

ANALDO TEIXEIRA DE SOUSA JÚNIOR

**PROPOSTA PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS
DINÂMICO COM JANELA DE TEMPO**

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

**LAVRAS
MINAS GERAIS – BRASIL
2007**

ANALDO TEIXEIRA DE SOUSA JÚNIOR

**PROPOSTA PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS
DINÂMICO COM JANELA DE TEMPO**

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

Área de Concentração:

Otimização

Orientador:

Prof. Guilherme Bastos Alvarenga

**LAVRAS
MINAS GERAIS – BRASIL
2007**

**Ficha Catalográfica preparada pela Divisão de Processos Técnico
da Biblioteca Central da UFLA**

Júnior, Analdo Teixeira de Sousa

Proposta para o Problema de Roteamento de Veículos Dinâmico com Janela de Tempo
/Analdo Teixeira de Sousa Júnior. Lavras – Minas Gerais, 2007. 46p : il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência
da Computação.

1. Informática. 2. Otimização Combinatória. 3. Algoritmo Evolutivo. I. JÚNIOR, A.T.S.
II. Universidade Federal de Lavras. III. Título.

ANALDO TEIXEIRA DE SOUSA JÚNIOR

**PROPOSTA PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS
DINÂMICO COM JANELA DE TEMPO**

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

Aprovada em 14/1/2008

Prof. Dr. Plínio de Sá Leitão Júnior

Prof. Dr. André Vital Saúde

Prof. Dr. Guilherme Bastos Alvarenga
(Orientador)

**LAVRAS
MINAS GERAIS – BRASIL
2007**

“Enquanto estivermos tentando, estaremos felizes, lutando pela definição do indefinido, pela conquista do impossível, pelo limite do ilimitado, pela ilusão de viver. Quando o impossível torna-se apenas um desafio, a satisfação está no esforço, e não apenas na realização final”. Gandhi

Dedico este trabalho a Deus e a minha família, que sempre me dão força para eu alcançar meus objetivos.

Agradecimentos

Agradeço primeiramente a Deus, que sem Ele nada seria possível.

A minha família, pelo amor e apoio que sempre me deram.

Ao meu orientador Guilherme, pelos seus ensinamentos.

Ao Humberto pela ajuda no princípio do projeto.

Aos colegas de curso, pelo companheirismo, espero encontrar todos, mesmo depois do fim do curso.

Aos amigos de República, que se tornaram uma segunda família para mim.

Aos professores do DCC e os funcionários, especialmente, Ângela e Deivson.

Aos meus avós, tios e primos.

E todos os meus amigos, que sempre fizeram com que minha vida tivesse mais alegrias que tristezas.

PROPOSTA PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS DINÂMICO COM JANELA DE TEMPO

RESUMO

O Problema de Roteamento Dinâmico de Veículos com Janela de Tempo é um problema de otimização muito estudado ultimamente, principalmente, devido à economia no preço de custos de transporte de pessoas e mercadorias que a sua solução oferece. O presente trabalho mostra uma contextualização dos problemas de roteamento de veículos, além de dois algoritmos usados para a solução desse problema, o Algoritmo Evolutivo e o Push-Forward Insertion Heuristic.

Palavras-chave: Problema de Roteamento de Veículo, Problema de Roteamento Dinâmico de Veículos com Janela de Tempo, Algoritmo Evolutivo, Otimização

PROPOSAL TO THE DYNAMIC VEHICLE ROUTING PROBLEM WITH TIME WINDOW

ABSTRACT

The Dynamic Vehicle Routing Problem with Time Window is an optimization problem studied lately, mainly, due to economy in the costs of people and merchandises transport that its solution offers. The present work shows a reference of the vehicles routing problems, besides two algorithms used for the solution of this problem, the Evolutionary Algorithm and the Push-Forward Insertion Heuristic.

Key-words: Vehicle Routing Problem, Dynamic Vehicle Routing Problem with Time Window, Evolutionary Algorithm, Optimization

SUMÁRIO

LISTA DE FIGURAS.....	vii
LISTA DE TABELAS.....	viii
1. INTRODUÇÃO.....	1
1.1. Considerações Iniciais.....	1
1.2. Objetivo.....	1
1.3. Escopo do Trabalho.....	2
2. REFERENCIAL TEÓRICO.....	3
2.1. Otimização.....	3
2.2. Problemas de Roteamento de Veículos.....	4
2.3. Problema de Roteamento de Veículos com Janela de Tempo.....	6
2.3.1. O Modelo Matemático do PRVJT.....	7
2.3.2. Complexidade do PRVJT.....	8
2.4. Problema de Roteamento de Veículos Dinâmico com Janela de Tempo.....	10
2.5. Métodos Exatos para Solucionar Problema de Roteamento de Veículos.....	11
2.5.1. Programação Dinâmica.....	11
2.5.2. Relaxação Lagrangeana.....	12
2.6. Métodos Aproximativos para Solucionar Problema de Roteamento de Veículos.....	12
2.6.1. Construção de Rotas.....	12
2.6.2. <i>Time-oriented-nearest-neighbour heuristic</i>	12
2.7. Métodos Heurísticos para Solucionar Problema de Roteamento de Veículos.....	13
2.7.1. Heurística <i>Push Forward Insertion Heuristic (PFIH)</i>	14
2.7.2. Colônia de Formigas.....	15
2.7.3. Recozimento Simulado.....	16
2.7.4. Busca TABU.....	17
2.7.5. Algoritmo Genético.....	18
2.8. Conjunto de Problemas Testes de Solomon.....	22
3. PROPOSTA DESTE TRABALHO.....	25
3.1. Algoritmo Evolutivo.....	25
3.2. Função Objetivo.....	25
3.3. Cálculo da Distância e Janela de Tempo.....	26
3.4. Representação dos Indivíduos.....	26
3.5. População Inicial.....	27
3.6. Cálculo de Aptidão.....	28
3.7. Seleção.....	28
3.7.1. Elitismo.....	29
3.8. Mutações.....	29
3.9. Resultados com Instâncias Estáticas.....	34
3.10. Adição e Remoção de Clientes.....	36
3.11. Coletas e Entregas.....	36
3.12. Esquema Geral da Proposta.....	37

3.13. Instâncias de Solomon Modificadas.....	39
4. CONCLUSÃO.....	40
5. TRABALHOS FUTUROS.....	41
6. REFERENCIAL BIBLIOGRÁFICO.....	42

