatório para algoritmos exatos é extremamente dificultada de acordo com a quantidade de entradas.

Foi utilizada a metaheurística ACO simulando as funções de roteamento de uma colônia de formigas naturais. Para isto, cada formiga possui conhecimento do ambiente (provido pela quantidade de feromônio nos possíveis caminhos) e também da situação de fila nos pontos de entroncamento do problema abordado (neste problema representados por pontos de carga e pontos de descarga).

Um elemento global de otimização, denominado *daemon* [DORIGO, 1999], foi modelado para dotar as formigas de uma visão mais abrangente do ambiente, fato que não ocorre apenas com o depósito de feromônio; e ainda ajuda na varredura do espaço de busca das soluções - principalmente no tocante à fuga de ótimos locais.

Um ponto fraco do método proposto está nas instâncias utilizadas em teste, que, mesmo criadas com rigor matemático, não são instâncias reais do problema. Cabe ressaltar que as mesmas foram concebidas com intuito de levar à exaustão os testes realizados, e se aproximam daquilo observado nas mineradoras supracitadas.

É importante deixar claro que o foco desta monografia não está na criação de novas heurísticas para resolução de um problema clássico da literatura, fugindo então do escopo de análise e comparação desta com relação aos métodos tradicionais de busca por soluções para, por fim, analisar o comportamento da ACO em situações que a remetem as suas origens.

A principal contribuição deste trabalho encontra-se na revelação do quão importante é a utilização de valores que destaquem a saliência do feromônio na tomada de decisões pelas formigas, mostrando o comportamento convergente da metaheurística ACO para instâncias relacionadas a grafos médio-conectados (possuem ligações o bastante para serem combinatórios e ligações o bastante para serem totalmente conectados).

Uma contribuição secundária está na capacidade de aplicação aos casos especiais do problema de coletas e entregas, como por exemplo na logística em mineradoras. Os resultados obtidos pela metaheurística revelaram um bom grau de convergência, quando comparados a formas tradicionais de roteamento (roteamento estático) e métodos de resolução determinísticos (algoritmo guloso, por exemplo).

A organização deste documento apresenta, no segundo capítulo, o Problema Geral de Coletas e Entregas (*General Pickup and Delivery Problem*, GPDP), que é

o conjunto de todos os problemas que envolvem o roteamento de veículos, e também o estado da arte relativo aos métodos de abordagem do problema-alvo deste trabalho. O terceiro capítulo detalha a metaheurística de otimização por colônia de formigas. O quarto capítulo contém a proposta de resolução apresentada, inclusive o modelo de otimização proposto para o problema abordado. O quinto capítulo apresenta os resultados obtidos e as conclusões deste trabalho. Por fim, são sugeridos alguns trabalhos futuros no capítulo seis.

## Capítulo 2

# Estado da Arte

### 2.1 Otimização

#### 2.1.1 Definição

Uma instância de um problema de otimização, segundo [PAPADIMITRIOU, 1982], consiste no par  $(F, c)$ , onde  $F$  é um conjunto qualquer, constituído pelos pontos viáveis, e  $c$  é uma função de custo, um mapeamento

$$c : F \mapsto R^1 \quad (2.1)$$

O problema consiste em encontrar um  $f \in F$  para cada

$$c(f) \leq c(y) \quad \forall \quad y \in F \quad (2.2)$$

Cada ponto  $f$  é denominado ótimo global para a instância, e é comumente referenciado por ótimo.

Desta forma, um *Problema de Otimização* estende-se ao conjunto  $I$  de instâncias de um problema de otimização. Os principais constituintes de um problema de otimização são:

- Vizinhança: dado um ponto viável  $f \in F$  num determinado problema com instâncias  $(F, c)$ , sua vizinhança consiste no mapeamento

$$N : F \mapsto 2^F \quad (2.3)$$



definido para cada instância.

- Ótimo Local e Global: em certas instâncias de problemas, encontrar uma solução ótima pode ser uma tarefa impossível do ponto de vista computacional. Nesses casos, há como encontrar uma solução  $f$ , sendo essa a melhor solução na vizinhança  $N(f)$ .

Em uma instância  $(F, c)$  de um problema de otimização, com vizinhança  $N$ , a solução viável  $f \in F$  é definida como sendo *ótimo local* em relação a  $N$  se

$$c(f) \leq c(g) \quad \forall \quad g \in N(f) \quad (2.4)$$

Na figura 2.1, observando os pontos ótimos assinalados, pode-se ter a clara distinção entre o ótimo local e o ótimo global: os pontos A, B e C são ótimos locais, mas apenas o ponto B é o ótimo global.

No exemplo, utilizado por [PAPADIMITRIOU, 1982], o problema abordado é conhecido por *1-dimensional Euclidean Problem*, ou simplesmente Problema Euclidiano Unidimensional. Considerando a instância  $(F, c)$  onde  $F = [0, 1] \subseteq R^1$  e a função de custos  $c$  traçada na figura 2.1, a vizinhança pode ser definida por

$$N_\epsilon(f) = \{x : x \in F \quad e \quad |x - f| \leq \epsilon\} \quad (2.5)$$

## 2.2 Problema Geral de Coletas e Entregas

Esta seção foi baseada na publicação de Savelsbergh [SAVELSBERGH, 2001]. O Problema Geral de Coletas e Entregas (*General Pickup and Delivery Problem*, (GPDP)) consiste em um conjunto de rotas que deve ser construído a fim de satisfazer pedidos de transporte. Uma frota de veículos é disponibilizada para operar nas rotas. Cada veículo tem uma certa capacidade, um local inicial e um local final. Cada pedido de transporte especifica um tamanho da carga a ser transportada, as localidades onde serão feitas as coletas (as origens) e as localidades onde

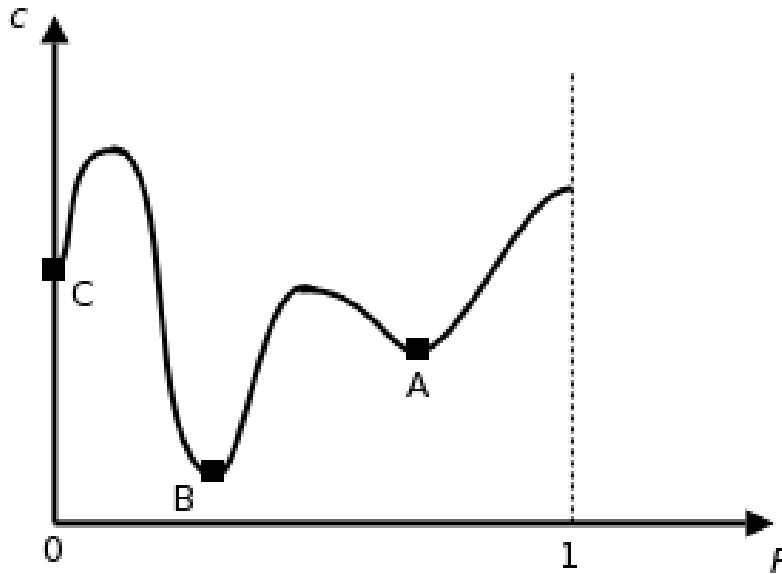


Figura 2.1: Problema Euclidiano Unidimensional

serão feitas as entregas (os destinos). Cada carga deve ser transportada por um veículo do seu conjunto de origens para o seu conjunto de destinos sem nenhuma baldeação em outras localidades. Existem três problemas de roteamento muito conhecidos e estudados extensivamente e que são casos especiais de GPDP. No Problema de Coletas e Entregas (*Pickup and Delivery Problem*, PDP), cada pedido de transporte especifica uma simples origem e um simples destino e todos os veículos partem de e retornam ao depósito central. O Problema de Dar uma Volta (*Dial-a-Ride Problem*, DARP) é um PDP em que as cargas a serem transportadas são pessoas. Consequentemente, chama-se de clientes ao invés de pedidos de transporte e todos os tamanhos de carga são iguais a um. O Problema de Roteamento de Veículos (*Vehicle Routing Problem*, VRP) é um PDP em que qualquer uma das origens ou dos destinos são localizadas no depósito.

O GPDP tem como finalidade ser capaz de lidar com várias características complicadas encontradas em muitos problemas de coleta e entrega, como pedidos de transporte especificando um conjunto de origens associadas a um único destino ou uma única origem associada com um conjunto de destinos, veículos com diferentes locais iniciais e finais, e pedidos de transporte evoluindo em tempo real.

Muitas situações práticas de coleta e entrega são sensíveis, isto é, novos pedidos de transporte tornam-se disponíveis em tempo real e ficam imediatamente aptos para serem considerados. Como consequência, o conjunto de rotas tem que ser reotimizado em algum ponto para incluir os novos pedidos de transporte. Sendo assim, é importante ressaltar que no momento da reotimização, veículos estão em suas rotas trabalhando e não existe a noção de depósito.

### 2.2.1 Formulação

Para uma melhor compreensão de como funciona o GPDP, este tópico aborda a formulação matemática do mesmo. Sendo assim,  $N$  é o conjunto de pedidos de transporte. Para cada pedido de transporte  $i \in N$ , um tamanho de carga  $q_i$  tem que ser transportado do conjunto de origens  $N_i^+$  para o conjunto de destinos  $N_i^-$ . Cada carga é subdividida em:  $q_i = \sum_{j \in N_i^+} q_j = - \sum_{j \in N_i^-} q_j$ , isto é, quantidades positivas para coletas e negativas para entregas. Define-se  $\mathcal{N}^+ = \cup i \in N \mathcal{N}_i^+$  como o conjunto de todas as origens e  $\mathcal{N}^- = \cup i \in N \mathcal{N}_i^-$  como o conjunto de todos os destinos. Tem-se, também,  $\mathcal{V} = \mathcal{N}^+ \cup \mathcal{N}^-$ . Além disso, tem-se que  $\mathcal{M}$  é o conjunto de veículos. Cada veículo  $k \in \mathcal{M}$  tem uma capacidade  $Q_k$ , um local inicial  $k^+$ , e um local final  $k^-$ . Define-se  $\mathcal{M}^+ = k^+ | k \in \mathcal{M}$  como o conjunto de locais iniciais e  $\mathcal{M}^- = k^- | k \in \mathcal{M}$  como o conjunto de locais finais. Tem-se, todavia que,  $\mathcal{W} = \mathcal{M}^+ \cup \mathcal{M}^-$ . E para todo  $i, j \in \mathcal{V} \cup \mathcal{W}$ , tem-se que  $d_{i,j}$ ,  $t_{i,j}$ ,  $c_{i,j}$  denotam, respectivamente, a distância, o tempo e o custo da viagem.

- Uma rota de coleta e entrega  $R_k$  para um veículo  $k$  é uma rota direcionada através de um subconjunto  $V_k \subset V$  tal como:
  1.  $R_k$  começa em  $k^+$ ;
  2.  $(\mathcal{N}_i^+ \cup \mathcal{N}_i^-) \cap V_k = \emptyset$  ou  $(\mathcal{N}_i^+ \cup \mathcal{N}_i^-) \cap V_k = \mathcal{N}_i^+ \cup \mathcal{N}_i^-$  para todo  $i \in N$ ;
  3. Se  $\mathcal{N}_i^+ \cup \mathcal{N}_i^- \subseteq V_k$ , então todas as localidades  $\mathcal{N}_i^+$  são visitadas antes das localidades em  $\mathcal{N}_i^-$ ;
  4. O veículo  $k$  visita cada localidade em  $V_k$  exatamente uma vez;
  5. A carga do veículo nunca excede  $Q_k$ ;

6.  $R_k$  termina em  $k_-$ .

- Um plano de coleta e entrega é um conjunto de rotas  $R = R_k | k \in \mathcal{M}$  tal como:

1.  $R_k$  é uma rota de coleta e entrega para um veículo  $k$ , para cada  $k \in \mathcal{M}$ ;
2.  $V_k | k \in \mathcal{M}$  é uma partição de  $\mathcal{V}$ .

Define-se  $f(R)$  como o preço do plano  $R$  correspondente a uma certa função objetivo  $f$ . Desta maneira, pode-se definir o problema geral de coletas e entregas como um problema:

$$\min \{f(R) \mid R \text{ é um plano de coleta e entrega}\}$$

Os casos especiais de GPDP mencionados nesta seção podem ser caracterizados da seguinte forma:

- PDP:  $|\mathcal{W}| = 1$  e  $|\mathcal{N}^+_i| = |\mathcal{N}^-_i| = 1$  para todo  $i \in \mathcal{N}$ . Neste caso define-se  $i^+$  como um elemento único de  $\mathcal{N}^+_i$  e  $i^-$  como um único elemento de  $\mathcal{N}^-_i$ .
- DARP:  $|\mathcal{W}| = 1$  e  $|\mathcal{N}^+_i| = |\mathcal{N}^-_i| = 1$  e  $q_i = 1$  para todo  $i \in \mathcal{N}$ .
- VRP:  $|\mathcal{W}| = 1$  e  $|\mathcal{N}^+_i| = |\mathcal{N}^-_i| = 1$  para todo  $i \in \mathcal{N}$ , e  $\mathcal{N}^+ = \mathcal{W}$  ou  $\mathcal{N}^- = \mathcal{W}$ .

Embora a literatura meramente cubra os problemas de coleta e entrega com  $|\mathcal{N}^+_i| = |\mathcal{N}^-_i| = 1$ , em muitas situações práticas  $|\mathcal{N}^+_i| > 1$  ou  $|\mathcal{N}^-_i| > 1$ . Em algumas destas situações, um pedido de transporte com  $|\mathcal{N}^+_i| > 1$  ou  $|\mathcal{N}^-_i| > 1$  pode ser descomposto em muitos pedidos independentes com um único ponto de coleta e um único ponto de entrega. Entretanto, em muitas outras situações, um pedido com múltiplos pontos de coleta e entrega deve ser atendido por um único veículo e conseqüentemente não pode ser descomposto.

Mesmo não havendo conhecimento de qualquer aplicação real onde pedidos ocorrem com ambos  $|\mathcal{N}^+_i| > 1$  e  $|\mathcal{N}^-_i| > 1$ , este caso é uma parte da definição do GPDP para a causa da simetria.

Para formular o GPDP como um programa matemático, são introduzidas quatro tipos de variáveis:

- $z_i^k(i \in \mathcal{N}, k \in \mathcal{M})$  igual a 1 se o pedido de transporte  $i$  é atribuído ao veículo  $k$  e 0 em caso contrário;
- $x_{i,j}^k((i, j) \in (\mathcal{V} \times \mathcal{V}) \cup (k_+, j) | j \in \mathcal{V} \cup (j, k_-) | j \in \mathcal{V}, k \in \mathcal{M})$  igual a 1 se o veículo  $k$  viaja da localidade  $i$  para a localidade  $j$  e 0 em caso contrário;
- $D_i(i \in \mathcal{V} \cup \mathcal{W})$ , especificando o tempo de partida no vértice  $i$ , e;
- $y_i(i \in \mathcal{V} \cup \mathcal{W})$ , especificando a carga do veículo que chegou ao vértice  $i$ .

Define-se  $q_{k_+} = 0$  para todo  $k \in \mathcal{M}$ . O problema é agora minimizar  $f(x)$ , sujeito a:

$$\sum_{k \in \mathcal{M}} z_i^k = 1 \quad \forall \quad i \in \mathcal{N}; \quad (2.6)$$

$$\sum_{j \in (\mathcal{V} \cup \mathcal{W})} x_{l,j}^k = \sum_{j \in \mathcal{V} \cup \mathcal{W}} x_{j,l}^k = z_i^k \quad \forall \quad i \in \mathcal{N}, l \in (\mathcal{N}^+_i \cup \mathcal{N}^-_i), k \in \mathcal{M}; \quad (2.7)$$

$$\sum_{j \in (\mathcal{V} \cup k_-)} x_{k_+,j}^k = 1 \quad \forall \quad k \in \mathcal{M}; \quad (2.8)$$

$$\sum_{j \in (\mathcal{V} \cup k_+)} x_{i,k_-}^k = 1 \quad \forall \quad k \in \mathcal{M}; \quad (2.9)$$

$$D_{k_+} = 0 \quad \forall \quad k \in \mathcal{M}; \quad (2.10)$$

$$D_p \leq D_q \quad \forall \quad i \in \mathcal{N}, p \in \mathcal{N}^+_i, q \in \mathcal{N}^-_i; \quad (2.11)$$

$$x_{i,j}^k = 1 \Rightarrow D_i + t_{i,j} \leq D_j \quad \forall \quad i, j \in (\mathcal{V} \cup \mathcal{W}), k \in \mathcal{M}; \quad (2.12)$$

$$y_{k_+} = 0 \quad \forall \quad k \in \mathcal{M}; \quad (2.13)$$

$$y_l \leq \sum_{k \in \mathcal{M}} Q_k z_i^k \quad \forall \quad i \in \mathcal{N}, l \in (\mathcal{N}^+_i \cup \mathcal{N}^-_i); \quad (2.14)$$

$$x_{i,j}^k = 1 \Rightarrow y_i + q_i = y_j \quad \forall \quad i, j \in (\mathcal{V} \cup \mathcal{W}), k \in \mathcal{M}; \quad (2.15)$$

$$x_{i,j}^k \in 0, 1 \quad \forall \quad i, j \in (\mathcal{V} \cup \mathcal{W}), k \in \mathcal{M}; \quad (2.16)$$

$$z_i^k \in 0, 1 \quad \forall \quad i \in \mathcal{N}, k \in \mathcal{M}; \quad (2.17)$$

$$D_i \geq 0 \quad \forall \quad i \in (\mathcal{V} \cup \mathcal{W}); \quad (2.18)$$

$$y_i \geq 0 \quad \forall \quad i \in (\mathcal{V} \cup \mathcal{W}). \quad (2.19)$$

A restrição 2.6 garante que cada pedido de transporte é atribuído para exatamente um veículo. Pela restrição 2.7 somente um veículo entra ou deixa a localidade  $l$  se a mesma é uma origem ou um destino de um pedido de transporte atribuído para o referido veículo. As restrições 2.8 e 2.9 asseguram que cada veículo começa ou termina no lugar correto. As restrições 2.10, 2.11, 2.12 e 2.13 formam as restrições de precedência. E por fim, as restrições 2.14, 2.15, 2.16 e 2.17 compõem as restrições de capacidade.