LISTA DE FIGURAS

Figura 2.1: Um exemplo de solução para um Problema de Roteamento de Veículos.....	4
Figura 2.2: Estrutura Básica de um AG Simples.....	19
Figura 2.3: Cruzamento de dois indivíduos num AG Simples.....	21
Figura 2.4: Disposição espacial dos consumidores nos problemas das classes C1 e C2.....	23
Figura 2.5: Disposição espacial dos consumidores nos problemas das classes R1 e R2.....	23
Figura 2.6: Disposição espacial dos consumidores nos problemas das classes RC1 e RC2.....	24
Figura 3.1: Representação de um veículo com a sua rota.....	26
Figura 3.2: Representação de uma Solução Indivíduo.....	27
Figura 3.3: Representação de uma Roleta Giratória utilizada na Seleção de Indivíduos.....	28
Figura 3.4: Diagrama esquemático da Mutaç�o de Eliminaç�o de Rotas.....	29
Figura 3.5: Diagrama esquemático da Mutaç�o de Retirada de Consumidor.....	30
Figura 3.6: Diagrama esquemático da Mutaç�o de Inserç�o de Consumidor.....	30
Figura 3.7: Diagrama esquemático Mutaç�o por Troca Simples de Consumidores em Rotas Distintas.....	31
Figura 3.8: Diagrama esquemático da Mutaç�o por Troca de Consumidores em Rotas Distintas.....	31
Figura 3.9: Diagrama esquemático da Mutaç�o <i>PFIH</i> parcial.....	32
Figura 3.10: Diagrama esquemático da Mutaç�o por Reinserç�o de Consumidores com ganho em Dist�ncia Total.....	33
Figura 3.11: Diagrama esquemático da Mutaç�o de Busca Exaustiva por Troca com Ganho de Custo.....	33
Figura 3.12: Diagrama do Algoritmo Evolutivo proposto.....	38

LISTA DE TABELAS

Tabela 2.1: A seleção do DVRP (Dynamic Vehicle Routing Problem) na literatura.....	5
Tabela 3.1: Resultados obtidos na classe R1.....	34
Tabela 3.2: Resultados obtidos na classe R2.....	34
Tabela 3.3: Resultados obtidos na classe C1.....	35
Tabela 3.4: Resultados obtidos na classe C2.....	35
Tabela 3.5: Resultados obtidos na classe RC1.....	35
Tabela 3.6: Resultados obtidos na classe RC2.....	36

1. INTRODUÇÃO

1.1. Considerações Iniciais

Um dos fatores que influenciam o custo das mercadorias é o seu transporte. Por isso, cada vez mais, as empresas trabalham para ter uma boa programação das rotas para o melhor aproveitamento de seus veículos e para minimizar os custos.

Para conseguir esses objetivos, o problema de roteamento de veículos com janela de tempo (PRVJT) tem sido alvo de grande estudo nas últimas décadas. Contudo, a maioria dos estudos sobre o assunto trata o problema estático, em que todos os dados relevantes para a obtenção das rotas são previamente conhecidos. Enquanto que no mundo real, geralmente, os dados não permanecem estáticos ao longo da construção das rotas.

Sendo assim, nota-se que a variável tempo nunca foi tão importante como nos dias atuais. Considerando o problema em questão, as restrições de tempo incluem não só o intervalo para a entrega nos clientes, o que já é considerado no PRVJT tradicional, mas também o curto período disponível para chegada, carregamento, roteamento dos veículos e distribuição das mercadorias, além do surgimento de outros fatores, tais como, novos clientes, cancelamentos de pedidos, alteração nos custos nos trajetos, quebra de veículos, atrasos nas entregas, entre outros.

Conseqüentemente, um modelo considerando o tempo como variável do problema, vem ao encontro das necessidades atuais. As poucas propostas atuais para solução do PRVDJT (Problema de Roteamento de Veículos Dinâmico com Janela de Tempo) têm utilizado as mais diferentes formulações, uma vez que não há problemas testes (*benchmarks*) consolidados na literatura tratando-o de forma comum, o que torna mais difícil a avaliação de uma solução para o problema dinâmico.

1.2. Objetivo

O objetivo deste projeto é propor uma solução para o Problema de Roteamento de Veículos Dinâmico com Janela de Tempo, utilizando-se de métodos heurísticos capazes de gerar uma solução de boa qualidade em um tempo mínimo.

1.3. Escopo do Trabalho

Este trabalho descreve uma visão geral dos problemas de roteamento de veículos (Seção 2.2), dando ênfase ao Problema de Roteamento de Veículos com Janela de Tempo Estático (Seção 2.3) e Dinâmico (Seção 2.4). Os métodos exatos (Seção 2.5), aproximativos (Seção 2.6) e heurísticos (Seção 2.7), para a solução de problemas de roteamento de veículos, também são encontrados no Capítulo 2.

No Capítulo 3, encontra-se a proposta deste trabalho, Algoritmo Evolutivo para o Problema de Roteamento de Veículos Dinâmico com Janela de Tempo. Cada etapa do Algoritmo Evolutivo é descrita nessa seção:

- População Inicial (Seção 3.5);
- Cálculo de Aptidão (Seção 3.6);
- Seleção (Seção 3.7);
- Mutações (Seção 3.8);
- Adição e Remoção de Clientes (Seção 3.9).

A conclusão do trabalho está no Capítulo 4 e as propostas de trabalhos futuros no Capítulo 5.

2. REFERENCIAL TEÓRICO

2.1. Otimização

Segundo Papadimitriou et.al. (1982), um problema de otimização estende-se ao conjunto I de instâncias de um problema de otimização.

Uma instância de um problema de otimização é um par (F, c) , em que o F é um conjunto qualquer, o domínio dos pontos viáveis; c é a função custo, um mapeamento, descrito em (2.1):

$$c: F \rightarrow R^l \quad (2.1)$$

O problema é encontrar um $f \in F$ para cada:

$$c(f) \leq c(y) \text{ para todo } y \in F \quad (2.2)$$

Cada ponto f é chamado de uma solução de ótimo global para uma dada instância, ou simplesmente, solução ótima.

Além da definição de instância de um problema de otimização, alguns termos são importantes para um melhor entendimento sobre otimização, entre eles, vizinhança e ótimo local e global.