## 2.3 Problema de Coletas e Entregas

"O problema de coletas e entregas é uma especialização do problema geral de coletas e entregas (referenciado na seção 2.2), definido por uma frota de veículos com o objetivo de atender um conjunto de pedidos de transporte. Cada veículo pertencente à frota possui uma dada capacidade de carga, um ponto inicial e um ponto terminal. No PDP, diferente do GPDP, todos os veículos possuem ponto inicial e final iguais (depósito central). Cada pedido de transporte é especificado por uma carga a ser transportada, uma localidade de origem e uma localidade de destino. Resumidamente, o PDP trata a construção de rotas ótimas que atendam todos os pedidos de transporte, satisfazendo restrições de paridade, precedência e capacidade"[SAVELSBERGH, 2001], [MITROVIC-MINIC, 1998], [MALCA, 2003].

### 2.3.1 Notações

Considere o conjunto de  $n$  requisições (*Req*) denotado por  $O_j$  ( $j \in [1, n]$ ). Para cada requisição  $O_j$ ,  $O_j^+$  consiste no nó de coleta e  $O_j^-$  no nó de entrega,  $q_j$  representa o tamanho da carga a ser transportada,  $d_j^+$  representa o período de duração do serviço de coleta e  $d_j^-$  a duração do serviço de entrega.

A frota possui o tamanho de  $m$  veículos. Para cada veículo  $V^k$  ( $k \in [1, m]$ ), tome  $V_+^k$  como o ponto inicial e  $V_-^k$  como o ponto final da rota  $R^k$  associada ao veículo  $V^k$ ; e  $Q^k$  constituindo a capacidade do veículo  $k$ .

O problema também denota o custo de viagem do arco  $(A, B)$  por  $c(A, B)$  e o tempo de viagem entre  $A$  e  $B$  por  $t(A, B)$ , com

$$\begin{aligned} & A \neq B; \\ & A, B \in \{O_j^+, O_j^-, V_+^k, V_-^k \mid j \in [1, n]; k \in [1, m]\}. \end{aligned}$$

### 2.3.2 Formulação do Problema

O problema de coletas e entregas consiste em traçar a rota  $R^k$  para cada veículo  $V^k$  de tal forma que

- (i) A rota  $R^k$  começa em  $V_+^k$  e termina em  $V_-^k$  se o veículo  $V^k$  atende um ou mais pedidos, caso contrário o mesmo veículo permanece em  $V_+^k$ ;
- (ii) Cada pedido é atendido por exatamente uma rota  $R^k$ . Para cada pedido atendido por uma rota  $R^k$ , ambas origem e destino devem pertencer também à mesma rota  $R^k$ , e seu nó de coleta deve ser sempre visitado antes do nó de entrega;
- (iii) A carga total de cada veículo  $V^k$  nunca deve ser maior que  $Q^k$ ;
- (iv) O número de pedidos atribuídos a cada rota é maximizado e o custo total das rotas é minimizado.

A restrição (ii) configura a precedência da coleta perante a entrega e o obediência da paridade coleta-entrega, ou seja, cada pedido deve ser atendido completamente para que então aconteça o tratamento do próximo. A restrição (iii) é

a relativa à carga transportada. (i) representa uma condição de movimentação, e, por fim, (iv) descreve as funções objetivo.

Alvarenga [ALVARENGA, 2002] afirma que, dada inexistência de um depósito central para coletas e/ou entregas, o problema de coletas e entregas configura um modelo contínuo e dinâmico. Contínuo por existir sequência entre a conclusão de um pedido e o início de outro; dinâmico pela possibilidade de chegada de novos pedidos de atendimento, ou seja, pode não haver o conhecimento prévio de  $O_j$  em sua totalidade -  $O_j$  será montado, segundo a fração de tempo  $t$ , por

$$O_j(t+1) = O_j(t) \cup \sum_{l=1}^p Req^l \quad (2.20)$$

com  $Req^l$  representando uma requisição completa e  $p$  o número de requisições que foram adicionadas no tempo  $t$ . A figura 2.2 demonstra, pelo caminho tracejado, a capacidade de atendimento dinâmico do pedido de transporte 5, que não foi determinado *a priori*.

Já uma instância estática do problema de coletas e entregas fornece, *a priori*, o conjunto  $O_j$  de requisições de transporte, não havendo a possibilidade de novas requisições a partir da fração de tempo  $t = 0$ . Na figura 2.2, não existe a possibilidade de atendimento a uma nova requisição de transporte (5) na instância atual. Para o atendimento de tal requisição, todas as requisições existentes até então devem ser atendidas, ou seja, a instância atual seria resolvida, sendo então criada uma nova instância do problema para o tratamento da requisição 5.

### 2.3.3 Variantes do Problema de Coletas e Entregas

As variantes do problema de coletas e entregas são divididas em problemas estáticos e dinâmicos, com ou sem janelas de tempo. Todas elas apresentam modelagens matemáticas mais elaboradas que fogem ao escopo deste trabalho. Em [SAVELSBERGH, 2001] pode-se encontrar um resumo mais estendido destas variantes.

As três principais variantes são o problema de coletas e entregas com um único veículo (1-PDP), com múltiplos veículos (m-PDP), com carga completa e o problema de dar a volta (*Dial-a-ride Problem*, DARP). O problema com um único



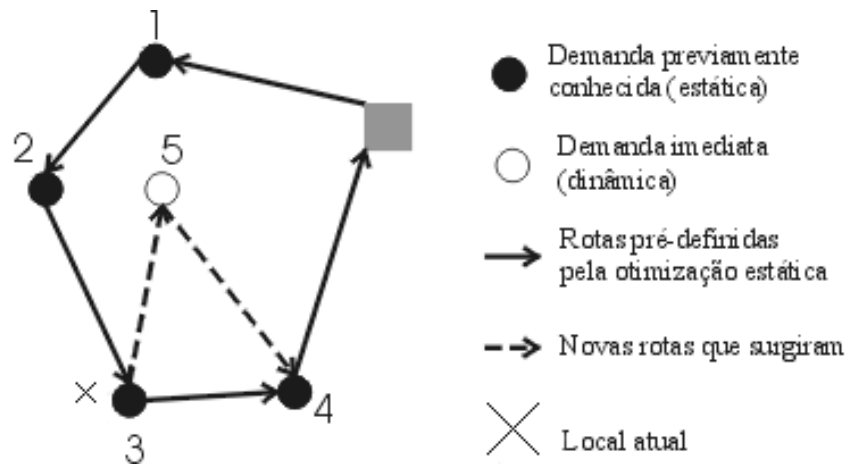


Figura 2.2: Exemplo de um Problema Dinâmico

veículo remete as requisições a um único veículo. É um dos mais estudados pois é um subproblema do de múltiplos veículos. O problema de múltiplos veículos, como já dito, é uma composição do problema com um único veículo e todas complicações a mais que estes mesmos veículos trazem consigo em uma operação em conjunto. O problema com carga completa reflete que um transportador vai direto do ponto de carga para o ponto de descarga, sem baldeações. O problema de dar a volta está especificado na seção 2.5.

## 2.4 Problema de Roteamento de Veículos

O mais conhecido e estudado problema de veículos segundo [LARSEN, 2000] é o problema do caixeiro viajante (*Traveling Salesman Problem*, TSP), que consiste em um vendedor que visita um conjunto de cidades e retorna à cidade onde começou a viagem. O objetivo consiste em minimizar a distância total percorrida. Este problema de programação matemática basicamente consiste na busca de um ciclo hamiltoniano em um grafo, sendo considerado um problema NP difícil segundo [MELO, 2001], ou seja, não existe algoritmo em tempo polinomial determinístico que resolva o problema; apenas algoritmos exponenciais.

O problema de roteamento de veículos, mais conhecido na literatura como VRP (*Vehicle Routing Problem*) é uma generalização do problema do caixeiro vi-

ajante, consistindo em determinar rotas de veículos, onde uma rota é um ciclo que começa em um depósito, visitando um subconjunto de clientes ou pontos de demanda em uma dada ordem e retornando ao depósito. Todos os pontos de demanda devem ser visitados apenas uma vez e o total da demanda dos clientes de uma rota não pode exceder a capacidade do veículo, segundo [LARSEN, 2000].

No mundo real o número de restrições aferidas ao problema é um fator complicante para a modelagem matemática. É perfeitamente aceitável que existam múltiplos pontos de fornecimento, os depósitos ou facilidades, veículos com diversas capacidades, restrições de entrega como tempo ou distância. Todos estes fatores devem ser considerados como essenciais para a correta e fiel modelagem do tipo de problema tratado.

A ênfase dada a cada tipo de restrição vincula os problemas em subclasses como roteamento de veículos com janela de tempo, com restrição de capacidade, entre outros. Em [FISCHER, 1995] encontra-se uma classificação mais detalhada sobre as subclasses supracitadas.

A formulação matemática do VRP também abrange o que foi descrito no problema de coletas e entregas, no entanto, algumas restrições são acrescentadas na mesma. No VRP, ou o veículo faz uma coleta ou ele faz uma entrega, e nunca faz os dois numa mesma viagem, ou seja,  $|\mathcal{W}| = |\mathcal{N}^+_i| = |\mathcal{N}^-_i| = 1 \quad \forall i \in \mathcal{N}$ , onde  $\mathcal{N}^+ = \mathcal{W}$  ou  $\mathcal{N}^- = \mathcal{W}$ . Desta forma, o pedido de transporte que o veículo irá atender só pode estar contido em um dos conjuntos, ou seja, ou no conjunto de origens ou locais de coleta, ou no conjunto de destinos ou locais de entrega.

## 2.5 Problema de Dar uma Volta

Mais conhecido como *Dial-a-Ride Problem* ou DARP, trata-se de uma subclasse do PDP na qual a carga a ser transportada equivale a pessoas. Na realidade, o DARP pode ser comparado a um táxi. Desta forma, o pedido de transporte parte de uma pessoa localizada em local qualquer de uma cidade. No entanto, o taxista parte de outro local que não precisa ser necessariamente o local onde está a pessoa que requisitou o pedido. E o taxista somente parte para o local de destino quando estiver com a pessoa, ou seja, quando estiver carregado.

Uma das principais restrições do DARP diz que depois de carregado, o veículo somente pode coletar outra carga (no caso, pessoa) depois que tiver descarregado. Assim, o local de destino do veículo é sempre estabelecido quando o veículo é carregado (quando alguma pessoa entra no carro) no local de origem.

Segundo Savelsbergh ([SAVELSBERGH, 2001]), a formulação matemática do DARP abrange um pouco do que foi descrito pelo PDP, entretanto algumas restrições são adicionadas. No DARP, a capacidade do veículo equivale a uma unidade, ou seja

$$|\mathcal{W}| = |\mathcal{N}^+_i| = |\mathcal{N}^-_i| = 1 \quad \forall \quad i \in \mathcal{N} \quad (2.21)$$

e

$$q_i = 1 \quad \forall \quad i \in \mathcal{N} \quad (2.22)$$

## 2.6 Soluções

Nem sempre é trivial a busca por soluções em problemas de otimização. Conforme observado anteriormente, alguns problemas não podem ser resolvidos por um computador em tempo hábil, outros podem requerer mais memória que aquela disponível em um computador atual. Também existem aqueles que não possuem solução exata conhecida, ou seja, ainda não foram resolvidos.

Segundo [CORMEN et al., 2001], os algoritmos podem ser classificados em relação a complexidade como:

- **P** (*Deterministic Polynomial Time*): são algoritmos cujo tempo de execução no pior caso não ultrapassa o teto de  $O(n^k)$ , para alguma constante  $k$ , onde  $n$  consiste no número de entradas do problema; ou
- **NP** (*Nondeterministic Polynomial Time*): são algoritmos que possuem tempo de execução com teto de  $O(k^n)$ , para alguma constante  $k$ , com  $n$  constituindo o número de entradas do problema.

Certamente os problemas NP possuem custo muito maior em relação a problemas da classe P, para os mesmos valores de  $n$  e  $k$ . O caráter exponencial de complexidade - NP - está presente em todos os problemas considerados intratáveis.

Os métodos para resolução de problemas estão diretamente ligados à viabilidade da busca pela solução ótima. Tal viabilidade está relacionada ao número de entradas para o mesmo, e, dependendo deste é que se escolhe um ou outro método de resolução, que são classificados como exatos (seção 2.6.1), aproximativos (seção 2.6.2) e heurísticos (seção 2.6.3). O caráter de NP-completude não é trivial, possuindo linguagens e métodos peculiares para sua distinção perante problemas polinomiais. Desta forma, este trabalho não os aborda com grande nível de detalhes, estando estes disponíveis em [CORMEN et al., 2001] e também em [PAPADIMITRIOU, 1982].

### **2.6.1 Métodos Exatos**

Os métodos de busca por soluções denominados exatos são aqueles que sempre encontram a melhor solução para o problema (se é que esta existe), ou seja, a solução ótima. Esta solução deve satisfazer de forma ótima a função objetivo correspondente ao problema em questão, respeitando todos os parâmetros que se aplicam à resolução do mesmo.

Segundo Cormen ([CORMEN et al., 2001]), um problema pode ou não possuir um algoritmo exato para sua solução. Existindo esse algoritmo, o mesmo pode não encontrar um ótimo em tempo hábil, ou seja, o algoritmo pode levar décadas para encontrar a solução desejada. Quando isso ocorre, diz-se que o mesmo é inviável para a instância abordada.

Com isto, mais uma questão deve ser analisada na escolha da forma exata de resolução dos problemas: a viabilidade do método escolhido. Como pode-se observar, nem sempre um método exato consiste na melhor forma de abordagem para um problema. Aprofundando ainda mais a questão de viabilidade, tanto algoritmos da classe P quanto algoritmos da classe NP podem ser inviáveis: tomando por exemplo um algoritmo exato com teto  $O(n^{100})$ , mesmo pertencendo à classe P este algoritmo pode não viável computacionalmente para, por exemplo,  $n = 10000$  entradas.

Especificamente, os problemas classificados como polinomiais - pertencentes à classe de problemas P - são em sua maioria viáveis para aplicação de métodos exatos para resolução.

### 2.6.2 Métodos Aproximativos

Cormen ([CORMEN et al., 2001]) denota os métodos aproximativos como uma possibilidade viável de se encontrar soluções para problemas exponenciais onde os métodos exatos não se aplicam computacionalmente. Os algoritmos de aproximação, por possuírem tempo polinomial, podem resolver problemas exponenciais com uma boa margem de aproximação à solução ótima.

Os métodos aproximativos possuem a característica de buscar soluções com desempenho consideravelmente bom, em tempo hábil e viáveis para a maioria dos problemas onde são utilizados. Segundo Cormen ([CORMEN et al., 2001]), os algoritmos de aproximação sempre se aproximam da solução ótima, com qualidade de solução dependente do tempo de execução. A relação de aproximação de um algoritmo é  $\rho(n)$  se, para qualquer entrada de tamanho  $n$ , o custo  $C$  da solução produzida pelo algoritmo de aproximação está dentro de um fator  $\rho(n)$  do custo  $C^*$  de uma solução ótima:

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n) \quad (2.23)$$

Vários problemas clássicos da literatura possuem soluções aproximativas como alternativa à vasta demanda computacional requerida por métodos exatos aplicados a problemas NP. Alguns deles, segundo Cormen [CORMEN et al., 2001], são:

- Problema de Cobertura de Vértices (*Vertex Cover Problem*);
- Problema do Caixeiro Viajante (*Travelling Salesman Problem*);
- Problema da Cobertura de Conjuntos (*Set Covering Problem*);
- Problema da Soma de Subconjuntos (*Subset Sum Problem*);

### 2.6.3 Métodos Heurísticos

Métodos heurísticos compõem uma gama relativamente nova de soluções para problemas de otimização combinatória. Tais métodos possuem origens distintas, geralmente utilizadas especificamente para determinado problema combinatório. Quando um método é aplicado especificamente no problema, sob a forma de um algoritmo, este é denominado um método heurístico, ou simplesmente uma heurística. Existe também na literatura o termo Metaheurística. A distinção entre tais denominações faz-se por:

- Metaheurística: possui grande abrangência, podendo ser aplicada à maioria dos problemas de otimização combinatória. Pode-se citar como exemplo as Metaheurísticas ACO (*Ant Colony Optimization*), GA (*Genetic Algorithm*), SA (*Simulated Annealing*) e TS (*Tabu Search*).
- Heurística: é a instanciação de uma metaheurística, ou seja, a aplicação da mesma em um problema específico de otimização. Por exemplo, em [DORIGO, 1991], denominou-se *AntSystem* a heurística ACO aplicada à resolução do Problema do Caixeiro Viajante. Na realidade, a heurística *AntSystem* foi primeiramente aplicada ao TSP, para depois ganhar dimensões de metaheurística, quando Dorigo [DORIGO, 1999] a flexibilizou para tal.

A princípio, os métodos aproximativos e os métodos heurísticos podem parecer iguais, o que não é fato. Ambos os métodos de resolução buscam, de maneira viável, trazerem soluções próximas àquela ótima, mas as semelhanças param por aqui.

Um método aproximativo garante que a solução se aproxima do ótimo a cada iteração, e realmente o faz, além de ser denotado matematicamente como eficaz. Já os métodos heurísticos não garantem qualquer tipo de melhora de solução conforme o número de iterações, além de não ter garantia de convergência.

Em primeira instância, os métodos aproximativos parecem ser mais eficientes que os heurísticos, o que também não ocorre. As heurísticas, embora tenham os problemas explicitados acima, costumam, na prática, convergir em tempo ex-

tremamente rápido quando comparadas aos métodos aproximativos, fato peculiar que fez com que esse método tivesse uma grande disseminação nos últimos anos.

## Capítulo 3

# Otimização por Colônia de Formigas

### 3.1 Preâmbulo: Agentes Inteligentes

Agentes inteligentes são *softwares* ou partes de *software* que estão situados dentro de um contexto ou ambiente e têm comportamento autônomo, ou seja, independente ou interdependente das outras partes do sistema.