Dado um ponto viável $f \in F$, é útil em muitas situações definir um conjunto $N(f)$ dos pontos que estão próximos do ponto f . Em um problema particular com instâncias (F, c) , sua vizinhança consiste no mapeamento:

$$N: F \rightarrow 2^F \text{ definido para cada instância} \quad (2.3)$$

Em certas instâncias de problemas, encontrar uma solução ótima pode ser, do ponto de vista computacional, uma tarefa impossível. 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:

$$c(f) \leq c(g) \text{ para todo } g \in N(f) \quad (2.4)$$

2.2. Problemas de Roteamento de Veículos

O Problema de Roteamento de Veículo (PRV) foi introduzido por Dantzig e Ramser em 1959 (DANTZIG et.al, 1959). Segundo Alvarenga (2005), entende-se por roteamento de veículos uma ampla série de problemas, cujo objetivo é melhorar o custo relacionado a entregas ou coletas de mercadorias, através da escolha de rotas a serem percorridas por veículos.

Entre os muitos problemas relacionados, o mais generalizado é o Problema Geral de Coleta e Entrega (PGCE), Savelsbergh et.al (1995), que consiste na escolha de rotas para veículos de uma rota, a fim de atender às requisições de transporte. Cada veículo tem um ponto de partida e um ponto de destino e, durante a sua rota, atenderá a um conjunto de requisições. Savelsbergh considera a existência de três casos particulares para o PGCE: o Problema de Coleta e Entrega (PCE), ou *Pickup and Delivery Problem (PDP)*, onde cada encomenda é coletada em uma única origem e possui um único destino, cada veículo deve retornar ao seu ponto de partida no final da rota; o *Dial-a-Ride (DARP)*, onde a carga transportada são pessoas; e o Problema de Roteamento de Veículos (PRV) ou *Vehicle Routing Problem (VRP)*, onde todos os pontos de partida e destino final são idênticos para todos os veículos, um depósito central.

Há muitas variações do Problema de Roteamento de Veículos, Xu et.al (2001), enfatiza dois deles, o Problema de Roteamento de Veículos Capacitado, que considera uma frota uniforme de veículos de capacidade limitada, localizados inicialmente no mesmo depósito e sem limitação de tempo na entrega, e o Problema de Roteamento de Veículos com Janela de Tempo, que inclui uma janela de tempo como intervalo obrigatório para começar o atendimento dos clientes.

Um exemplo de problema de roteamento pode-se ver visto na figura abaixo:

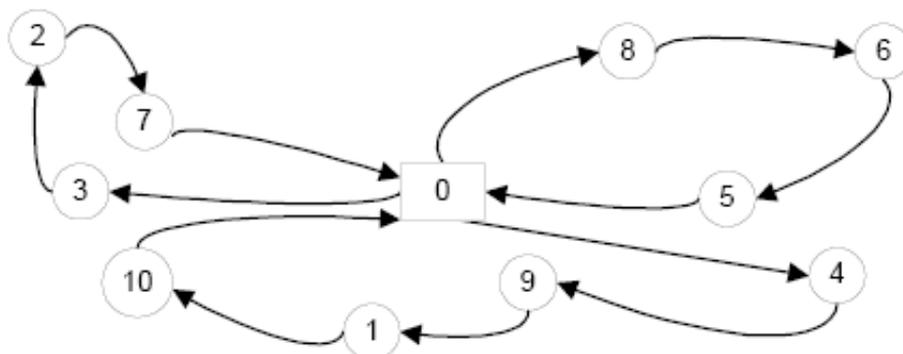


Figura 2.1: Um exemplo de solução para um Problema de Roteamento de Veículos

Uma outra variação do problema é encontrada em Archetti et.al (2001), *Split Delivery Vehicle Routing Problem*, em que os pedidos para cada veículo podem ser um número possivelmente maior do que a capacidade do veículo. Um consumidor pode precisar ser visitado mais de uma vez, ao contrário do que usualmente ocorre nos problemas de roteamento de veículo.

Já os problemas dinâmicos de roteamento, foram relacionados por Bianchi (2000), a tabela abaixo mostra o problema, a informação conhecida a-priori, a informação dinâmica e a referência na literatura.

Tabela 2.1: A seleção do *DVRP (Dynamic Vehicle Routing Problem)* na literatura, segundo Bianchi (2000)

Problema	Informação conhecida <i>a-priori</i>	Informação dinâmica	Referência na literatura
<i>DSVRP (Dynamic and Stochastic Vehicle Routing Problem)</i>	M veículos, capacidade finita	Quantidade de pedidos	Dror et.al. (1989)
<i>DTRP (Dynamic Traveling Repairman Problem)</i>	1 veículo, capacidade infinita	Nova localização dos pedidos	Bertsimas et.al. (1991); Papastavrou (1996)
<i>DTRP</i>	M veículos, capacidade finita e infinita	Nova localização dos pedidos	Bertsimas et.al. (1993)
<i>DTRP</i>	M veículos, capacidade finita e infinita	Localização dos pedidos	Bertsimas et.al. (1993)

Continuação da Tabela 2.1

<i>DVRPTW (Dynamic Vehicle Routing Problem with Time Window)</i>	M veículos, capacidade infinita, janela de tempo	Novos pedidos (localização, janela de tempo)	Gendreau et.al. (1999)
<i>DVRPTWD&P (Dynamic Vehicle Routing Problem with Time Window with Pick-up and Delivery)</i>	M veículos, capacidade infinita	Novos pedidos (localização de coleta e entrega, janela de tempo)	Gendreau et.al. (1998)
<i>DRIVE (Dynamic routing of independent vehicles)</i>	M veículos heterogêneos, mais depósitos	Novos pedidos (localização de coleta e entrega)	Savelsberg et.al. (1991)
<i>DTSP (Dynamic Traveling Salesman Problem)</i>	1 veículo, capacidade infinita	Exclusão e inserção de cidades	Guntsch et.al.(2000)

Maiores informações sobre os diferentes tipos de problemas de roteamento de veículos dinâmico podem ser encontradas em Psarafits (1995), Gendreau e Potvin (1998), Chen et al. (2005) e Potvin et al. (2005).

2.3. Problema de Roteamento de Veículos com Janela de Tempo

O Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) pode ser definido sobre um grafo completo $G = (V,A)$, em que o conjunto de vértices $V = \{v_0, v_1, \dots, v_N\}$ representa os clientes, exceto o v_0 que simboliza o depósito central e o

conjunto de arestas $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$ representa a distância entre os vértices. No depósito central (v_0), encontram-se m veículos idênticos, com capacidade máxima uniforme, que devem atender um conjunto de clientes, com um tempo fixo de atendimento e uma janela de tempo $[a_i, b_i]$ no qual a_i e b_i representam, respectivamente, a abertura e o fechamento da janela de tempo do vértice i .

O objetivo do PRVJT é encontrar um conjunto de rotas de custo mínimo iniciando e terminando no depósito central. Outros detalhes são citados em Gendreau et al. (1999):

- a) cada veículo deve atender uma única rota;
- b) cada cliente pode ser visitado uma única vez;
- c) se a chegada do veículo for antes da abertura da janela de tempo (a_i), o veículo deve aguardar, caracterizando um tempo de espera;
- d) o veículo não pode chegar no cliente i depois do fechamento da janela de tempo (b_i), no caso da janela de tempo rígida ou *hard time window*.

Entre as funções objetivo desse problema, as mais usadas são a distância total percorrida pelos veículos e a minimização do número total de veículos utilizados para chegar a solução.

2.3.1 – O Modelo Matemático do PRVJT

O modelo matemático, conforme Larsen (1999) para o Problema de Roteamento de Veículo com Janela de Tempo como:

$$\text{Minimize } \sum_{v \in \mathcal{V}} \sum_{i \in C} \sum_{j \in C} c_{ij} x_{ijv} \quad \text{Sujeito a} \quad (2.5)$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in C} x_{ijv} = 1 \quad \forall i \in C \quad (2.6)$$

$$\sum_{i \in C} q_i \sum_{j \in C} x_{ijv} \leq Q \quad \forall v \in \mathcal{V} \quad (2.7)$$

$$\sum_{j \in C} x_{0jv} = 1 \quad \forall v \in \mathcal{V} \quad (2.8)$$

$$\sum_{i \in C} x_{ihv} - \sum_{j \in C} x_{hjh} = 0 \quad \forall h \in C, \forall v \in \mathcal{V} \quad (2.9)$$

$$\sum_{i \in C} x_{i(N+1)v} = 1 \quad \forall v \in \mathcal{V} \quad (2.10)$$

$$s_{iv} + ts_i + t_{ij} - K(1 - x_{ijv}) \leq s_{jv} \quad \forall i, j \in C; \forall v \in \mathcal{V} \quad (2.11)$$

$$a_i \leq s_{iv} \leq b_i \quad \forall i \in C; \forall v \in \mathcal{V} \quad (2.12)$$

$$x_{ijv} \in \{0,1\} \quad \forall i, j \in C; \forall v \in \mathcal{V} \quad (2.13)$$

A equação (2.5) representa a soma de todos os custos associados a cada arco (i, j) utilizado na solução, percorrido por veículos v . Neste modelo o custo está associado unicamente a utilização ou não de cada arco, geralmente a distância percorrida do consumidor i para o j . A restrição da equação (2.6) garante que somente um veículo chega a cada consumidor i , ou seja, cada consumidor é atendido por um único veículo. A equação (2.7) garante que cada veículo v atenderá somente um conjunto de consumidores cuja demanda total não ultrapasse a sua capacidade Q . As equações (2.8) e (2.10) garantem, respectivamente, que cada veículo parte do depósito central e que ainda retorna ao mesmo ao final de sua rota. A equação (2.9) garante a continuidade das rotas, ou seja, se um veículo chega a um consumidor ele deverá sair do mesmo para o consumidor seguinte. A equação (2.11) garante que o instante de chegada de um veículo v a um consumidor j (s_{jv}) não poderá ocorrer antes do tempo de chegada no consumidor anterior i (s_{iv}) mais o tempo de serviço no primeiro (ts_i), mais o tempo de percurso no trecho (i, j) que é t_{ij} . É assumida uma velocidade constante tal que o tempo de percurso t_{ij} é diretamente proporcional à distância entre i e j . A constante K sendo suficientemente grande garante que a equação (2.11), ainda linear, seja somente uma restrição efetiva quando x_{ijv} seja igual a 1, ou seja, quando o veículo v percorre a trecho (i, j) . Adicionalmente este instante deve estar dentro da janela de tempo para o consumidor em questão, o que é garantido pela (2.12). Finalmente a equação (2.13) garante a integralidade binária das variáveis do problema.

2.3.2 – Complexidade do PRVJT

Segundo Alvarenga (2005), um problema é dito tratável se existe um algoritmo cuja complexidade de tempo e espaço seja polinomial, do tipo $O(n^k)$, para resolvê-lo, e

intratável caso contrário. Entretanto o desconhecimento de tal algoritmo não significa que ele não exista.

Em 1971, Cook (1971) deu uma contribuição significativa nesta matéria, estabelecendo a classe de problemas NP-completos. A classe de problemas cuja solução possa ser obtida por um algoritmo polinomial determinístico é chamada P. Muitos problemas, embora não possam ser resolvidos por algoritmos polinomiais determinísticos, uma solução pode ser testada em tempo polinomial, ou opcionalmente, a solução pode ser encontrada com algoritmos polinomiais não-determinísticos. Estes problemas são ditos da classe NP (*Non-deterministic Polynomial*). Cook prova a existência de uma sub-classe de problemas em NP correlacionados entre si, onde existe uma grande chance de serem realmente intratáveis, ou seja, de não existir um algoritmo polinomial determinístico para resolvê-lo. Atualmente, a complexidade dos algoritmos conhecidos para resolvê-los são exponenciais $O(k^n)$, fatoriais $O(n!)$, mas nunca polinomiais. Os problemas dessa classe contida em NP são chamados NP-completos. O primeiro problema NP-completo, provado por Cook, é o Problema da Satisfabilidade ou SAT, que consiste em dizer se existe ou não uma combinação de estados (sim/não) para variáveis de uma expressão booleana normal conjuntiva, que a tornem verdadeira. Cook provou que se houver um algoritmo determinístico capaz de resolver o SAT em tempo polinomial então $P=NP$. Através de transformações em tempo polinomial, o SAT já foi reduzido a uma série de outros problemas. Conseqüentemente, se algum destes problemas forem resolvidos por algoritmos polinomiais, o SAT também estará resolvido indiretamente, resultando $P=NP$. Por isso, acredita-se que dificilmente tal algoritmo exista.

Entretanto, os problemas NP-completos são todos do tipo decisão, como o SAT (Sim ou Não), e ainda devem estar na classe NP. Outra classe, de problemas também intratáveis como os problemas NP-completos, é a classe NP-difícil. Um problema Π é NP-difícil se todos os problemas NP são dedutíveis em tempo polinomial a ele. Na prática, basta provar que um problema NP-completo ou outro já comprovadamente NP-difícil pode ser transformado polinomialmente no problema Π em questão, para classificá-lo como pertencente à classe NP-difícil. Assim como os problemas NP-completos, um problema NP-difícil só pode ser resolvido em tempo polinomial se $P=NP$.