Segundo Thagiah [THAGIAH et al., 2001], existem muitos atributos importantes aferidos a agentes inteligentes dentro dos quais, sobressaem como mais importantes:

- Antecipação - capacidade de antecipar mudanças no ambiente e agir baseado nestas mudanças;
- Comunicação - capacidade de se comunicar com outros agentes dentro de um sistema fechado;
- Autonomia - capacidade de controlar suas ações e tomar decisões por si próprio;
- Reação - capacidade de perceber mudanças no sistema e agir baseado nessas mudanças;
- Orientação a objetivo - cada agente tem um objetivo específico e é pró-ativo;



- Mobilidade - capacidade de trafegar de um local para outro sem interferência humana;
- Aprendizado - adaptação de acordo com experiências passadas;
- Persistência - capacidade de ficar longos períodos sem operações;
- Colaboração - capacidade de interagir com outros agentes ou outras partes do sistema, inclusive humanos;
- Distribuição - não existe uma autoridade central controlando um agente.

### 3.2 TSP por Ant System: a primeira implementação

Dorigo [DORIGO, 1991] foi o primeiro pesquisador da heurística de otimização por colônia de formigas, nomeando-a *Ant System*. Neste trabalho, a heurística *Ant System* foi aplicada ao Problema do Caixeiro Viajante (TSP).

O TSP consiste num conjunto de localidades a serem visitadas, obrigatoriamente, apenas uma vez, por um agente qualquer que, após tal ciclo, deve voltar a sua posição de origem. Uma instância do TSP pode ser representada pelo grafo valorado  $G(V, E)$ , onde  $V$  representa o número de vértices (localidades) e  $E$  consiste no número de arestas (estradas, por exemplo) do grafo, cada aresta com seu respectivo custo (distância).

Com  $b_i(t)$  relacionando o número de formigas presentes na localidade  $i$  em dado tempo  $t$ , o total de formigas no sistema é dado por  $m$ , sendo:

$$m = \sum_{i=1}^n b_i(t) \quad (3.1)$$

Na abordagem utilizada, algumas peculiaridades do ambiente das formigas foram utilizadas como parte da resolução do problema. Tais detalhes são:

- Ao tomar o caminho de uma localidade  $i$  até o destino  $j$ , a formiga deixa no mesmo uma certa quantidade de feromônio. Mais detalhes sobre os feromônios podem ser encontrados no apêndice A.

- Uma formiga escolhe determinada localidade  $j$  de acordo com uma função probabilística envolvendo a distância entre a localidade atual  $i$  e a localidade destino  $j$  e a quantidade de feromônio presente no caminho entre  $i$  e  $j$ .
- Para evitar que as formigas escolham vértices inválidos (ou seja, aqueles que já foram visitados), após a escolha os vértices são dispostos em uma lista tabu.

Nesta disposição, cada formiga representa um agente autônomo, cada qual com sua lista tabu (caminho percorrido até então). Com isto, em [DORIGO, 1991] denota-se um sistema com as características a seguir.

Primeiro, toma-se  $\tau_{ij}(t)$  pela quantidade de feromônio existente no caminho entre  $i$  e  $j$  em dado instante  $t$ . A cada iteração, a quantidade de feromônio em determinado caminho é constituída por

$$\tau_{ij}(t+1) = \rho * \tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (3.2)$$

sendo  $\rho$  um coeficiente de forma que o fator de evaporação do feromônio existente seja equivalente a  $(1 - \rho)$  e  $\Delta\tau_{ij}(t, t+1)$  consistindo na quantidade de feromônio depositada no intervalo de  $t$  a  $t+1$ , de acordo com

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (3.3)$$

onde  $k$  representa a  $k$ -ésima formiga.

No instante inicial ( $t = 0$ ), tem-se que  $\tau_{ij}(0)$  deve conter um valor baixo, escolhido arbitrariamente. Em [DORIGO, 1991] foi utilizado o mesmo valor para todas as arestas. O coeficiente  $\rho$  deve ser setado no intervalo  $[0 \sim 1]$ .

Retomando a lista tabu citada acima, Dorigo [DORIGO, 1991] enfatizou a diferença entre tal lista para a metaheurística *Tabu Search*. A única função da lista proposta no trabalho citado é de armazenamento de vértices utilizados para montagem da solução.

A escolha de uma aresta é um processo que deve ser feito com cautela. Um problema que pode acontecer no processo de escolha é a saturação do sistema de feromônios, levando o sistema a tender por soluções locais (ótimos locais).

Conforme sistema proposto em [DORIGO, 1991], dois fatores devem ser levados em conta no processo de escolha de uma determinada localidade:

- A visibilidade da localidade, denotada por  $\eta_{ij}$ , que leva em conta a proximidade (custo) da localidade no processo de escolha, dando ao algoritmo a característica de uma heurística construtiva; e
- A intensidade de feromônio no caminho, dada por  $\tau_{ij}(t)$ , onde o sistema leva em conta o quão interessante é o caminho, visto que quanto maior o valor de  $\tau_{ij}(t)$ , mais o caminho foi escolhido ao longo do sistema. Com este fator, a heurística ganha a característica de *autocatalítica*.

### 3.3 A Metaheurística ACO

#### 3.3.1 Introdução

Como uma heurística é um método aplicado a um problema específico, o trabalho [DORIGO, 1999] faz o *Ant System* tomar a forma de uma metaheurística, denominada *Ant Colony Optimization*, ou Otimização por Colônia de Formigas. Nesta metaheurística, uma colônia de formigas artificiais cooperam entre si com o objetivo de encontrar boas soluções para problemas de otimização discreta. A essência da metaheurística ACO é justamente a cooperação entre os agentes, no caso as formigas propriamente ditas, na varredura do espaço de busca pela solução ótima.

O experimento apresentado na figura 3.1 levou em conta um ninho de formigas disposto próximo a uma fonte de alimento, onde haviam dois caminhos possíveis para a escolha das formigas [DORIGO, 1999].

#### 3.3.2 Diferenças e Semelhanças entre Formigas Reais e Artificiais

A maioria das idéias utilizadas na metaheurística ACO provém do mundo real, ou seja, ocorrem realmente nas colônias de formiga. As características que conferem tal semelhança entre o natural e o artificial são:

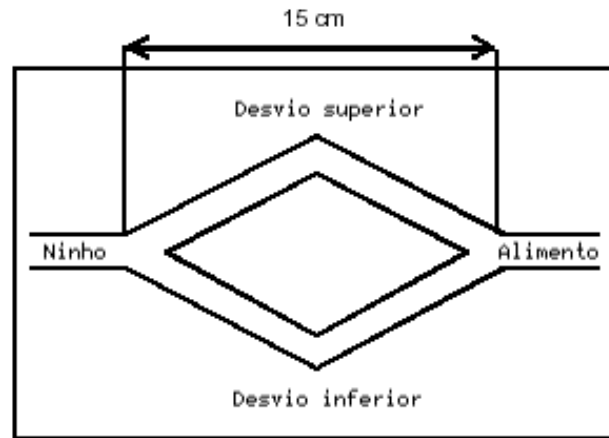


Figura 3.1: Experimento realizado para uma bifurcação no caminho das formigas em busca do alimento

- Ambos sistemas são constituídos por múltiplos agentes cooperando entre si. A metaheurística prevê uma colônia de agentes independentes entre si, em sinergia na busca de boas soluções. Tal cooperação entre os agentes ocorre pela leitura e escrita em um ambiente compartilhado;
- Ambos os sistemas utilizam um fator de cooperação, através do qual acontece a sinergia entre os agentes (feromônio). O feromônio representa a informação coletiva, e é essencial no desenvolvimento de ambos os sistemas. Como no mundo real, também ocorre a evaporação do feromônio, causando o *esquecimento* do passado remoto e trazendo às formigas a possibilidade de explorarem novos horizontes;
- Os agentes dividem a mesma função em ambos os universos: o de busca do menor caminho (ou do caminho com menor custo associado, no caso da metaheurística) entre uma origem (ninho) e um destino (alimento);
- O comportamento estocástico e local dos agentes na busca por soluções, ou seja, os agentes probabilisticamente escolhem a solução adjacente a ele, conferindo tal comportamento local e estocástico. Não há, em ambos os agentes, a visão supra adjacente (um agente não consegue visualizar um nó que se encontra a frente daqueles que estão em seu ambiente de visão atual).

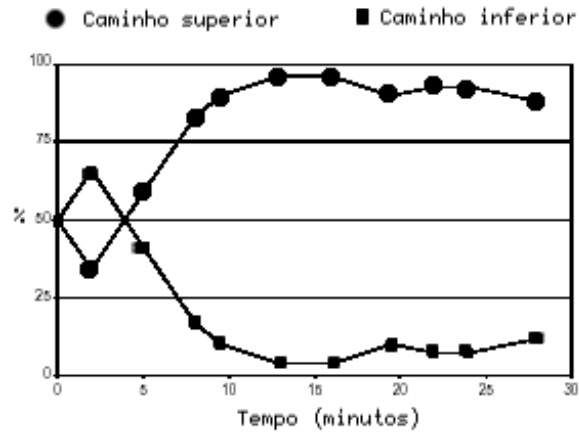


Figura 3.2: Valores obtidos em relação ao percentual de escolha entre o caminho superior e inferior

Da mesma forma, existem também dissemelhanças entre os dois universos. Segundo Dorigo ([DORIGO, 1999]) são elas:

- As formigas artificiais possuem movimentação discreta, sendo que seus movimentos consistem em origens e destinos discretos;
- Existe, nas formigas artificiais, um estado interno ou memória, para que não haja sobreposição de movimentos. No heurística aplicada ao TSP tal memória foi denominada *lista tabu*;
- O depósito de feromônio no mundo artificial ocorre com base na qualidade da solução encontrada;
- O diferencial do item anterior acarreta no depósito temporal do feromônio, diferentemente das formigas reais que depositam feromônio sob demanda;
- Como consiste em uma metaheurística, a ACO pode possuir adendos para otimizar ainda mais a busca por soluções. Tais recursos podem ser:
  - *Lookahead*, que consiste em analisar não apenas a próxima vizinhança de movimentos como também as vizinhanças subsequentes;

- Otimizações locais, onde um agente ataca o problema localmente antes de decidir por qual caminho seguir. Estas otimizações locais recuperam uma sequência de passos a seguir;
- *Backtracking*, ou seja, otimizações em caminhos já escolhidos - a fim de melhorar a rota até então designada. Também existem implementações com otimizações locais por *backtracking*, onde o subcaminho até então montado é melhorado.

### 3.3.3 ACO: Detalhes e Peculiaridades

A metaheurística ACO foi proposta em 1999 por Marco Dorigo, e está referenciada em [DORIGO, 1999]. A proposta é de um grupo de agentes inteligentes artificiais, as formigas, que compartilham informações através de um meio comum, o espaço de busca de soluções. Cada agente inteligente parte de um estado inicial  $p_i$  levando consigo uma memória pessoal com o caminho percorrido até o momento, sendo responsável por formar uma solução viável (não necessariamente ótima ou sequer competitiva) ou parte dela.

Ao construir uma solução, a formiga tem em mãos o ambiente, de onde ela retira informações em relação à presença de feromônio e também onde ela retira informações relativas à vizinhança atual. Não existe comunicação direta entre duas formigas - o meio onde as mesmas se encontram se torna um agente disseminador. O sistema é totalmente paralelizável, dada independência de cada formiga em construir uma solução. Ainda assim, para haver contribuição (cooperação), são necessárias algumas formigas no ambiente, pois só assim há sinergia entre o meio e as formigas.

A solução para o problema abordado toma a forma do caminho de custo mínimo entre os estados do problema, de acordo com suas restrições. Dada independência de uma formiga, até mesmo um sistema com uma única formiga é capaz de produzir uma solução completa, que possivelmente constituirá uma solução ruim. Para a otimização propriamente dita, é necessário um sistema com número adequado de formigas, o que irá resultar provavelmente em um bom resultado, que irá emergir da cooperação existente entre os agentes do sistema.

A construção de uma solução ocorre através da escolha, seguindo uma função probabilística, do melhor vizinho determinado por tal função. Cabe salientar que o conceito de vizinhança é totalmente dependente do problema. Esta função, estocástica, é uma busca local guiada pela (a) memória interna de cada formiga, quando disponível e pelo (b) feromônio, disponível através do sistema como um todo mais informações locais conhecidas a priori, geralmente constituídas por diretivas relacionadas ao problema.

Como especificado anteriormente, cada agente possui um estado interno para armazenagem relacionada a seu passado, utilizada para construir a solução final. Este estado interno é geralmente utilizado no cálculo da competitividade da solução atual, bem como no cálculo específico de possíveis impactos de escolhas tomadas ou iminentes. Por exemplo, especificamente para problemas de otimização combinatória, os agentes podem usar suas memórias para detectar uma escolha mal-feita, que estaria levando a solução atual para um estado inviável ou extremamente ruim.

A informação global, publicamente disponível, contém informações relacionadas ao problema bem como o conhecimento do sistema, constituído pelo feromônio desde a primeira iteração do sistema. Com isso, o feromônio constitui uma memória de longo prazo que influencia diretamente na decisão das formigas.

A forma de depósito do feromônio é livre, específica de cada implementação, e pode ser (a) *online step-by-step* (sob demanda), (b) *online delayed* (sob solução) ou (c) utilizando de ambas táticas. Em (a), o feromônio é atualizado a cada passo na montagem de uma solução, diferente de (b) onde o feromônio é atualizado a cada solução montada.

Denota o fator autocatalítico do sistema o fato propriamente especificado acima: quanto mais agentes escolherem um determinado arco, mais o mesmo será beneficiado na escolha dos agentes sub-sequentes. Como o feromônio é depositado de acordo com a contribuição do arco para a competitividade de uma solução, os arcos que mais contribuirão com soluções boas serão destacados no sistema. Um questionamento que se faz a partir deste ponto é em relação ao problema de se cair em ótimos locais com os arcos em questão. O fator decisivo na resposta negativa para tal questão é, mais que a evaporação ambiente, as variáveis específicas do

problema. Tais dados também influenciam o sistema a não viciar em pontos de ótimo local.

Por fim, ao encontrar sua solução, a formiga morre, ou seja, o agente inteligente é removido do sistema. Sua contribuição para o mesmo é constituída pela memória deixada para o grupo (feromônio), bem como pela solução encontrada, caso esta seja a melhor até então encontrada.

Além dos elementos-chave explicitados acima (geração e movimentação de formigas e evaporação de feromônio), a metaheurística de otimização por colônia de formigas permite também a adição de um elemento com função específica para otimizar o desempenho do sistema, um elemento com visão mais ampla (ou seja, no contexto global) em relação à visão local das formigas.

Dorigo [DORIGO, 1999] denominou este elemento por *daemon\_actions*, denotando um *thread* independente responsável por observar as ações tomadas pelas formigas e coletar informações globais que vão implicar em possíveis alterações na memória do sistema (feromônio). Resumidamente, um *daemon* consiste num agente global com função de otimizar o sistema a partir de sua visão ampla da atual configuração do mesmo.

Para coordenar todo o sistema supracitado, foi necessário um sincronizador, denominado *schedule\_activities*, cuja função é sincronizar as atividades de geração de novas formigas, depósito de feromônio e atuação de *daemons*. Além disso, o depósito de feromônio, tanto *online step-by-step* quanto *online delayed* é uma atividade que necessita de exclusividade para acesso no sistema, para não haver distorções no mesmo.

Com todos os aspectos acima descritos, a metaheurística ACO torna-se útil para problemas estocásticos com número relativamente baixo de vizinhanças por estado. Especificamente, o comportamento da metaheurística foi considerado viável, segundo Dorigo [DORIGO, 1999], para problemas relacionados a redes de comunicação ou transporte. Por outro lado, problemas com uma vizinhança muito grande em cada estado (por exemplo, grafos totalmente conectados) não são eficientemente solucionados, apresentando (a) tempo elevado de busca por soluções; e/ou (b) soluções não-competitivas.



Encontra-se descrito no Anexo B o algoritmo geral da metaheurística ACO, proposto em [DORIGO, 1999].

## Capítulo 4

# Proposta

### 4.1 Descrição do problema

O problema de transporte de cargas em ambientes reduzidos tem algumas peculiaridades que o difere dos problemas clássicos. A primeira e mais importante característica é que sistemas deste tipo operam em um espaço geográfico relativamente pequeno dado a distância entre os locais de carga e descarga, e o número de veículos transportadores. Por exemplo, um armazém de um atacadista onde coexistem vários pontos de carga, ou seja, locais onde as cargas estão ou devem ser armazenadas, vários pontos de embarque e desembarque de mercadorias, e um número considerável de empilhadeiras.

Estes ambientes estão sujeitos a um número limitado de rotas, geralmente curtas, o que implica em viagens de período reduzido entre os pontos de embarque e desembarque. Este tempo de coleta e entrega reduzido remete ao sistema um grande tráfego entre os pontos, o que pode gerar engarrafamentos e filas de transportadores. Como já apresentado na introdução deste trabalho, as filas são muito dispendiosas uma vez que acarretam a perda de produtividade do sistema e contribuem com o aumento do gasto de combustível e desgaste de maquinário.

A segunda característica destes sistemas é que não existem pedidos de transportes explícitos como no modelo geral do GPDP (seção 2.2.1). Tanto no GPDP como no PDP, os locais de coleta e entrega de determinado pedido são bem definidos, e este fato por si só já garante o tipo de carga, ficando para o sistema, a

decisão de qual transportador deve pegar e entregar a carga dado sua capacidade. O problema tratado aqui não contém estes pedidos explicitamente, e deve trabalhar sobre a produtividade e garantia da qualidade do que é carregado, transportado e entregue. Mas especificamente, os locais de descarga possuem, explicitamente, quais os tipos de carga que podem tratar. Assim quando um transportador qualquer carrega um determinado tipo de carga, o mesmo deve decidir para qual local de descarga deve ir, não obedecendo um cronograma dado por pedidos de transporte do modelo original. Este fato força o sistema a controlar a entrega nos locais corretos, diferente do modelo original onde isto já é pré-estabelecido. É muito fácil de perceber que, se um transportador deve escolher sua rota, o mesmo escolherá por um carregador caso esteja em um local de descarga, ou um local de descarga caso esteja em um carregador, tornando o sistema um grafo bipartido.

Estas duas características tornam o sistema altamente dinâmico, e mecanismos de controle devem ser feitos para que não ocorra disformidades na coleta e entrega. A terceira característica que deve ser levada em conta é o quanto a produtividade do sistema difere percentualmente da produtividade requerida. Esta característica está intimamente ligada com a segunda característica, pois dados os percentuais de produção requerida de cada carga, os mesmos devem ser respeitados ao máximo. Nas próximas seções, são apresentadas detalhadamente as características de sistemas deste tipo, com todas suas restrições.

#### **4.1.1 Dimensões do Problema**

O problema de transporte de cargas em ambientes reduzidos, onde a distância entre os pontos de coleta e entrega é relativamente pequena, tem como dimensões:

$T$  onde  $t \triangleq 1, \dots, m$ : conjunto de transportadores: os transportadores são os objetos destinados ao transporte de algum tipo de carga. Um veículo como um caminhão ou uma empilhadeira são exemplos transportadores;

$C$  onde  $c \triangleq 1, \dots, n$ : conjunto de carregadores: os carregadores são os objetos responsáveis pela carga ou preenchimento de um transportador. Geralmente um carregador trata apenas de um tipo de carga específico. Esta generalização será tomada como padrão neste modelo;

$D$  onde  $d \triangleq 1, \dots, o$ : conjunto de pontos de descarga: os pontos de descarga recebem a carga transportada, tratando-a. Um ponto de descarga pode receber um ou vários tipos de carga;

$I$  onde  $i \triangleq 1, \dots, p$ : conjunto de tipos de carga: é a classificação do que é extraído, transportado e entregue.

Em alguns modelos, têm-se o conceito de garagem ou oficina de onde os transportadores partem para início da operação e retornam logo após o término desta. Geralmente o termo depósito é utilizado na literatura para descrever tal conceito, e é representado aqui por  $g$ .

O modelo geral do GPDP (seção 2.2.1) e suas especificações nos modelos de PDP, DARP e VRP não levam em conta explicitamente o tipo de carga transportada, razão pela qual o conjunto de tipos de carga  $I$  não ter equivalente no modelo geral. Nestes modelos, o tipo está implícito nas requisições de transporte, de um ponto a outro, dado que se um pedido qualquer de um ponto  $a \in \mathcal{N}^+$  a um ponto  $b \in \mathcal{N}$  prevê que os próprios operam sobre os mesmos tipos ou tipos compatíveis de carga, já que a quantidade da mesma é conhecida em ambos pontos por  $q_i$ . Portanto, uma extensão deve ser feita pois o tipo de carga tem importância crucial para a consistência do sistema, uma vez que o problema estudado não apresenta pedidos de requisição de transportes, mas sim uma política de atendimento de demanda. Esta é a principal razão da descrição do modelo por completo.

#### 4.1.2 Parâmetros do Problema

Os parâmetros do sistema podem ser inúmeros dado o detalhamento do problema. Estes parâmetros podem levar o modelo a mudar de classificação facilmente uma vez que podem influenciar diretamente nas restrições. Portanto, a escolha dos parâmetros se faz mediante ao objetivo primordial do modelo. Aqui é apresentado um grupo de parâmetros e restrições que tendenciam o modelo a minimizar o tempo de fila, que como dito, diminui o custo da operação.

Nos transportadores têm-se os seguintes parâmetros:

$k_t$  : indica a capacidade de carga do transportador. Representado por  $Q_k$  no modelo geral ([SAVELSBERGH, 2001]);

$s_t$  onde  $s_t \in I$ : indica o tipo de carga em um determinado instante do tempo de um transportador;

$o_t$  onde  $o_t \in \{0, 1\}$ : indica se o transportador está em operação ou não.

No início da operação, os transportadores estão localizados em  $g$  (garagem), e retornam a este último no final do turno, que é o período de operação do maquinário representado por  $p$  onde  $p \triangleq 1, \dots, h$ .

Para os carregadores tem-se que:

$s_c$  onde  $s_c \in I$ : indica o tipo de carga com a qual determinado carregador trabalha. Como dito anteriormente, é considerado nesta monografia que um carregador trata apenas um tipo de carga;

$k_c$  : indica a taxa de carga de um carregador, ou seja, quanto o carregador consegue carregar em um transportador por unidade de tempo;

$o_c$  onde  $o_c \in \{0, 1\}$ : indica se o carregador está em operação ou não.

Os pontos de descarga podem variar muito de tipo de acordo com a aplicação do modelo. Determinados parâmetros podem deixar de existir dependendo da modelagem, mas em geral tem-se que:

$S_d$  onde  $S_d \subset I$ : conjunto de tipos de carga com os quais o local de descarga pode operar;

$k_d$  : indica a taxa de processamento de carga de um local de descarga;

$pr_d$  : indica a produção requerida em um determinado local de descarga;

$pa_d$  : indica a produção atual no local de descarga;

$br_{ds}$  onde  $s \in S_d$ : indica a proporção requerida de um determinada carga em um local de descarga. Geralmente utiliza-se o termo *blendagem* que vem do inglês *blending* que quer dizer mistura;

$qr_{ds}$  onde  $s \in S_d$ : quantidade requerida de determinada carga;

$ba_{ds}$  onde  $s \in S_d$ : indica a proporção atual de um determinada carga em um local de descarga;

$qa_{ds}$  onde  $s \in S_d$ : quantidade atual de determinada carga;

$o_d$  onde  $o_d \in \{0, 1\}$ : indica se o local de descarga está em operação ou não.

Todos estes parâmetros definem o detalhamento do modelo e o remetem a funções-objetivo diferentes. Este trabalho tem como finalidade minimizar o tempo de permanência em fila pelos transportadores nos carregadores e locais de descarga garantindo a qualidade da produção. Então alguns parâmetros não são tratados pelo algoritmo como será mostrado nas seções 4.2.2 e 4.2.3.

O trânsito dos transportadores entre os carregadores e os locais de descarga segue um ciclo bem definido que pode ser dividido em 4 sub-ciclos principais mostrados na figura 4.1. O ciclo completo compreende todo processo de carga e descarga contemplando a escolha do carregador, a viagem até este último, o processo de carga, a escolha do local de descarga, a viagem até este último e, por fim, o processo de descarga.

$tcc_{dc}^t$  : Tempo de Ciclo de Carga.

O tempo de ciclo de carga é o tempo gasto pelo transportador de um local de descarga específico até um carregador também especificado. Representado pela transição número 1 da figura 4.1;

$tc_c^t$  : Tempo de Carga.

O tempo de carga é o tempo entre a chegada do transportador em um carregador até sua saída. Representado pela transição número 2 da figura 4.1. Pode ser expresso como:

$$tc_c^t = \frac{k_t}{k_c}$$

$ttc_{dc}^t$  : Tempo Total de Carga.

Tempo total até a carga efetiva do transportador. Pode ser expresso como:

$$ttc_{dc}^t = tcc_{dc}^t + tc_c^t$$



Figura 4.1: Ciclos de Carga e Descarga

$tcd_{cd}^t$  : Tempo de Ciclo de Descarga.

O tempo de ciclo de descarga é o tempo gasto pelo transportador de um carregador específico até um local de descarga também especificado. Representado pela transição número 3 da figura 4.1;

$td_d^t$  : Tempo de Carga.

O tempo de carga é o tempo entre a chegada do transportador em um local de descarga até sua saída. Representado pela transição número 4 da figura 4.1. Pode ser expresso como:

$$td_d^t = \frac{k_t}{k_d}$$

$ttd_{cd}^t$  : Tempo Total de Carga.

Tempo total até a descarga do transportador. Pode ser expresso como:

$$ttc_{cd}^t = tcd_{cd}^t + td_d^t$$

Para efeito de cálculos pode-se considerar também o tempo médio de carga e o tempo médio de descarga e em conjunto os desvios padrões de ambos. Estes valores são muito importantes para análise do sistema uma vez que auxiliam a medição da performance geral. Além disso, são indicativos claros de filas e perdas de produtividade.

Além disso, faz-se necessária o parâmetro  $l^t$  onde  $l \in (C \cup D \cup \{g\})$  que indica a localização de determinado transportador. Esta variável é utilizada na composição de restrições que se seguem.

#### 4.1.3 Variáveis de Decisão

Dada as dimensões do problema, a decisão a ser tomada é:

$$R_{ij}^t \quad \text{tal que} \quad \begin{cases} i \in C \rightarrow j \in D; & \text{ou} \\ i \in D \rightarrow j \in (C \cup \{g\}); & \text{ou} \\ i = g \rightarrow j \in C \end{cases} \quad (4.1)$$

onde  $R$  é a rota a ser tomada pelo transportador.

#### 4.1.4 Restrições

##### Restrições de Tipo

As variáveis  $br_{di}$  e  $br_{di}$  assume valores entre 0 e 1, indicando percentagem.

##### Restrições Principais

A primeira restrição, que não de tipo, garante que um transportador esteja em apenas um localidade:

$$\sum_{l \in (C \cup D \cup \{g\})} l^t = 1 \quad \forall t \quad (4.2)$$



A segunda restrição garante que, se o transportador estiver operante, o mesmo seguirá por apenas uma rota:

$$\sum_{i \in D} \sum_{j \in C} R_{ij}^t \leq o_t \quad \forall t \quad (4.3)$$

A terceira restrição evita que um transportador leve uma carga para um local de descarga que não a possa tratar:

$$(l^t \in D) \rightarrow (s_t \in S_d) \quad \forall t \quad (4.4)$$

O transportador, quando dentro do período de operação, deve transitar entre carregadores e locais de descarga e, ao final deste, ir para a garagem:

$$(i = l^t | l^t \in D) \quad \text{então} \quad \begin{cases} p \geq h \rightarrow R_{ig}^t \\ \text{caso contrário } R_{ic}^t \end{cases} \quad (4.5)$$

#### 4.1.5 Função Objetivo

Um sistema de transportes admite variadas funções-objetivo, dentro das quais pode-se priorizar a obtenção de várias metas, e mesmo, balanceá-las. Frequentemente, a produção é o item mais importante destes sistemas. Produção não se dá apenas por elevado nível de extração, mas também pela maior utilização dos transportadores, carregadores e locais de descarga. Maior utilização significa produzir o quanto mais com o mínimo de gastos possíveis, sejam estes relativos a combustível, maquinário ou tempo. Dentro deste escopo, procurou-se minimizar ao máximo o tempo fila nos extremos dos sistema (carregadores e locais de descarga):

$$\text{minimizar} \left( \sum ttc_{dc}^t + \sum ttd_{cd}^t \right) \quad (4.6)$$

Esta função também contribui diretamente para maximizar a produção, uma vez que faz que o sistema tenda para um processo contínuo, sem esperas ou inatividades dos elementos.

## 4.2 Modelagem Baseada em Colônia de Formigas

### 4.2.1 Relacionamento entre entidades do modelo

Ao se observar a atividade de transporte de cargas em ambientes reduzidos, pode-se notar que todo processo é muito parecido com uma colônia de formigas ou abelhas. O sistema, geralmente baseado em extração, transporte e processamento pode ser modelado ou interpretado como uma destas colônias.

Cada carregador, ou seja, uma máquina ou pessoa, é incumbida de realizar a carga dos transportadores. Relacionam-se diretamente com as formigas cortadeiras, responsáveis em extrair alimento da natureza. As formigas cortadeiras são objetos semi-móveis que ficam em um lugar por determinado tempo, geralmente até o término do material extraído. Dependendo da qualidade do alimento extraído, estas formigas também podem se mover de um local a outro de extração em benefício da qualidade. É importante salientar que existem múltiplas formigas cortadeiras em diversos locais de extração, o que confere produtividade a colônia. Do mesmo modo, em um sistema de transporte de cargas, múltiplos carregadores, ou decarga, são utilizados para este fim também. Cada veículo ou transportador pode ser interpretado como uma formiga carregadeira, que tem como objetivo, transportar o alimento para a cultura de fungos de sua colônia, garantindo portanto a sobrevivência desta colônia e de si própria. A formiga carregadeira é um objeto de grande autonomia no sistema e suas decisões são tomadas individualmente baseadas no estado atual do sistema. As carregadeiras seguem seus caminhos baseados na informação de outras formigas, utilizando para isso, o **feromônio** que é um hormônio de rastreamento que as formigas seguem. As formigas têm a tendência de seguir a maior concentração de feromônio de agregação, convergindo para um local comum de extração. Isto é **vital** para a sobrevivência da espécie. No contexto de transportes, isto não é desejável, pois pode gerar filas imensas em apenas um local de carga ou descarga, reduzindo a produtividade do sistema. O processo de transporte de cargas é então **inversamente** proporcional ao sistema de feromônio de agregação das formigas já que as mesmas não enfrentam o problema de filas. O sistema de transportes pode se beneficiar do conceito de feromônio de repulsão existem na natureza e que tem a função de repelir outros indivíduos.

Têm-se também os locais de depósito ou descarga do material transportado. Em uma colônia, esse material concentra-se no formigueiro, na região onde as formigas cultivam o fungo que é seu alimento. Em um sistema de transporte, podem existir múltiplos locais de depósito, interpretados como múltiplas câmaras de cultivo de fungos no formigueiro. Nesses locais, devem ser atendidas algumas restrições e metas como a quantidade e o tipo correto de alimento para o fungo. Claramente, pode-se fazer um comparativo entre depósitos de carga onde determinados cargas não podem coexistir em um mesmo espaço, como por exemplo, produtos alimentícios e produtos químicos perigosos. A estes depósitos, também é atribuída uma capacidade de armazenamento e/ou processamento.

Conclui-se que as operações de um formigueiro são inerentemente operações de transporte de cargas, um sistema de excelente qualidade, dado que garante a sobrevivência da colônia.

#### 4.2.2 Funcionamento do modelo

Como se pode observar na seção anterior (4.2.1), um sistema de transporte de cargas em um ambiente complexo reduzido, é facilmente interpretado ou visto como uma colônia de formigas. Contudo existe uma peculiaridade que difere muito de um sistema para outro: a utilização de feromônio. Para as formigas, o feromônio é a única maneira de identificação e orientação espacial, destinado a levar uma formiga ao seu destino. O feromônio é o meio de convergência de todos os possíveis caminhos. Convergência é um grande problema nos transportes pois a utilização de apenas uma via, ou um ponto extremo do sistema gera **filas**, diminuindo a produtividade do sistema e promovendo um maior desgaste do maquinário, ou objetos envolvidos no sistema. Com isto, é preciso que se adapte então o modelo.

Uma solução para o problema seria **inverter** o significado do feromônio não como atrator mas como repulsor, um feromônio de alarme, garantindo que se um determinado transportador tenha escolhido uma determinada rota, outro transportador evitará seguir esta mesma rota afim de não gerar filas. Mas apenas isso não é suficiente pois ao se desviar a rota de um transportador que transporta determinada carga, deve-se garantir que seu novo destino suporta ou requer aquele tipo de carga. Deste modo, utiliza-se um novo parâmetro que tem o dever de garantir

que cada ponta do sistema, ou seja, um carregador e um local de descarga, mantenham a proporção correta das cargas que os mesmos podem e/ou devem tratar.

Portanto, dois parâmetros que influenciaram na escolha de um local pelo transportador: o feromônio de **tempo de fila**, repulsor, e importância de **carga**, atratora. A importância de carga é uma variável importante que, de maneira alguma, pode ser violada, pois garante a qualidade das cargas depositadas nos locais de descarga. Ao modo do exemplo da seção anterior (4.2.1), não pode ser permitida a estocagem de cargas alimentícias e produtos químicos em um mesmo local. O transportador deve fazer uma escolha **determinística e exata** segundo o tipo de carga. Além disso, deve ser garantida uma produtividade mínima para cada ponto extremo do sistema (locais de carga e descarga) evitando a postergação indefinida no atendimento das requisições destes locais. Já o feromônio de tempo de fila é uma restrição que pode ser relaxada, e nem sempre atendida. Segundo o modelo da colônia de formigas, cada formiga executa uma escolha probabilística referente ao caminho a tomar baseado na quantidade de feromônio que encontra pela frente. Toma-se como exemplo, um sistema saturado onde cada ponta do sistema está ocupada, tratando da carga ou descarga de um carregador. Um carregador que acabou de descarregar, deve ir a um ponto de carga. Dado que todos estes pontos estão ocupados, ou seja, repletos de feromônio de tempo de fila, o carregador então deve escolher o menor tempo de fila, mesmo que este ponto apresente o feromônio de repulsão. Uma escolha determinística pelo menor tempo. É importante salientar que esta escolha pode ainda ser relaxada e transformada em uma **escolha probabilística**, pois é provado estatisticamente que para um sistema convergir ao seu estado ótimo, o mesmo deve apresentar um componente probabilístico que evita o vício a um ótimo local.

### 4.2.3 Algoritmo

O algoritmo foi construído baseado nas premissas descritas nas seções anteriores e é um algoritmo distribuído baseado em eventos dentro de uma linha de tempo. O objetivo primordial é garantir a produtividade evitando a geração de filas no sistema, e garantir a qualidade das cargas entregues aos pontos de descarga evitando a possível mistura incorreta de cargas.

Cada ponto de carga e descarga contém um grupo de variáveis que representam o tipo de feromônio e as informações do sistema no modelo. O feromônio de fila é baseado em unidades temporais, e as informações de carga baseado em unidades de produtividade e asserção. Estas variáveis são modificadas constantemente pelos transportadores e pela simulação da evaporação, garantindo conformidade com a colônia de formigas. É crucial a adoção de medidas normalizadoras nestas variáveis impedindo variações extremas no sistema. Estas normalizações são dependentes da implementação do algoritmo.

Cada transportador tem uma estimativa de tempo de tráfego de um determinado ponto a outro, e utiliza esta informação para o depósito de feromônio de tempo de fila nos pontos extremos do sistema que pretende visitar. Cada par de (*local de carga*, *local de descarga*) tem uma estimativa baseada na média do tempo das viagens de um ponto a outro. É importante salientar que o tempo de um local de carga, dito *A*, a um ponto de descarga, dito *B*, é diferente de *B* para *A*, pois no primeiro caso, o transportador está carregado e conseqüentemente mais lento o que implica maior tempo de trânsito de *A* para *B*. No outro caso, o transportador estará descarregado conferindo maior velocidade e menor tempo de trânsito. O diagrama de colaboração 4.2 escrito em UML representa a interação básica entre elementos do sistema.