Os problemas de roteamento de veículos estão, em sua grande maioria, na classe NP-difícil. Isso significa que, a menos que $P=NP$ seja verdadeiro, o que parece não ser verdade, não existe um algoritmo polinomial capaz de resolvê-los.

2.4. Problema de Roteamento de Veículos Dinâmico com Janela de Tempo

O Problema de Roteamento de Veículos Dinâmico com Janela de Tempo (PRVDJT), segundo Larsen (2000) assume-se que não são conhecidas todas as informações relevantes antes do início do processo de roteamento. As informações referentes ao problema, como novos pedidos, cancelamento de entregas e coletas, alteração nos tempos e custos de trechos, mudam a todo instante.

Conforme Ribeiro e Lorena (2005), no momento da reprogramação das rotas, esse problema pode ser visto como um problema estático a ser resolvido.

A dinamicidade do problema faz com que apareçam pontos críticos não vistos no caso estático, eles são, de acordo com Alvarenga (2005):

- Coleta e entrega: quando o tipo de serviço no PRVDJT são coletas, estas podem ser realizadas a qualquer momento, mesmo o veículo já em trânsito, desde que exista ainda capacidade de carga, pois não há necessidade de voltar ao depósito central. O mesmo não ocorre na situação em que é necessário realizar novas entregas. Neste caso, poderá haver a necessidade de coletar a encomenda primeiro no depósito.
- Conhecimento das demandas: que pode ocorrer em tempo real ou em intervalos específicos.
- Atualização Cíclica ou em Tempo Real: novos pedidos, novas variáveis ou restrições podem surgir a qualquer momento. É comum, no entanto, aguardar um determinado período de tempo fixo antes de atualizar as variáveis e mudar a busca para uma nova solução. Isso pode ser feito visando ao agrupamento de uma certa quantidade de novas variáveis e novos pedidos, para que as alterações possam justificar as alterações nas rotas atuais.
- Preemptivo e Não-Preemptivo: é a possibilidade ou não de um veículo alterar a sua tarefa no meio do caminho. Imagine que um veículo sai de uma localidade em direção a outra, com uma entrega específica. O surgimento de uma necessidade em uma localidade nas margens do trecho, onde se encontra o veículo, poderá ou não ser atendida antes do atual destino: se este atendimento é considerado e possível, o sistema é dito preemptivo.

Muitas pesquisas estão sendo realizadas considerando o PRVDJT. Gendrau et al. (1999) desenvolveram uma heurística paralela de Busca Tabu dividindo o problema dinâmico em problemas estáticos aplicando-a nas instâncias propostas por Solomon (1987). Krumke et al. (2001) formalizaram a liberação em tempo real de unidades de serviço e propôs uma técnica de solução baseada em geração de colunas e particionamento. Gunstsh e Middendorf (2002) propuseram um algoritmo heurístico baseado em colônias de formiga para o problema do caixeiro viajante dinâmico. Montemanni et al. (2003) desenvolveram uma estratégia de solução para o PRVDJT baseada em Sistemas de Colônia de Formigas, que foi testada sobre um conjunto de testes proposto por Kilby et al. (1998) que alteraram as instâncias propostas por Taillard (1994), Christofides e Beasley (1984) e Fisher et al. (1981).

2.5. Métodos Exatos para solucionar Problema de Roteamento de Veículos

Segundo Andrade et. al. (2004), os métodos de busca por soluções são aqueles que sempre encontram a solução ótima, ou seja, a melhor solução para o problema, se é que esta existe.

Segundo 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.

Por esta razão, as abordagens exatas devem ser utilizadas para pequenas instâncias do problema, pois para os demais (problemas maiores), elas demandariam muito tempo de processamento (BEZERRA, 2005).

Nos métodos exatos são consideradas principalmente técnicas do tipo *relax-and-cut* que visa a combinar relaxação lagrangeana com teoria poliédrica e *branch-and-bound*.

2.5.1. Programação Dinâmica

Conforme Passos et. al. (2003) citado por Oliveira (2004), a programação dinâmica baseia-se na técnica de *branch-and-bound* para alcançar o ótimo. Um limite inferior é verificado para cada nó incluído na solução. Quando algum nó propicia um limite inferior

atual, este limite inferior torna-se o novo limite inferior. Cada nó referente ao cliente é visitado somente uma vez. Mais informações do algoritmo sendo aplicado ao problema de roteamento pode ser encontrado em Kaan et. al. (1987).

2.5.2 Relaxação Lagrangeana

No método de relaxação lagrangeana, o problema original com inúmeras restrições, normalmente complexo, tem algumas restrições relaxadas, permitindo sua simplificação. Essas restrições relaxadas são penalizadas através de multiplicadores de Lagrange, os quais são gradativamente ajustados ao longo das iterações. A seqüência de soluções gerada cada vez mais se aproxima da solução ótima do problema. Uma visão dessa técnica poderá ser vista em Maculan et. al. (1984) e Fisher (1985).

2.6. Métodos Aproximativos para solucionar Problema de Roteamento de Veículos

Os algoritmos aproximativos (especialmente os determinísticos aproximativos), foram introduzidos por Johnson (1974), e são algoritmos polinomiais que buscam sacrificar o mínimo possível da qualidade, obtida nos métodos exatos, ganhando, simultaneamente, o máximo possível em eficiência (tempo polinomial). Como discutido em Hochbaum (1997), a busca do equilíbrio entre estas situações conflitantes é o grande paradigma dos algoritmos aproximativos.

2.6.1. Construção de Rotas

O algoritmo inicia-se com todas as rotas simples possíveis com apenas um consumidor. E a cada interação, calcula-se quais duas rotas podem ser combinadas com a maior economia de recursos. Este algoritmo foi utilizado em Backer et.al (1986).

2.6.2. *Time-oriented-nearest-neighbour heuristic*

Proposta por Solomon (1986), toda rota é inicializada por um consumidor ainda não roteado mais próximo do depósito central. Esta relação de proximidade é tanto temporal quanto geográfica. A cada iteração o consumidor mais próximo, geográfica e temporalmente, ao último cliente adicionado é considerado para inserção na rota em

questão. Quando a pesquisa por um novo consumidor é feita, e um consumidor não é encontrado, uma nova rota é iniciada.

2.7. Métodos Heurísticos para solucionar Problema de Roteamento de Veículos.

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*).