Pode-se notar que existem dois elementos a mais que não os já conhecidos *transportador*, *carregador*, e *local de descarga*. O elemento **concentrador** é um objeto controlador, e representa o sistema *per si*. Ele contém as referências para todos os carregadores e locais de descarga, oferecendo um ponto de acesso único para consultas aos transportadores. Na seção 4.3.1 será explicado detalhadamente sua implementação e o porquê de sua escolha. O elemento **otimizador** também é um objeto de controle que tem por finalidade garantir que as metas do sistema sejam atingidas através de atualizações nos feromônios.

O algoritmo é dividido em três fases: a primeira fase refere-se ao ciclo de carga que remete ao transportador o local onde deve pegar sua carga; a segunda fase refere-se ao processo inverso da primeira, ou seja, remete ao transportador o local de descarga; a terceira fase modifica as quantidades de feromônio afim de garantir que as metas sejam atingidas. Nestas fases estão inseridos grupos de eventos de-

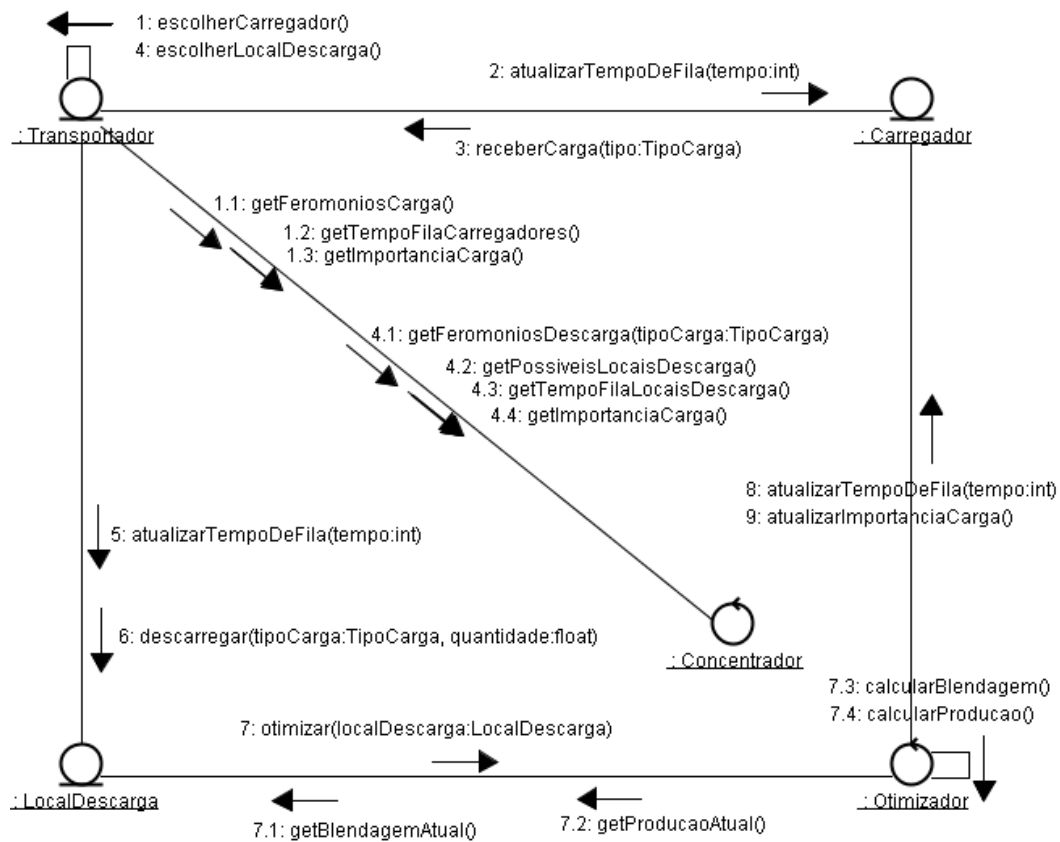


Figura 4.2: Diagrama de Colaboração do Algoritmo

pendentes diretamente da implementação e da implantação do algoritmo. A figura 4.1 mostra os ciclos de carga e descarga. Pode-se considerar como eventos:

**Escolher Carregador** : evento ativado quando o transportador é incitado a escolher um carregador afim de iniciar ou continuar sua produção;

**Carregar** : evento ativado quando um carregador começa a carregar um transportador.

**Escolher Local de Descarga** : evento ativado após o carregamento, e indica ao transportador que deve escolher um local para descarga;

**Descarregar** : evento ativado quando um transportador descarrega em um local de descarga.

Estes eventos podem ser resumidos em **Iniciar Processo de Carga** que abarca os eventos *EscolherCarregador* e *Carregar*, e **Iniciar Processo de Descarga** que abarca os eventos *Escolher Local de Descarga* e *Descarregar*.

### Fase 1

A primeira fase refere-se ao ciclo de carga representado na figura 4.1. Este ciclo parte do momento em que o transportador entra em operação ou acaba de descarregar em algum ponto de descarga. Primeiramente o transportador realiza uma requisição ao concentrador a fim de conhecer os feromônios correntes de cada ponto de carga (ação 1 do diagrama 4.2). Estes feromônios são compostos por um nível de fila, ou tempo de fila, e um nível de importância da carga (ações 1.1 e 1.2 do diagrama 4.2). Ao tempo de fila é acrescido o tempo estimado de percurso entre o local de descarga e o carregador. De posse dos feromônios, é aplicada sobre estes a seguinte função objetivo:

$$OBJ = \left( \frac{1}{Tempo\ de\ Fila + Tempo\ de\ percurso} \right)^\alpha \times \pm |Importância\ da\ carga|^\beta \quad (4.7)$$

onde  $\alpha$  e  $\beta$  são constantes definidas no modelo padrão de [DORIGO, 1999]. A importância da carga pode assumir valores tanto positivos, caso esta seja muito requisitada, como valores negativos, caso o sistema já esteja saturado desta última. Portanto, quando a importância da carga estiver positiva, a equação assume a característica aditiva. Quando negativa, a importância da carga força a redução no valor da função objetivo transformando o sinal em subtrativo. Esta característica é muito importante uma vez que os transportadores tentarão a escolher locais onde a importância estiver maior, mantendo o balanceamento dos tipos de carga. A importância de carga é calculada a partir de dados dos locais de descarga e será explicitada detalhadamente na fase 3 (4.2.3).

Após aplicada a função objetivo, o transportador escolhe o melhor valor encontrado, indicando o carregador escolhido através do incremento do tempo de fila do mesmo. Este incremento representa o depósito do feromônio, e é calculado sobre a equação 4.8. É importante salientar que nesta escolha, pode-se incorpo-

rar um componente probabilístico, tomando a decisão mais concisa com o modelo natural, e como já foi falado anteriormente na seção 4.2.2, evita que o sistema vicie em ótimos locais.

$$\text{Tempo de Fila}_c = tfa_c + ttc_{dc}^t \quad (4.8)$$

onde  $tfa_c$  é o tempo atual de fila no carregador  $c$ .

A capacidade do transportador e a capacidade de carga do carregador devem utilizar unidades de medidas compatíveis como por exemplo toneladas e toneladas por minuto, quilos e quilos por segundo, respectivamente. É claro que o tempo de fila também deve ser compatível com essas unidades. Caso isso não seja possível, uma normalização dos dados deve ser realizada.

Posteriormente, o transportador recebe uma mensagem de aviso de carga, indicando o tipo e quantidade da mesma que foi carregada. É ativada a segunda fase.

Como algoritmo básico tem-se:

1. O transportador requisita ao concentrador os feromônios dos carregadores;
2. É aplicada a função objetivo 4.7 sobre os feromônios;
3. O transportador escolhe o carregador que apresentar maior resultado;
4. O carregador escolhido é avisado através do incremento de seu tempo de fila;
5. O transportador recebe a carga.

É importante salientar que o recebimento da carga pertence a este bloco do algoritmo uma vez que o mesmo trata do ciclo de carga embora a carga efetiva seja feita em outro ponto do tempo. Esta abordagem temporal de efetivação do processo de carga é muito dependente da implementação e implantação do algoritmo. Se considerarmos um algoritmo com 4 eventos, a primeira fase toma para si os eventos **Iniciar Processo de Carga** e **Carregar** corretamente distribuídos na linha do tempo. Caso seja escolhida a abordagem com 2 eventos, o carregamento



é feito imediatamente após a escolha do local de carga. Isto pode ser perigo pois o sistema considera que o transportador já está carregado embora efetivamente esteja a caminho do carregador para realizar a carga.

## **Fase 2**

A segunda fase refere-se ao ciclo de descarga representado na figura 4.1. O ciclo de descarga compreende as ações entre o momento imediato ao carregamento até o descarregamento. O algoritmo da segunda fase é muito parecido com o da primeira fase, onde pode-se variar o  $\alpha$  e o  $\beta$  dado a meta a ser atingida. Segue-se o algoritmo:

1. O transportador requisita ao concentrador os feromônios dos locais de descarga;
2. É aplicada a função objetivo 4.7 sobre os feromônios;
3. O transportador escolhe o local de descarga que apresentar melhor resultado;
4. A importância de carga do carregador atual é decrementada a partir da equação 4.9;
5. O local de descarga escolhido é avisado através do incremento de seu tempo de fila;
6. O transportador descarrega no local de descarga escolhido.

Nota-se que ao escolher um local de descarga, o transportador emite um aviso ao carregador para que o mesmo diminua a importância do tipo de carga com que trabalha. Isso se faz necessário uma vez que se pressupõe que o transportador ao carregar e levar determinada quantidade de carga, o mesmo irá descarregar em algum local, diminuindo a importância deste tipo de carga para o sistema. Este tipo de abordagem pode ser problemática uma vez que o transportador tem a possibilidade de não descarregar o que está transportando, tendendo à inconsistência. Isso é resolvido com o modelo de atualização de importância de carga descrito na fase 3 (4.2.3) do algoritmo. A equação 4.9, para decremento da importância de carga,

calcula a proporção da importância daquele tipo de carga para um determinado local de descarga, dado a totalização desta mesma importância por todo o sistema:

$$Importância\ da\ Carga_c = ica_c - \left( \frac{ba_{dsc}}{nldt(s_c)} \right) \quad (4.9)$$

Então:

$ica_c$  : importância de carga atual de um determinado carregador;

$nldt(s_c)$  : número de locais de descarga que tratam determinado tipo de carga dado por:

$$nldt(s_c) = \sum_d s_c \in S_d$$

O incremento do tempo de fila também é praticamente similar ao incremento da primeira fase. A única diferença é que o tempo que o transportador leva para descarregar e a capacidade de processamento de carga do local de descarga devem ser levados em conta. Esse tipo de depósito pode variar muito de acordo com o tipo de local de descarga pois há locais que a capacidade de processamento pode ser desconsiderada, como por exemplo, um depósito de lixo não tratado. Em outros tipos de locais, como um equipamento de silagem de grãos ou gramíneas, ou um guindaste portuário, a capacidade de processamento deve ser levada em conta pois tem grande potencialidade de gerar filas, e em consequência, aumentar o custo da operação. Portanto:

$$Tempo\ de\ Fila_d = tfa_d + tdt_t + ttd_{cd}^t \quad (4.10)$$

tal que:

$tfa_d$  : Tempo de fila atual no local de descaga;

$tdt_t$  : Tempo de descarregamento do transportador.

Após incrementar o tempo de fila, o transportador emite um aviso de descarga para o local de descarga. Embora soe estranho, ou mesmo errado, a descarga é considerada anterior ao sua efetivação, devido a sua pertinência ao ciclo de descarga. Como no processo de carga, a escolha entre 2 ou 4 eventos impactua diretamente na estabilidade do sistema: com 4 eventos, mantemos as ações consistentes na linha do tempo das efetivas realizações destes. Com 2 eventos, consideramos que o transportador já descarregou embora efetivamente não tenha o feito.

### Fase 3

A terceira fase refere-se a atualização da importância de cargas nos carregadores. Estas atualizações devem garantir que a meta da operação seja atendida e que se mantenha a blendagem o mais próximo possível da blendagem requerida. Como já dito nas seções anteriores, esta garantia é dada pelo conceito de *importância de carga*, que é um percentual do nível de importância ou necessidade por determinado tipo de carga no sistema. Esta necessidade é tratada separadamente em cada local de descarga que conhece suas próprias necessidades. A importância de determinado tipo de carga em um local de descarga pode ser calculada como:

$$Importancia_{ds} = \frac{br_{ds} - ba_{ds}}{br_{ds}} + \frac{qr_{ds} - qa_{ds}}{qr_{ds}} \quad (4.11)$$

Esta equação retorna um valor percentual, dado que a primeira componente retorna um valor percentual entre 0 e 1, e a segunda uma aproximação também percentual afim de normalizar os dados. A importância é então dividida pelo número de locais de descarga que tratam o tipo de carga que esta sendo trabalhado, aos moldes da equação 4.9 mas agora em modo aditivo, ou seja, ao invés de diminuir a importância, a mesma é elevada nos carregadores. Isso garante que a porcentagem correta do tipo de carga seja alcançada.

Esta fase é ativada quando um transportador descarrega em um local de descarga, atualizando a produção atual. Por consequência, uma nova proporção é gerada, o que impele um novo cálculo na importância do tipo de carga específico no sistema. Então tem-se que, dados  $d$ ,  $t$  e um  $i \in S_d$ , para cada  $\{c | s_c = s_t \text{ e } s_c \in S_d\}$ , atualizar a importância da carga com a nova importância calculada  $Importancia_{ds}$ .

## **Evaporação**

O sistema de formigas admite uma atualização em alguns dados do sistema, geralmente nas informações utilizadas como feromônio. Esta alteração é comparada à evaporação que o feromônio sofre na natureza. Esta evaporação tem papel fundamental pois evita o acúmulo excessivo de feromônio em lugares específicos, por consequência, evita convergência prematura para ótimos locais.

O tempo de fila foi apresentado como um feromônio, e sua abstração permite, de modo muito natural, a evaporação. Toma-se como exemplo uma operação com 2 transportadores onde um dos mesmos está a caminho de um carregador, e outro está a caminho do local de descarga. Como já fora falado, o primeiro transportador ao escolher o carregador, aumenta o tempo de fila do mesmo baseado no tempo de tráfego e tempo de carga, por exemplo 10 unidade de tempo acrescidas no tempo de fila do carregador. O segundo transportador, ao descarregar no local de descarga 5 unidades de tempo depois da escolha do primeiro transportador, também irá ser incitado a escolher um novo carregador. Em tese, o carregador tratado no exemplo, tem 10 unidades de tempo em fila que na verdade deveriam estar em 5 unidades, pois no momento de sua escolha, o primeiro transportador está muito próximo de seu objetivo. Portanto, este tempo de fila deve ser atualizado constantemente para que seja mantida a consistência do sistema. Naturalmente é observável que a evaporação do tempo de fila pode e deve ser feita em unidades de tempo.

A importância de carga, se considerada como feromônio, também deve ser atualizada. Esta operação é feita naturalmente nos locais de descarga quando o transportador descarrega, e através da equação 4.9 nos carregadores. Esta operação está melhor descrita na fase 2 do algoritmo (4.2.3).

## **4.3 Implementação**

O algoritmo foi desenhado observando unidades interdependentes, que com comportamentos distintos, podem ser vistos como agentes inteligentes num ambiente de produção. Dado que estes agentes podem conviver em uma mesma máquina, ou em máquinas distintas, o algoritmo suporta uma implementação distribuída e

paralela, que aumenta o poder de processamento como um todo, além de distribuir a carga deste mesmo.

Num ambiente simulado, onde todos os agentes convivem em uma mesma máquina, utilizando o mesmo *pool* de recursos, a comunicação entre os mesmos se torna relativamente fácil de ser controlada, pois há a garantia de tráfego das informações necessárias para a operação. Em um ambiente distribuído, a comunicação entre estes agentes toma foco importante na construção do sistema. Primeiro porque as informações que trocadas entre os agentes tem importância crucial para a tomada de decisão. Segundo, que se há falhas nestas, o sistema inteiro pode ficar inconsistente e tender ao caos, segundo a teoria dos sistemas complexos e teoria do caos. Portanto a comunicação deve ser o mais confiável possível.

Uma forma de mitigar os riscos é avaliar o meio de transmissão destas informações e adequar o volume e a qualidade dos dados segundo a confiabilidade da rede de comunicação. O primeiro passo é conhecer o modo de transmissão a ser utilizado. Existe uma grande tendência que este tipo de operação seja feito em sistema de comunicação sem fio, um vez que o mesmo apresenta vários elementos móveis. Já é consagrado que o meio sem fio é extremamente suscetível a ruídos e esta característica diminui muito a qualidade na comunicação. A perda desta qualidade pode gerar lentidão na comunicação, e atraso no processamento de informações e tomada de decisão por parte dos agentes. Outro ponto relevante é quanto a cobertura da rede. Embora este trabalho trate de ambientes com dimensões reduzidas, que presuppõe uma área de cobertura relativamente pequena, o algoritmo pode ser aplicado em operações onde o espaço geográfico de trabalho é extenso e a cobertura da rede não o contempla todo. Neste caso, a comunicação é feita em pontos específicos do sistema, geralmente nos pontos de tomada de decisão que são nos carregadores e locais de descarga, e deve ser feita rapidamente, uma vez que a permanência do transportador nestes locais é momentânea.

Portanto a implementação depende de onde o algoritmo irá trabalhar e como este sistema será arquitetado, tanto em termos de comunicação, como em dimensões físicas e temporais.