A utilização de diferentes metaheurísticas para o problema de roteamento de veículos podem ser encontradas em Gendreau, Laporte e Potvin (1994) e Hjorring(1995).

- Heurística: é a instanciação de uma metaheurística, ou seja, a aplicação da mesma em um problema específico de otimização. Por exemplo, em Dorigo et. al (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 et. al (1999) a flexibilizou para tal.

A princípio, os métodos aproximativos e os métodos heurísticos podem parecer iguais, o que não é fato. Conforme Andrade et. al. (2004), 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 extremamente rápido quando comparadas aos métodos aproximativos, fato peculiar que fez com que esse método tivesse uma grande disseminação nos últimos anos.

2.7.1. Heurística Push Forward Insertion Heuristic (PFIH)

Segundo Thangiah et al. (1994), o método *Push-Forward Insertion* para inserção de consumidores nas rotas do PRVJT, que foi introduzido por Solomon (1987), é um eficiente método para calcular o custo da adição de um novo consumidor na rota corrente. Assumindo uma rota $R_p = \{C_1, \dots, C_m\}$ onde C_1 é o primeiro consumidor e C_m é o último, com seus tempos de chegada e partida definidos como e_1, l_1 e e_m, l_m respectivamente. A viabilidade de inserir um cliente na rota R_p é verificada pela adição do consumidor entre todas as arestas da rota e selecionando a aresta que possui o menor custo de viagem. Para um consumidor C_i ser inserido entre C_0 e o C_1 , a viabilidade da inserção é verificada calculando a quantidade de tempo que a chegada em t_1 é aumentada. Uma mudança no tempo de chegada t_1 pode afetar o tempo de chegada de todos os consumidores sucessores de C_1 na rota corrente. Portanto, a viabilidade de inserção de C_i precisa ser computada sequencialmente checando os valores do *Push-Forward* de todos os consumidores C_j de C_i . O valor *Push-Forward* para um consumidor C_j é 0 se o tempo propagado pelo predecessor de C_j , pela inserção de C_i na rota, não afetar o tempo de chegada t_j . A checagem seqüencial para a viabilidade prossegue até que o valor do *Push-Forward* do consumidor é 0 ou um consumidor é “empurrado” sendo atrasado o seu atendimento. No pior caso, todos os consumidores são checados pela viabilidade.

O *Push-Forward Insertion Heuristic (PFIH)* começa uma nova rota pela seleção de um consumidor inicial e então outros vão sendo inseridos na rota corrente até que a capacidade do veículo é excedida ou não há tempo viável para inserir um outro consumidor na rota atual. O custo para selecionar o primeiro consumidor C_i é calculado usando a fórmula abaixo:

$$\text{Custo de } C_i = -\alpha d_{0i} + \beta l_i + \gamma ((p_i/360)d_{0i}) \quad (2.14)$$

Onde:

d_{0i} = distância do depósito central ao consumidor i ;

l_i = limite superior (janela de tempo) para chegada do veículo ao consumidor i ;

p_i = ângulo da coordenada polar do consumidor i , referente ao depósito central.

Os consumidores como menor custo são selecionados assim como o primeiro, para serem visitados. Os pesos para os três critérios foram derivados empiricamente e são $\alpha = 0.7$, $\beta = 0.1$ e $\gamma = 0.2$.

A prioridade na regra em (2.14) para a seleção de um consumidor depende a distância, do ângulo da coordenada polar e o limite superior. O ângulo da coordenada polar do consumidor com o depósito em (2.14) é normalizado em termos da distância. Esta normalização permite comparações de distância, do último prazo para entrega e o valor do ângulo do consumidor em termos de uma unidade comum.

Uma vez que o primeiro consumidor é selecionado para a rota, a heurística seleciona de um conjunto de consumidores sem rota aquele j^* que minimiza o custo total de inserção entre todas as arestas $\{k, l\}$ da rota sem violar as restrições de tempo e capacidade. O consumidor j^* é inserido na posição de menor custo entre $\{k^*, l^*\}$ na rota e o processo de seleção é repetido até não conseguir inserir mais consumidores. Neste estágio, uma nova rota é criada e o processo é repetido até todos consumidores serem roteados. Assumindo que há um número ilimitado de veículos, K , que é amplo e determinado pela heurística para rotear todos consumidores.

2.7.2. Colônia de Formigas

A metaheurística da colônia de formigas proposta por Dorigo e Caro(1999), que é um desenvolvimento natural dos algoritmos de formigas propostos por Dorigo (1992) para a solução de problemas de otimização combinatória, como o problema do caixeiro viajante.

As formigas são insetos sociais que possuem um sistema complexo de organização e divisão de tarefas, cuja principal função é garantir a sobrevivência do formigueiro. Chama atenção a complexidade das tarefas executadas pelo formigueiro quando comparada com a simplicidade de cada formiga individualmente. A metaheurística da colônia de formigas foi inspirada na observação das colônias de formigas reais, em particular em como elas encontram o menor caminho entre a fonte de alimentos e o formigueiro.

Para a obtenção do alimento para o formigueiro, a colônia resolve um interessante problema de otimização. Inicialmente, as formigas percorrem de modo aleatório a região próxima ao formigueiro em busca do alimento. Cada formiga, enquanto percorre o seu caminho, deposita sobre o solo uma substância chamada feromônio, formando um caminho ou rastro de feromônio. As formigas subseqüentes detectam a presença desta substância e tendem a escolher o caminho marcado com a maior concentração de feromônio.

O feromônio portanto, além de possibilitar a formação de um caminho de volta para a formiga, também tem a função de informar as outras formigas sobre quais os melhores caminhos até o alimento. Depois de algum tempo, os caminhos mais eficientes – ou de menor distância percorrida até o alimento – acumulam uma quantidade maior de feromônio.

Inversamente, os caminhos menos eficientes – ou de maior distância percorrida até o alimento – apresentam uma pequena concentração de feromônio, devido ao menor número de formigas que passaram por ele e ao processo de evaporação natural do feromônio.

No problema de otimização que o formigueiro se defronta, cada formiga é capaz de construir uma solução completa do problema; contudo, a melhor solução só é obtida mediante cruzamento das diversas soluções encontradas.

2.7.3. Recozimento Simulado

Os algoritmos de Recozimento Simulado surgiram em 1983, tendo como precursor Kirkpatrick (1983) que se baseou nas idéias de Metrópolis (METROPOLIS et. al.,1953). Metrópolis introduziu um método numérico simples que representa o estado de um conjunto de átomos em equilíbrio a uma dada temperatura, esse método usa analogia com o processo de recozimento (*annealing*) da metalurgia.