### 4.3.1 Arquitetura do Software

Como dito na seção anterior, o ambiente onde o algoritmo irá trabalhar é um dos fatores decisivos na escolha da arquitetura e implementação desta. A princípio, a arquitetura desenvolvida neste trabalho fora uma arquitetura distribuída, mitigando ao máximo a transmissão de dados. Devido a inviabilidade técnica para testar esta arquitetura em um ambiente móvel distribuído, a mesma foi implementada em uma máquina que visa simular a linha temporal da operação, sem levar em conta a perda de comunicação. Ainda assim, foi mantido o princípio da arquitetura primordial, e apenas modificados, os modos de evaporação do feromônio e otimizadas algumas operações locais, para aumento na velocidade da simulação.

Primeiramente foram modelados os agentes dos transportadores que efetivamente, são os agentes que tomam decisão no ambiente escolhendo a rota a seguir. Estes agentes devem se comunicar com os carregadores e locais de descarga do sistema afim de realizarem sua operação. Dado que a comunicação é relativamente cara e custosa nos ambientes distribuídos, o transportador não pode guardar consigo todas as referências para os demais objetos. Assim é utilizado um objeto de controle, chamado aqui de *Concentrador*, que faz um papel de fachada, segundo os padrões de desenvolvimento vigentes. O concentrador funciona como um *pool* de recursos para os transportadores, guardando as referências para todos os carregadores e locais de descarga. Quando um transportador qualquer necessita de alguma informação, seja dos carregadores ou locais de descarga, o mesmo realiza uma chamada ao concentrador que repassa ao objeto específico. O concentrador realiza outras operações que não só repasse, uma vez que conhece todos os carregadores e locais de descarga. Outra razão de sua implementação é que o concentrador funciona como ponto de acesso único, o que reduz o número de chamadas e dados de controle, que não as informações principais do sistema. A figura 4.3 mostra a arquitetura básica. Nela não é mostrada a camada de comunicação entre os transportadores e o concentrador uma vez que depende da tecnologia empregada para tal.

Esta arquitetura é passível de implementação tanto distribuída como local (*stand-alone*).

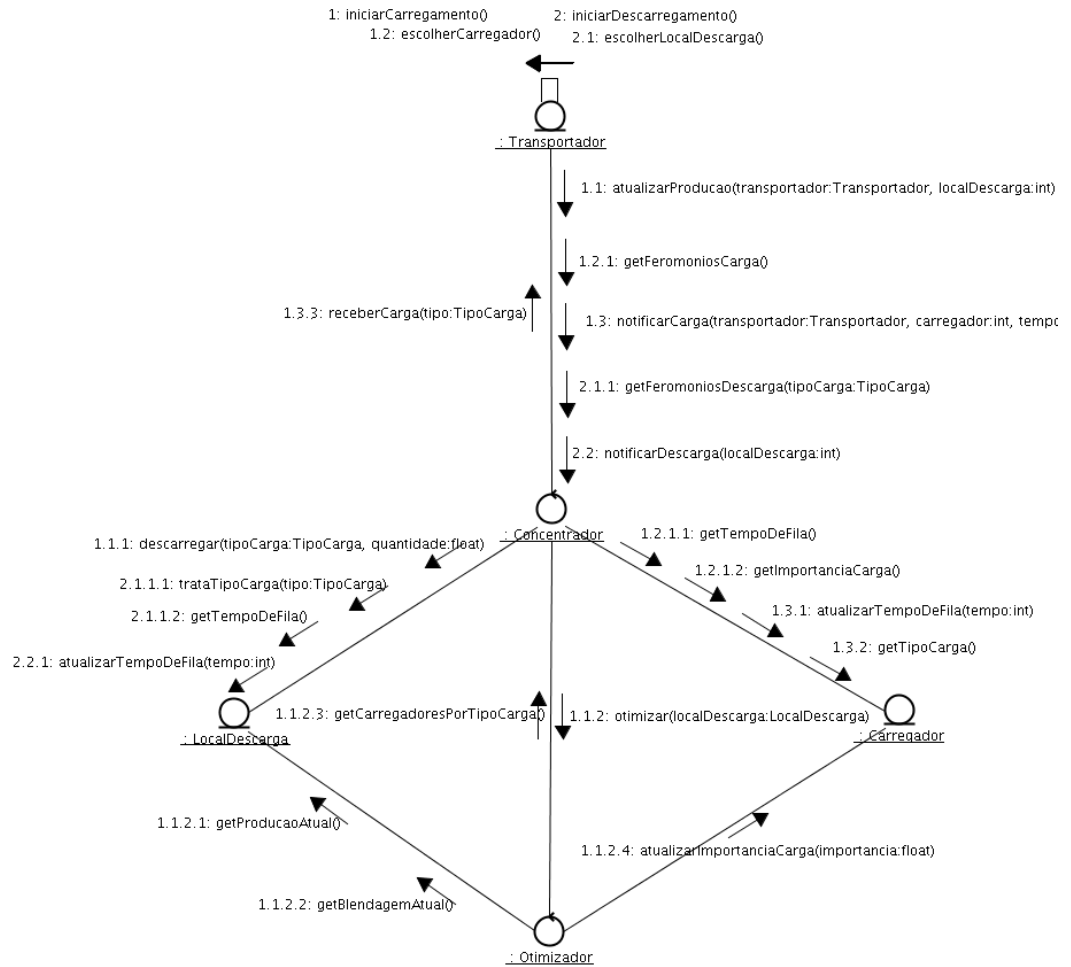


Figura 4.3: Arquitetura Básica

Para que seja testada, a arquitetura foi implementada em um programa principal que simula a linha de tempo e os eventos de um sistema real. Os eventos utilizados foram o **Iniciar Carregamento** e **Iniciar Descarregamento** como descritos na seção 4.2.3.

Este simulador constrói uma linha de tempo em unidades discretas, aqui tratando minutos, o que pode ser generalizado para outras unidades mediante normalização das unidades do problema. O mesmo mantém um linha de eventos de cada transportador, e sobre esta linha, decide quando chamar os eventos Iniciar Processo de Carga e Iniciar Processo de Descarga. Junto a estas linhas, tem-se as

linhas de tempo de cada carregador e cada local de descarga onde são detectadas e medidas as filas. O processo recebe um tempo de simulação compatível com a unidade utilizada na construção das linhas de tempo. O processo termina quando este tempo é atingindo.

#### 4.3.2 Tecnologia Utilizada

A implementação de um sistema distribuído é inerentemente ligada a uma tecnologia de implementação lógica e geralmente ligada a uma tecnologia de comunicação. Existem várias tecnologias para programação distribuída e várias linguagens que as suportam.

Neste trabalho foi utilizada a linguagem Java, por ser uma linguagem multiplataforma, o que facilita a implementação em hardwares não-padrões como PDAs, celulares e plataformas proprietárias. Outra vantagem de Java é sua forte convicção na orientação a objetos, que torna a modelagem do sistema mais natural e fácil de se realizar. A modelagem em si foi feita utilizando padrões da UML, em especial utilizando diagramas de classe, colaboração e seqüência, alguns destes mostrados em seções anteriores deste documento.

Embora Java suporte o conceito de objetos distribuídos utilizando RMI (*Remote Method Invocation*) ou CORBA (*Common Object Request Broker Architecture*), nenhuma destas tecnologias foi empregada. A razão para tal é que, em ambas, o *overhead* de comunicação é muito alto, e a comunicação em sistemas de transporte como os sistemas tratados aqui, é crítica em termos de banda e confiabilidade. Para isso, foi implementada uma camada de comunicação, que na realidade é um *framework* para utilização em diversos sistemas. Portanto, a comunicação entre os transportadores e o concentrador se dá através de interfaces bem definidas para módulos de comunicação que devem ser implementados de acordo com o sistema onde será implantado o algoritmo. Nestes trabalho em específico, foram supridos os módulos de comunicação por se tratar de um ensaio local ou *stand-alone*.





## Capítulo 5

# Resultados e Discussão

Um dos maiores problemas do modelo tratado neste trabalho é que não existem bases de dados para teste do mesmo. Para outros problemas, clássicos, pode ser encontradas várias bases de teste sólidas e confiáveis, geradas com rigor estatístico considerado correto, como as **instâncias de Solomon para VRPTW** e a **OR-Library**. Contudo, pelo conjunto de restrições aplicadas ao modelo gerado, nenhuma das instâncias citadas puderam ser adaptadas ao mesmo. Desta forma, não existem bases de comparação na literatura científica que possam mostrar a eficiência do modelo implementado *versus* modelos ou métodos já publicados.

Assim, para validação do modelo, foram utilizadas instâncias cuidadosamente criadas para levar o modelo à prova. As instâncias foram, então, simuladas de três maneiras:

- Atribuição estática por transportador: aqui, cada transportador, ao sair em turno, recebe um local de descarga e um carregador, que permanecem fixos até o final do turno;
- Atribuição determinística: de forma determinística, ou seja, em cada processo de carga ou descarga, os transportadores recebem seu destino automaticamente, levando em conta o melhor local de destino. Ótimos locais podem ser atingidos, o que torna essa abordagem menos interessante;
- Atribuição estocástica: a escolha de cada local de carga e descarga ocorre mediante uma função probabilística, da mesma forma que ocorre na meta-

heurística de otimização por colônia de formigas. É a forma proposta para resolução do problema. Para tanto foi utilizada a biblioteca científica The Colt Distribution versão 1.0.3 encontrada em <http://hoschek.home.cern.ch/hoschek/colt/>.

## 5.1 Instâncias Propostas

Em vista do problema enfatizado no capítulo 5, foram criadas 7 (sete) instâncias para teste, baseadas em observações empíricas de uma empresa que fornece serviços terceirizados para mineração. As instâncias descrevem um turno de operação com 8 (oito) horas, os locais de carga com seus respectivos tipos de minérios, os locais de descarga com as requisições destes minérios, e a frota de veículos transportadores onde tem-se capacidade e tempo de trajeto entre os diversos locais, como descrito na seção 4.1. Estas instâncias foram utilizadas na simulação do algoritmo proposto e na simulação de um roteamento estático.

A tabela 5.1 mostra as principais entradas das instâncias. As instâncias ímpares contém 30 transportadores. Cada instância par tem a mesma descrição da instância ímpar anterior mas o número de transportadores é reduzido para 10 (dez), e sendo assim, a produção é reduzida proporcionalmente a este número.

As instâncias 1 e 2 operam com 3 tipos de cargas/minérios (tabela 5.2) onde o tipo de carga 1 é tratado apenas pelo local de descarga 1, e o tipo de carga 3 é apenas tratado pelo local de descarga 2, e ambos locais tratam a carga 1. Isso gera um tráfego considerável no carregador 2, pois embora o minério que trate seja proporcionalmente menor em importância nos locais de descarga, ambos locais o requisitam.

Nas instâncias 3 e 4 é encontrado um certo balanceamento na distribuição das cargas pelo locais de descarga, mostrado na tabela 5.3.

Nas instâncias 5 e 6, foi feita uma restrição de entrega de um tipo de carga a um só local de descarga, proporcionando a geração de filas no carregador e local de descarga 2 (tabela 5.4).

Tabela 5.1: Descrição das Instâncias

Instâncias	Transportadores	Carregadores	Locais Descarga
01	30	3	2
02	10	3	2
03	30	3	2
04	10	3	2
05	30	4	2
06	10	4	2
07	30	3	2

Tabela 5.2: Blendagens das Instâncias 1 e 2

Local Descarga	Quantidade(Instância01/Instância02)	Cargas		
		Tipo 1	Tipo 2	Tipo 3
01	8000/3000	60	40	-
02	6000/2000	-	30	70

Tabela 5.3: Blendagens das Instâncias 3 e 4

Local Descarga	Quantidade(Instância03/Instância04)	Cargas		
		Tipo 1	Tipo 2	Tipo 3
01	8000/3000	40	30	30
02	6000/2000	20	30	50

Tabela 5.4: Blendagens das Instâncias 5 e 6

Local Descarga	Quantidade(Instância05/Instância06)	Cargas		
		Tipo 1	Tipo 2	Tipo 3
01	6000/6000	-	30	70
02	8000/8000	100	-	-

Tabela 5.5: Blendagens da Instância 7

Local Descarga	Quantidade	Cargas		
		Tipo 1	Tipo 2	Tipo 3
01	8000	60	40	-
02	6000	60	30	10

Por fim a instância 7 foi construída com 2 tipos de transportadores com capacidades de transporte diferente e montada às porcentagens similares as instâncias anteriores.

## 5.2 Gráficos e Análise dos Resultados Obtidos

### 5.2.1 Apresentação dos Resultados

Os resultados da simulação das sete instâncias propostas, contidas no anexo C, foram plotados em gráficos tridimensionais, utilizando o SCILAB (<http://www.scilab.org/>), um pacote de software científico para computação numérica.

Primeiramente, é apresentado o gráfico relativo ao desvio percentual do tempo de fila médio para as instâncias citadas e executadas deterministicamente. Desta forma, pode observar-se na figura 5.1 a menor média de tempo de fila obtido. É importante ressaltar que foi realizada uma normalização nos tempos de fila apresentados na figura 5.1 uma vez que os mesmos apresentaram uma discrepância muito grande na análise de todas as instâncias. Sendo assim, os valores conferidos no gráfico da figura 5.1 estão compreendidos entre 0 (zero) e 1 (um). Estes valores representam o cálculo percentual relativo à maior média de tempo de fila obtida entre todas as sete instâncias.

Ainda observando o gráfico da figura 5.1, nota-se que o menor desvio encontrado em relação à maior média de tempo de fila foi de 26.75%, sobre os valores de  $\alpha$  8 e  $\beta$  6.

A figura 5.2 apresenta a média dos desvios percentuais relativos às produções requeridas das sete instâncias estocásticas. Este desvio foi calculado comparando-

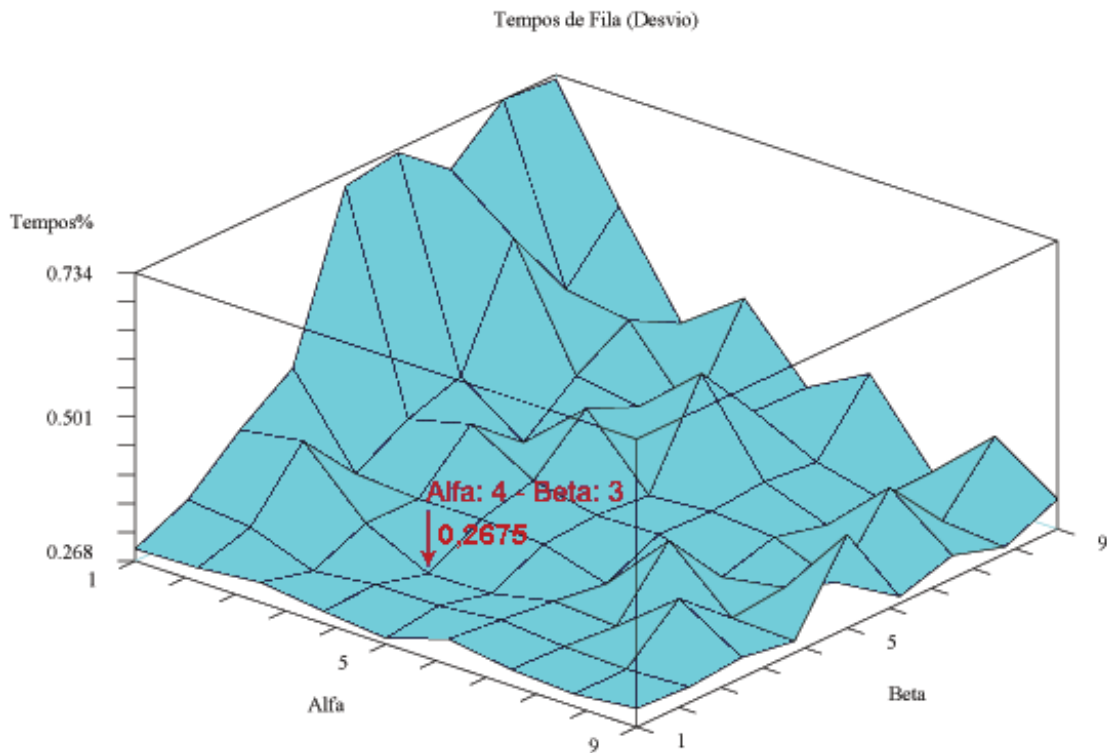


Figura 5.1: Desvio percentual das médias do tempo de fila para execução determinística

se a produção atingida pelo *software* para uma dada instância e a sua produção requerida. Desta maneira, quanto mais o valor do desvio médio aproximar-se de 0 (zero), melhor é este desvio. E quanto mais o mesmo aproximar-se de 1 (um), pior é este desvio. Pelo gráfico da figura 5.2 observa-se que a menor média dos desvios encontrada foi equivalente a 10, 19% para alfa 6, beta 5.

Os testes também foram realizados através de execuções estocásticas para a escolha do melhor carregador e do melhor local de descarga para o qual deveria encaminhar-se um dado transportador. A figuras 5.3 e 5.4 mostram, respectivamente, os gráficos relativos ao menor desvio médio de tempo de fila e ao menor desvio médio de produção. As unidades tratadas nestes gráficos seguem os mesmos padrões das explicitadas para as figuras 5.1 e 5.2.

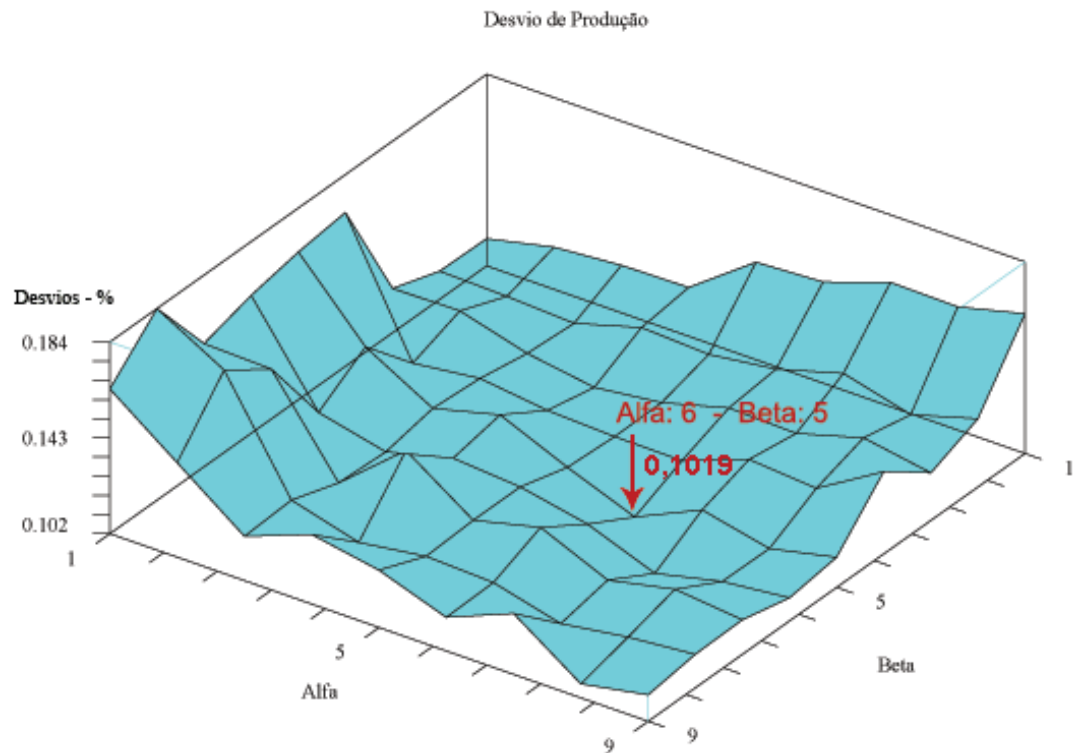


Figura 5.2: Desvio percentual das médias da produção para execução determinística

Para o gráfico da figura 5.3, observa-se que o menor valor encontrado foi em alfa igual a 9 e beta igual a 3.

Por outro lado, para os desvios médios de produção, ou seja, no gráfico da figura 5.4, o menor valor encontrado foi em alfa igual a 1 e beta igual a 8.

Na tabela 5.6, em instâncias ímpares (30 transportadores), pode-se observar uma clara vantagem no método estocástico. Já em instâncias com menos transportadores (instâncias pares, 10 transportadores), o método determinístico demonstra clara redução nos tempos de fila. Já nos desvios de produção ocorre o inverso: quanto maior o número de caminhões, maior o desvio de produção, na maioria das instâncias utilizadas. A análise destes resultados encontra-se na seção 5.3,

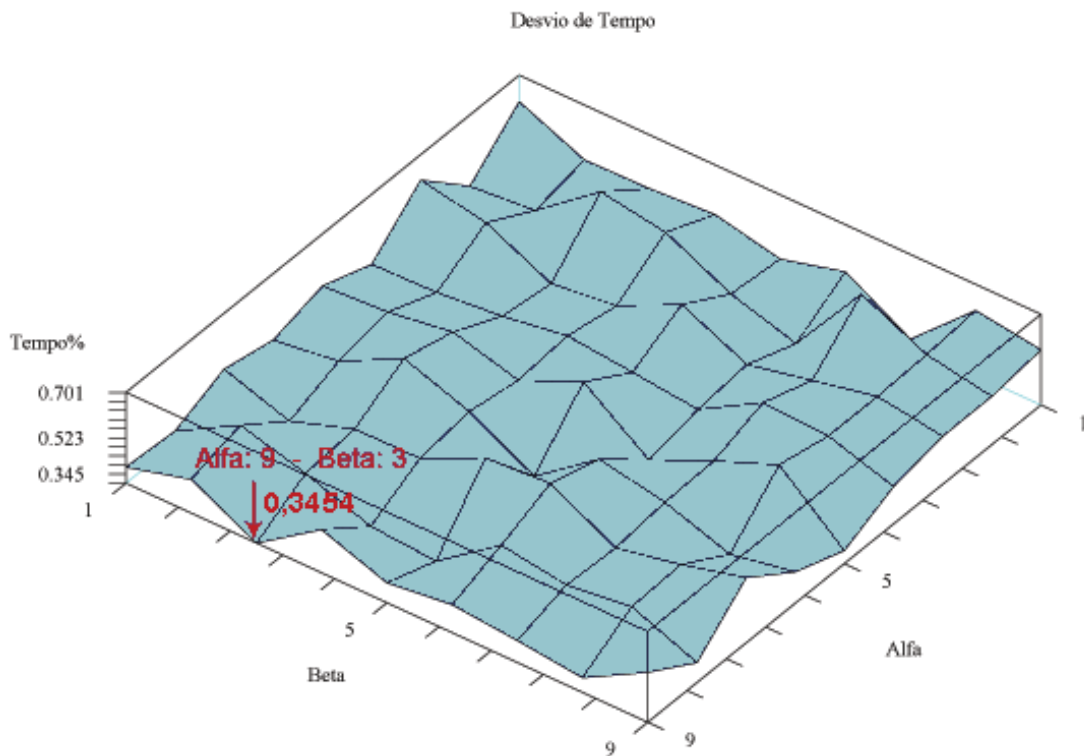


Figura 5.3: Desvio percentual das médias do tempo de fila para execução estocástica

### 5.3 Análise dos Resultados

A análise demonstrou uma clara melhoria nos resultados ao se aplicar a meta-heurística ACO da forma como foi modelada (estocástica). A tabela 5.6 demonstra que, quando existe diminuição no tempo médio de fila, na maioria das vezes há aumento no desvio médio de produção, ou seja, produziu-se mais.

Também pode-se notar como, em modo determinístico, ocorre a facilidade de queda em ótimos locais, ou seja, de soluções não ótimas. Nos tempos médios de fila existe uma diferença destacável entre os modos determinístico e estocástico, deixando claro o fato de que, nem sempre a escolha da melhor solução na vizinhança reflete numa solução ótima.

Ainda fica comprovado o aumento da inteligência do sistema (memória do sistema) ao se aumentar o número de transportadores (instâncias ímpares). Os



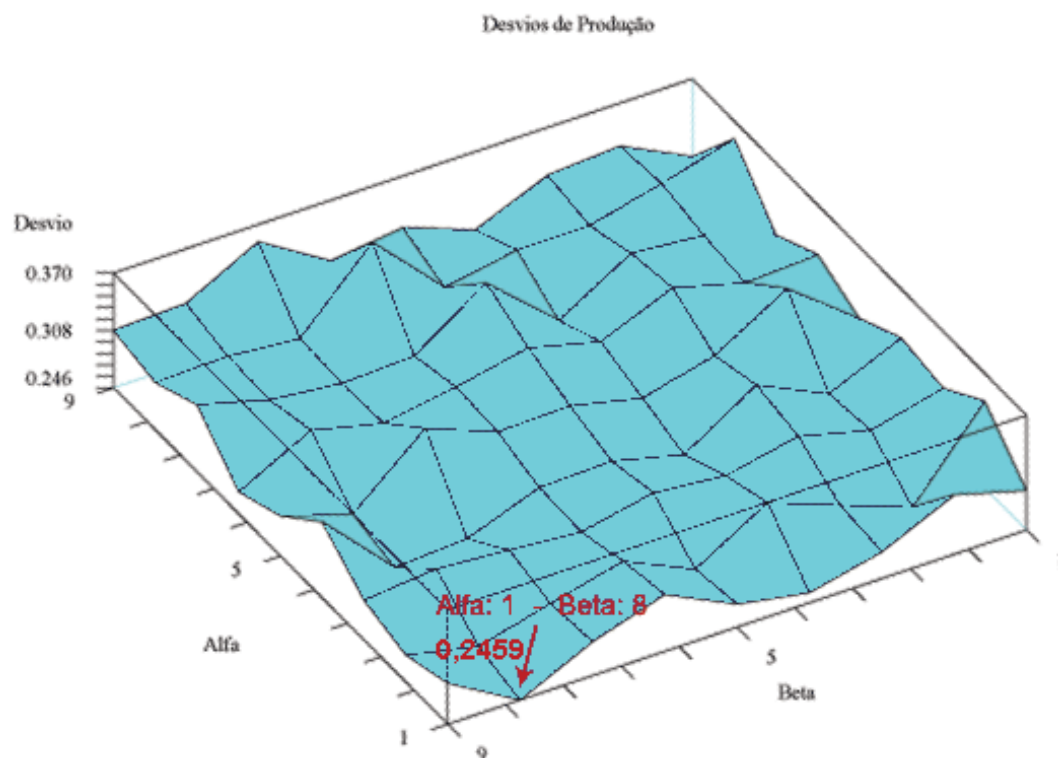


Figura 5.4: Desvio percentual das médias da produção para execução estocástica

resultados estocásticos se provaram de excelente qualidade quando comparados aos resultados estáticos ou determinísticos. Para melhor demonstrar tal fato, foram geradas as tabelas 5.6 e 5.7, que comparam os métodos de solução utilizados entre si. A tabela de instâncias ímpares possui simulação com 30 caminhões; já a tabela de instâncias pares foram simuladas com 10 caminhões.

Tabela 5.6: Tempos de fila e Desvios de Produção obtidos na simulação

Instâncias	Tempo Médio de Fila			Desvio Médio na Produção		
	Estático	Determinístico	Estocástico	Estático	Determinístico	Estocástico
1	4,20	11,76	2,77	0,35	0,18	0,07
2	0,20	0,10	0,06	0,06	0,14	0,11
3	4,83	4,22	2,72	0,22	0,16	0,21
4	0,00	0,02	0,11	0,17	0,15	0,24
5	1,20	3,09	2,70	0,04	0,06	0,24
6	0,00	0,01	0,12	0,04	0,07	0,22
7	9,63	14,66	4,40	0,06	0,12	0,99

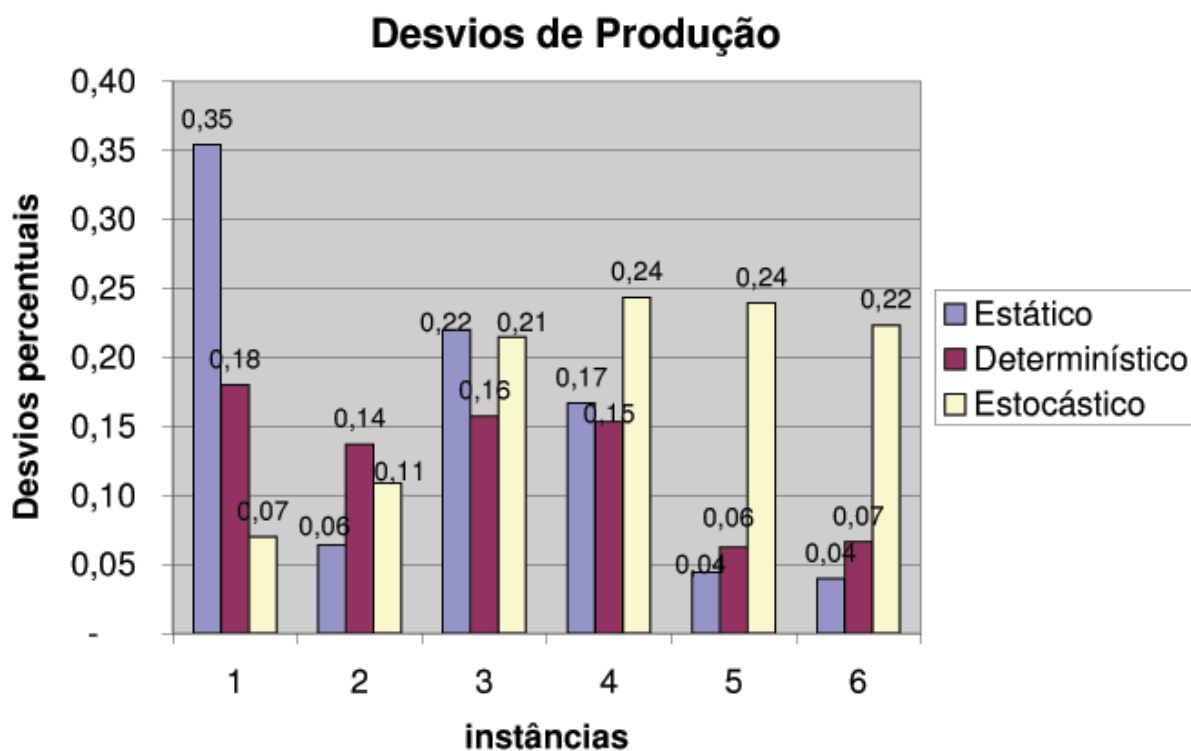


Figura 5.5: Desvios de produção nas simulações estáticas, determinísticas e estocásticas.

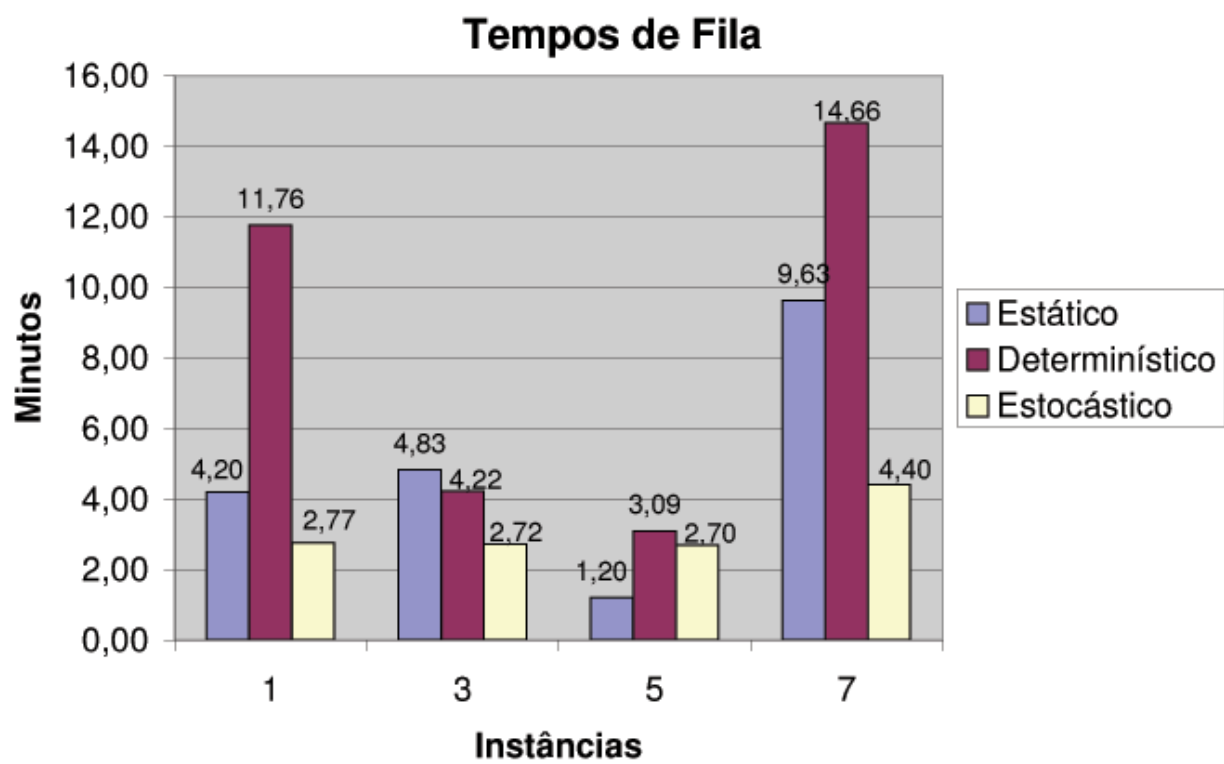


Figura 5.6: Comparação dos tempos de fila nas instâncias ímpares.

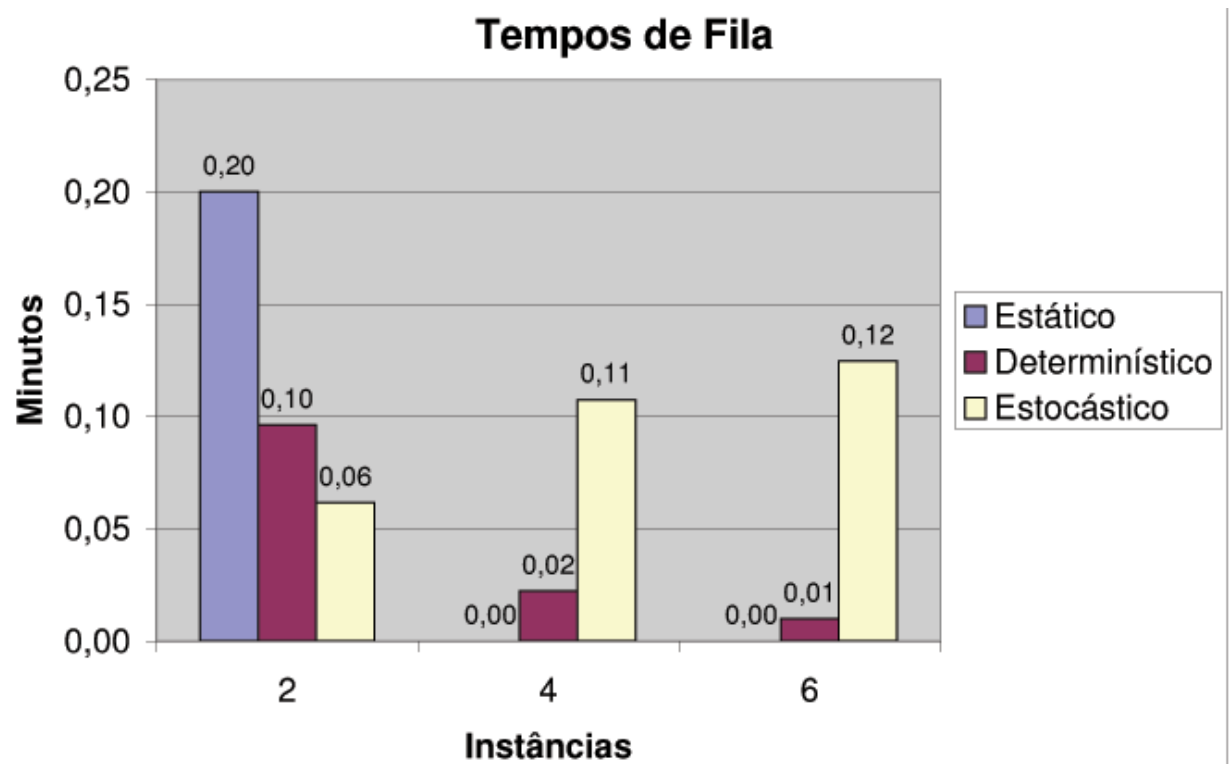


Figura 5.7: Comparação dos tempos de fila nas instâncias pares.