Nesse processo eleva-se o metal a altas temperaturas, forçando os átomos a vibrarem violentamente, sendo em seguida esfriado gradualmente para que os átomos atinjam padrões estáveis. Da formulação matemática desse processo originou-se o algoritmo de simulação computacional Recozimento Simulado.

O Recozimento Simulado é um método estocástico que visa à minimização da função custo. Esse algoritmo não requer o uso de derivadas da função custo e não é afetado por descontinuidades ou não-linearidades.

O processo computacional começa gerando um conjunto de números aleatórios para as variáveis de projeto, dentro do espaço de busca, elevados a altas temperaturas. Logo após, verifica-se os respectivos valores da função custo (nível de energia), onde os melhores valores definem o centro das iterações para a próxima temperatura. Realizado esse passo, se reduz à temperatura e repete-se o processo até que o critério de parada ou de convergência seja satisfeito. No recozimento simulado a energia representa a função custo e a temperatura é um parâmetro de controle da mesma.

Dentre os trabalhos utilizando o algoritmo de recozimento simulado para a solução de problemas de roteamento estão Chiang e Russell (1996), Thangiah et. al. (1994), Czech e Czarnas (2002) e Li et. al. (2003).

2.7.4. Busca TABU

A busca tabu (*Tabu Search*) é um procedimento heurístico proposto por Fred Glover, (GLOVER, 1986) para resolver problemas de otimização combinatória. A idéia básica é evitar que a busca por soluções ótimas termine ao encontrar um mínimo local.

Este tipo de algoritmo faz uma busca agressiva no espaço de soluções do problema de otimização com o intuito de obter sempre as melhores alternativas que não sejam considerados tabu. A heurística busca tabu algumas vezes aceita a solução considerada tabu, baseado no critério de aspiração que determina quando as restrições tabu podem ser ignorados.

Vários exemplos de aplicação de Busca Tabu em problemas de roteamento (PRV) podem ser encontrados na literatura. Geandreau, Hertz e Laporte (1994) tratam do problema de roteamento com restrições capacidade imposta aos veículos. Willard (1989); Pureza e França (1991); Osman(1993) e Taillard(1994) consideram restrições de tempo e capacidade. Potvin, Kervahut, Garcia e Rousseau(1993) utilizam Busca Tabu para o PRV com Janelas de Tempo (*Time Windows*).

2.7.5. Algoritmo Genético

O Algoritmo Genético (AG) foi desenvolvido por Holland (1975), com intuito de aplicar a teoria da evolução das espécies elaborada por Darwin (1859), ou seja, utilizar os conceitos da evolução biológica, tais como genes, cromossomos, cruzamento, mutação e seleção, em outros problemas de otimização através de algoritmos computacionais.

Segundo Bräysy (2001) é uma heurística de método de busca adaptativa baseada na genética de populações. A criação de uma nova geração de indivíduos envolve primeiramente quatro passos principais: representação, seleção, recombinação e mutação.

Para o melhor entendimento sobre o algoritmo, alguns conceitos devem ser definidos:

- cromossomo (genótipo) - cadeia de bits que representa uma solução possível para o problema;
- gene - representação de cada parâmetro de acordo com o alfabeto utilizado (binário, inteiro ou real);
- fenótipo - cromossomo codificado;
- população - conjunto de pontos (indivíduos) no Espaço de Busca ;
- geração - iteração completa do AG que gera uma nova população;
- aptidão bruta - saída gerada pela função objetivo para um indivíduo da população;
- aptidão normalizada - aptidão bruta normalizada, entrada para o algoritmo de seleção;
- aptidão máxima - melhor indivíduo da população corrente;
- aptidão média - aptidão média da população corrente.

Segundo Mitchell (1996), deve ser observado que cada cromossomo, chamado de indivíduo no AG, corresponde a um ponto no espaço de soluções do problema de otimização. O processo de solução adotado nos algoritmos genéticos consiste em gerar,

através de regras específicas, um grande número de indivíduos, população, de forma a promover uma varredura tão extensa quanto necessária do espaço de soluções.

As operações básicas de um AG simples são mostradas na figura abaixo:

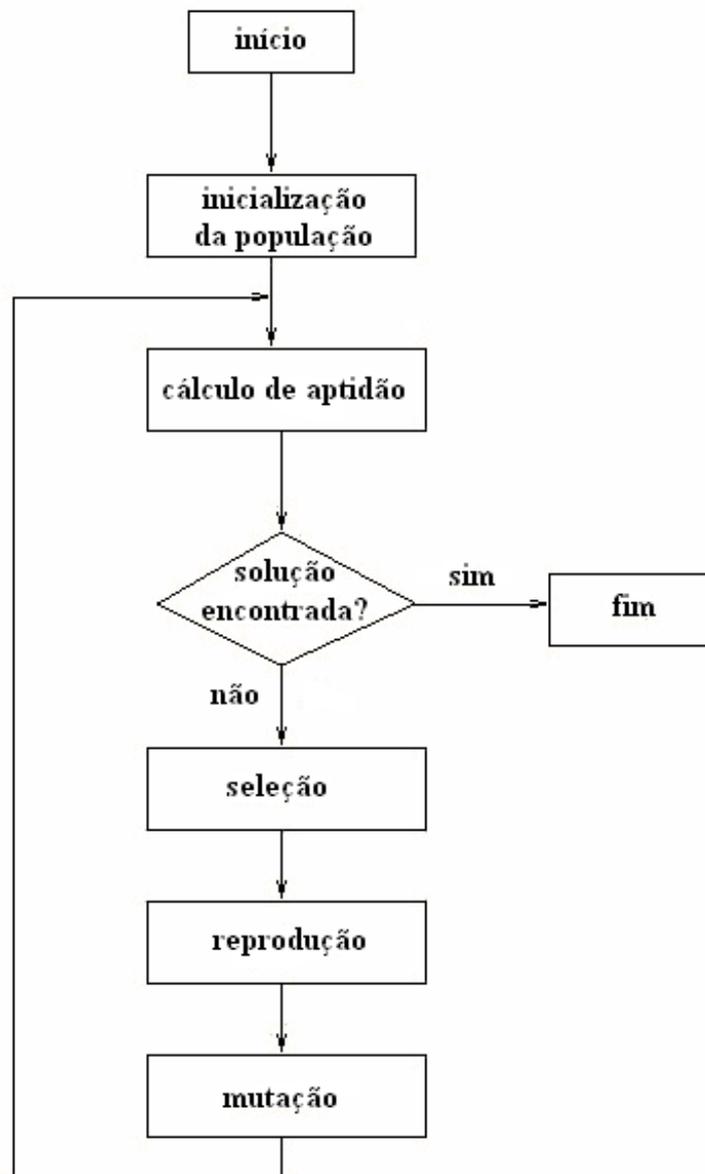


Figura 2.2: Estrutura Básica de um AG Simples. Fonte: Miranda (2007)