## Capítulo 6

# Trabalhos Futuros

Como trabalhos futuros a este, pode-se enumerar:

1. **Implementação do modelo utilizando outras metaheurísticas:** além da ACO, existem outras metaheurísticas com comportamento convergente em vários problemas clássicos, dentre as quais estão os Algoritmos Genéticos (AG) - *Genetic Algorithms* -, a Busca Tabu (*Tabu Search*) e Têmpera Simulada (*Simulated Annealing*). O maior estímulo está em buscar resultados melhores para as instâncias encontradas no anexo C.
2. **Utilização de métodos aproximativos:** pela não-garantia da convergência para uma solução ótima, os métodos heurísticos não são soluções elegantes na resolução de problemas de otimização; eles apenas conseguem obter soluções boas em tempo relativamente baixo. Assim, uma excelente via de abordagem se encontra nos métodos aproximativos, relatados no tópico 2.6.2, que trazem soluções na maioria das vezes viáveis, dependendo apenas da forma com que são concebidos.
3. **Modelagem paralela das formigas:** como num ambiente real, onde as formigas caminham em paralelo (enquanto uma formiga caminha em direção ao alimento, outra está levando-o à colônia, por exemplo), o problema acima também pode ter a mesma abordagem. Tratando cada caminhão como um agente inteligente, em Java<sup>TM</sup> pode-se modelá-los como *threads*, que consistem em rotinas paralelizáveis em processos simples. Cada *thread* é

executado de forma independente aos outros, e basta haver cautela no tratamento do ambiente compartilhado (provendo exclusividade de acesso mútuo através de semáforos, etc).

4. **Tradução do modelo em ambiente cliente-servidor:** o modelo de software utilizado possui claramente a capacidade de ser facilmente portado em ambientes cliente/servidor. Implementando o *Concentrador* como o servidor e dotando cada elemento com seu cliente específico (*Carregador*, *LocalDescarga* e *Transportador*), têm-se claramente o modelo cliente/servidor.
5. **Implementação do PDP clássico utilizando a metaheurística ACO:** com essa implementação pode-se comparar o comportamento da metaheurística ACO com relação a outras metaheurísticas existentes. A maior motivação se encontra no fato das formigas constituírem um modelo ideal de roteamento.
6. **Modelo para atender demandas dinâmicas:** atender a demandas dinâmicas é um problema de complexidade muito maior ao apresentado neste trabalho. Um modelo dinâmico tem de ser adaptável às mudanças do ambiente, como quebra de caminhos e recebimento de demandas em tempo real, por exemplo.

# Referências Bibliográficas

- [ALVARENGA, 2002] ALVARENGA, G. B. *Problema de Coletas e Entregas Dinâmico com Posicionamento Real (DPDP-RP)*. Departamento de Ciência da Computação, Universidade Federal de Minas Gerais.
- [BERTSIMAS, 1990] BERTSIMAS, D.; JAILLET, P.; ODoni, A. R. *A Priori Optimization*. Operations Research, 38(6):10191033, 1990.
- [BERTSIMAS, 1991] BERTSIMAS, D.; VAN RYZIN, G. *A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane*. Operations Research, 39:601615, 1991.
- [BULLNHEIMER, 1999] BULLNHEIMER, B.; HARTL, R. F.; STRAUSS C. (1999). *An Improved Ant System Algorithm for the Vehicle Routing Problem*. Paper presented at the Sixth Viennese workshop on Optimal Control, Dynamic Games, Nonlinear Dynamics and Adaptive Systems, Vienna (Austria), May 21-23, 1997, to appear in: Annals of Operations Research (Dawid, Feichtinger and Hartl (eds.): Nonlinear Economic Dynamics and Control, 1999.
- [COLORNI, 1994] COLORNI, A.; DORIGO, M.; MANIEZZO, V.; TRUBIAN, M. (1994). *Ant system for Job-shop Scheduling*. JORBEL- Belgian Journal of Operations Research, Statistics and Computer Science, 34(1):39-53.



- [CORMEN et al., 2001] CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Algoritmos: Teoria e Prática*. Tradução da Segunda Edição Americana por Vandenberg D. de Souza - Rio de Janeiro: Campus, 2002, p.763-807.
- [DORIGO, 1991] DORIGO, M.; MANIEZZO, V.; COLORNI A. (1991). *The Ant System: An Autocatalytic Optimizing Process*. Technical Report No. 91-016 Revised, Politecnico di Milano, Italy.
- [DORIGO, 1999] DORIGO M.; Di CARO, G.; CAMBARDELLA L. M. (1999). *Ant Algorithms for Discrete Optimization*. Artificial Life, 5(2):137-172. Also available as Technical Report No. 98-10 (IRIDIA), Université Libre de Bruxelles, Belgium.
- [FISCHER, 1995] FISCHER, M. L. *Network Routing*. v. 8, p. 133. Elsevier Science, Amsterdam, 1995.
- [GALLO et al, 2002] GALLO, D.; NAKANO, O.; CARVALHO, R. P. L.; BAPTISTA, G. C.; BERTI FILHO, E.; PARRA, J. R. P.; ZUCCHI, R. A.; ALVES, S. B.; VENDRAMIM, J. D.; MARCHINI, L. C.; LOPES, J. R. S.; OMOTO, C. *Entomologia Agrícola*. Fundação de Estudos Agrários Luiz de Queiroz - FEALQ, Piracicaba, 2002. p.270.
- [GAMBARDELLA, 1997] GAMBARDELLA, L. M.; DORIGO, M. (1997). *HAS-SOP: An Hybrid Ant System for the Sequential Ordering Problem*. Tech. Rep. No. IDSIA 97-11, IDSIA, Lugano, Switzerland.
- [GENDREAU, 1996] GENDREAU, M.; LAPORT, G.; SÉGUIN, R. *Stochastic Vehicle Routing*. Jornal Europeu de Pesquisa Operacional, 88:312,1996.
- [JAILLET, 1985] JAILLET, P. *Probabilistic Traveling Salesman Problems*. Tese de PhD, Instituto de Tecnologia de Massachusetts, Cambridge, Mass., 1985.

- [LARA, 1992] LARA, F. M. *Princípios de Entomologia*. 3ª edição. Editora Ícone, 1992. p.143-250.
- [LAPORTE et al., 1994] LAPORTE, G.; LOUVEAUX, F. V.; MERCURE, H. A *priori Optimization of the Probabilistic Traveling Salesman Problem*. Operations Research, 42:543-549, 1994.
- [LARSEN, 2000] LARSEN, A. *The Dynamic Vehicle Routing*. Tese de PhD. Departamento de Modelagem Matemática (IMM), Universidade Técnica da Dinamarca (file:/home/rtoso/DTU), 2000.
- [MALCA, 2003] MALCA, F.; SEMET, F. *A Tabu Search Heuristic for the Pickup and Delivery Problem with Time Windows and a Fixed Size Fleet*. LAMIH/ROI, UMR CNRS 8530, Université de Valenciennes et du Hainaut-Cambrésis, Le Mont Houy, 59313 VALENCIENNES Cedex 9.
- [MANIEZZO, 1994] MANIEZZO, V.; COLORNI, A.; DORIGO, M. (1994). *The Ant System Applied to the Quadratic Assignment Problem*. Tech. Rep. IRIDIA/94-28, Université Libre de Bruxelles, Belgium.
- [MELO, 2001] MELO, V. A. *Metaheurísticas para o Problema do Caixeiro Viajante com Coleta de Prêmios*. Dissertação de Mestrado, Instituto de Computação, Universidade Federal Fluminense, Niterói, Rio de Janeiro, 2001.
- [MITROVIC-MINIC, 1998] MITROVIC-MINIC, S. *Pickup and Delivery Time with Time Windows: A Survey*. SFU CMPT TR 1998-12.
- [PAPADIMITRIOU, 1982] PAPADIMITRIOU, C. H.; STEIGLITZ, K. *Combinatorial Optimization: Algorithms and Complexity*. USA, Dover Publications Inc., 1982.
- [PSARAFTIS, 1988] PSARAFTIS, H. N. *Vehicle Routing: Methods and Studies*. p. 223-248. Elsevier Science Publishers B.V. North Holland, 1988.

- [SAVELSBERGH, 1985] SAVELSBERGH, M. W. P. *Local Search for Routing Problems with Time Windows*. Annals of Operations Research 4, 285-305, (1985).
- [SAVELSBERGH, 2001] SAVELSBERGH M. W. P. *The General Pickup and Delivery Problem*. Escola de Engenharia Industrial e de Sistemas, Instituto de Tecnologia da Georia, 2001.
- [SILVA, 2003] SILVA, R. M. A.; RAMALHO, G. L. *Otimização Baseada em Colônia de Formigas Aplicada ao Problema da Cobertura de Conjuntos*. Tese de Doutorado apresentada ao CIn-UFPE (2003).
- [STÜTZLE, 1999] STÜTZLE, T.; DORIGO, M. I. *ACO Algorithms for the Traveling Salesman Problem*. Universidade Livre de Bruxelas, Bélgica, 1999.
- [TAHARA et al, 2001] TAHARA, Y.; YOSHIOKA, N.; OHSUGA A.; HONIDEN, S. *Secure and Efficient Mobile Agent Application Reuse Using Patterns*. SSR'01, Toronto, Canadá, 2001. Publicação da ACM, 2001.
- [THAGIAH et al., 2001] THAGIAH, S. R.; SHMYGELSKA, O.; MENNELL, W. *An Agent Architecture for Vehicle Routing Problems*. Departamento de Ciência da Computação, Universidade Slippery Rock. SAC'2001 Publicado na ACM, 2001.

## Apêndice A

### Feromônios

Segundo [GALLO et al, 2002] os insetos são, dentre os animais, aqueles que mais dependem do olfato para desempenhar suas atividades comportamentais. Os odores são muito importantes na localização de presas, na defesa e no ataque, na seleção de plantas, na escolha de locais de oviposição, na corte e no acasalamento, na organização social, entre outros tipos de comportamento. Para comunicar-se, geralmente os insetos utilizam substâncias químicas.

A palavra feromônio deriva do grego *pherein* que quer dizer carregar e de *horman* que quer dizer excitar, estimular. Esse termo foi introduzido por KARLSON & BUTERNANDT em 1959 (citado em [LARA, 1992]) e definido como sendo uma substância que, quando secretada por um animal e solta no meio ambiente, provoca uma reação específica num indivíduo receptor da mesma espécie. Cabe citar que essa definição encontra-se atualmente controversa, já que alguns feromônios agem em diversas espécies.

Segundo Lara ([LARA, 1992]), o início do estudo sobre feromônios começou em 1931 quando Ripley & Hepburn tentaram identificar a composição química do feromônio sexual

da mosca das frutas *Pterandrus rosa*, seguidos por Collins *et all* com a mariposa cigana *Porthetria dispar*. Mas foi em 1959 que o primeiro feromônio foi isolado e identificado. Este feromônio se chama *bombicol* e é o feromônio sexual do bicho-da-seda.

Existem 4 (quatro) tipos de feromônio:

**1. Feromônios de alarme:**

São feromônios utilizados para sinalizar perigo e ameaça, provocando fuga (ex.:pulgão), agressão contra outro inseto (abelha) ou inibição de agressão (formiga). Segundo [LARA, 1992], na maioria dos casos, esses feromônios estão associados às substâncias de defesa, como na subfamília *Formicinae* em que o veneno secretado é rico em ácido fórmico que age como feromônio de alarme. Existem certas espécies de formiga que produzem até 28 tipos diferentes de feromônio de alarme;

**2. Feromônios de dispersão ou rastros ou marcadores de trilhas:**

São feromônios utilizados para a manutenção de um espaço mínimo para sobrevivência(formiga) ou para anti-agregação (moscas-das-frutas). Geralmente servem para orientação espacial funcionando como atrator ou repulsor;

**3. Feromônios de agregação:**

Estes feromônios servem para manutenção das sociedades de insetos como as abelhas, colonização de novos habitats e agregação antes do acasalamento;

**4. Feromônios sexuais:**

Estas substâncias têm a finalidade de atrair o sexo oposto para reprodução. Têm sido, dentre outros feromônios, as mais estudadas, pois são as que mais oferecem possibilidades de uso, devido à sua alta potencialidade, mesmo em baixas concentrações.

Na metaheurística ACO o feromônio exerce um papel fundamental: o de comunicação e troca de conhecimento entre os agentes inteligentes (formigas artificiais). Sem o feromônio a capacidade do sistema como um todo seria bastante reduzida, com sérias impactações no que diz respeito à tendência de escolha de ótimos locais.



## Apêndice B

# Algoritmo para a Metaheurística ACO

```
1.  procedimento ACO_Metaheuristic()  
2.      enquanto( criterio_de_parada_nao_attingido )  
3.          schedule_activities  
4.              ants_generation_and_activity();  
5.              pheromone_evaporation();  
6.              daemon_actions();    // opcional  
7.          fim_schedule_activities  
8.      fim_enquanto  
9.  fim_procedimento  
  
10. procedimento ants_generation_and_activity()  
11.     enquanto( houverem_recursos_disponiveis )  
12.         agendar_a_criacao_de_uma_nova_formiga();  
13.         nova_formiga_ativa();  
14.     fim_enquanto  
15. fim_procedimento  
  
16. procedimento nova_formiga_ativa()
```



```

17.      inicializar_formiga();
18.      M = atualizar_memoria_da_formiga();
19.      enquanto( estado_atual != estado_final )
20.          A = ler_tabela_rotas_locais();
21.          P = computar_probabilidades( A, M, inf_prob );
22.          proximo_estado = politica_decisao( P, inf_prob );
23.          mover_para( proximo_estado );
24.          se( atualizar_feromonio = online_step_by_step )
25.              depositar_feromonio_no_arco_visitado();
26.              atualizar_tabela_de_rotas();
27.          fim_se
28.          M = atualizar_estado_interno();
29.      fim_enquanto
30.      se( atualizar_feromonio = online_delayed )
31.          avaliar_solucao();
32.          depositar_feromonio_nos_arcos_da_solucao();
33.          atualizar_tabela_de_rotas();
34.      fim_se
35.      morrer();
36. fim_procedimento

```

## Apêndice C

# Instâncias

### C.1 Descrição do formato do arquivo das instâncias

Na primeira linha tem-se o turno de trabalho ou operação do sistema, 480 minutos, por exemplo. Na segunda. Na segunda, terceira e quarta, respectivamente, o numero de transportadores, carregadores e locais de descarga.

Logo após, na quinta linha, iniciam-se a descrição de cada transportador, sendo que na linha ímpar, tem-se a capacidade do transportador e, na linha par subsequente tem-se os tempos de viagem de um determinado carregador para um determinado local de descarga e vice-versa.

Terminada a especificação para cada transportador, inicia-se a descrição de cada transportador, ou seja, cada linha descreve a capacidade de carregamento do carregador seguido, na mesma linha, do tipo de carga que ele carrega.

Depois dos carregadores, são detalhados os locais de descarga, onde a primeira linha, corresponde à capacidade de processamento do local de descarga, seguido, na mesma linha, da ca-

pacidade total de armazenamento do mesmo. As linha subseqüentes detalham o tipo de carga que cada local de descarga suporta e a quantidade (em porcentagem) necessária de cada um para atender este local de descarga.

**C.2 Instância 1**

480  
30  
3  
3  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30

[illegible]

0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
0.25 0  
0.25 1  
0.25 2  
0.25 8000  
2  
0 60  
1 40  
0.25 6000  
2  
1 30  
2 70

### C.3 Instância 2

480  
10  
3  
3  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
0.25 0  
0.25 1  
0.25 2  
0.25 3000  
2  
0 60

1 40  
0.25 2000  
2  
1 30  
2 70

### C.4 Instância 3

[illegible]



[illegible]

```
30
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20
0.25 0
0.25 1
0.25 2
0.25 8000
3
0 40
1 30
2 30
0.25 6000
3
0 20
1 30
2 50
```

## C.5 Instância 4

480  
10  
3  
3  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
30  
0 8 10 0 15 20 0 30 25 10 5 7 15 10 14 20 22 20  
0.25 0  
0.25 1  
0.25 2  
0.25 3000  
3  
0 40

1 30  
2 30  
0.25 2000  
3  
0 20  
1 30  
2 50

## C.6 Instância 5

480

30

4

3

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30

0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10

30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
 30  
 0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12

30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10  
0.25 0  
0.25 0  
0.25 1  
0.25 2  
0.25 6000  
2  
1 30  
2 70  
0.25 8000  
1  
0 100

## C.7 Instância 6

480  
10  
4  
3  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
30  
0 0 15 0 0 18 0 15 0 0 17 0 10 8 13 13 12 15 17 12 10 14 10 12  
0.25 0  
0.25 0  
0.25 1  
0.25 2  
0.25 6000  
2



1 30  
2 70  
0.25 8000  
1  
0 100

## C.8 Instância 7

[illegible]

30  
 0 8 10 0 15 20 0 30 25 11 5 7 18 10 14 27 22 20  
 30  
 0 8 10 0 15 20 0 30 25 11 5 7 18 10 14 27 22 20  
 30  
 0 8 10 0 15 20 0 30 25 11 5 7 18 10 14 27 22 20  
 30  
 0 8 10 0 15 20 0 30 25 11 5 7 18 10 14 27 22 20  
 30  
 0 8 10 0 15 20 0 30 25 11 5 7 18 10 14 27 22 20  
 30  
 0 8 10 0 15 20 0 30 25 11 5 7 18 10 14 27 22 20  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
 50  
 0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25

50  
0 12 15 0 17 25 0 35 30 14 8 13 21 13 17 29 25 25  
0.25 0  
0.25 1  
0.25 2  
0.25 8000  
2  
0 60  
1 40  
0.25 6000  
3  
0 60  
1 30  
2 10