Com referência ao diagrama da Figura 2.2, observa-se que cada iteração do algoritmo genético corresponde à aplicação de um conjunto de quatro operações básicas: cálculo de aptidão, seleção, cruzamento e mutação. Ao fim destas operações cria-se uma nova população, chamada de geração que, espera-se, representa uma melhor aproximação da solução do problema de otimização que a população anterior. A população inicial é

gerada atribuindo-se aleatoriamente valores aos genes de cada cromossomo. A aptidão bruta de um indivíduo da população é medida por uma função de erro, também chamada de função objetivo do problema de otimização. A aptidão bruta é em seguida normalizada (aptidão normalizada), para permitir um melhor controle do processo de seleção. Como critérios de parada do algoritmo em geral são usados a aptidão do melhor indivíduo em conjunto com a limitação do número de gerações. Outros critérios podem envolver, por exemplo, um erro abaixo de um valor especificado pelo projetista para um determinado parâmetro do problema.

Cada uma das quatro operações básicas e a inicialização do Algoritmo Genético estão descritas abaixo:

- Inicialização: uma população de n indivíduos é gerada aleatoriamente. Cada um dos indivíduos da população representa uma possível solução para o problema, ou seja, um ponto no espaço de soluções.
- Cálculo da Aptidão: geralmente a aptidão do indivíduo é determinada através do cálculo da função objetivo, que depende das especificações de projeto. Nesta fase os indivíduos são ordenados conforme a sua aptidão.
- Seleção: Nesta fase os indivíduos mais aptos da geração atual são selecionados. Esses indivíduos são utilizados para gerar uma nova população por cruzamento. Cada indivíduo tem uma probabilidade de ser selecionado proporcional à sua aptidão.
- Cruzamentos (*Crossover*): os indivíduos selecionados na etapa anterior são cruzados da seguinte forma: a lista de indivíduos selecionados é embaralhada aleatoriamente criando-se, desta forma, uma segunda lista, chamada lista de parceiros. Cada indivíduo selecionado é então cruzado com o indivíduo que ocupa a mesma posição na lista de parceiros. A forma como se realiza este cruzamento é ilustrada na figura 2.3. Os cromossomos de cada par de indivíduos a serem cruzados são particionados em um ponto, chamado ponto de corte, sorteado aleatoriamente. Um novo cromossomo é gerado permutando-se a metade inicial de um cromossomo com a metade final do outro. Deve-se notar que se o cromossomo for representado por uma cadeia de bits, como na figura 2.3, o ponto de corte pode incidir em qualquer posição (bit) no interior de um gene, não importando os limites do gene. No caso de genes representados por

números reais, a menor unidade do cromossomo que pode ser permutada é o gene.

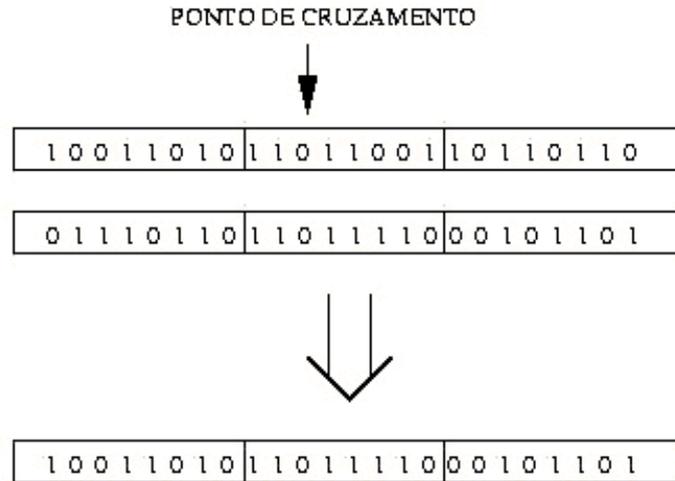


Figura 2.3: Cruzamento de dois indivíduos num AG Simples. Fonte: Miranda (2007)

- **Mutação:** a operação de mutação é utilizada para garantir uma maior varredura do espaço de estados e evitar que o algoritmo genético convirja muito cedo para mínimos locais. A mutação é efetuada alterando-se o valor de um gene de um indivíduo sorteado aleatoriamente com uma determinada probabilidade, denominada probabilidade de mutação, ou seja, vários indivíduos da nova população podem ter um de seus genes alterado aleatoriamente.

Alguns trabalhos utilizando Algoritmos Genéticos podem ser encontrados em Thangiah et. al. (1991), Alvarenga e Mateus (2004), Homberger e Gehring(1999) e Berger (2001)

2.8. Conjunto de Problemas Testes de Solomon

O conjunto de problemas teste proposto por Solomon em 1987 (SOLOMON,1987), baseado em dados de alguns problemas usados por Christofides et al. (1979) para o problema de roteamento padrão, trata-se de diferentes classes de instâncias, cada qual com características geográficas e de restrições características. Os consumidores estão distribuídos em um plano XY de dimensões 100x100.

As instâncias são divididas em 6 grupos denominados R1, R2, C1, C2, RC1 e RC2. Cada classe contém entre 8 e 12 instâncias. Os grupos R1 e R2, possuem uma disposição geográfica dos clientes de forma aleatória, já nos grupos C1 e C2, a disposição é na forma de agrupamentos, e nos grupos RC1 e RC2 são tipos mistos (parte em agrupamentos e parte aleatória). Nas classes R1, C1 e RC1, as janelas de tempo e o horizonte total são curtos, diminuindo assim o número de consumidores por rota. Já as classes R2, C2 e RC2, possuem um longo horizonte total fazendo com que as rotas tenham mais consumidores viáveis.

Cada instância tem 100 consumidores, mas pode-se considerar apenas os primeiros 25 ou 50 consumidores dependendo do caso.

No caso do uso das instâncias de Solomon, nos problemas de roteamento dinâmico, adaptações devem ocorrer para que não se tenha conhecimento de todos os consumidores no início do roteamento, o que tornaria o problema estático.

Uma extensão das instâncias de Solomon, foi proposta por Homberger (2000), em que um conjunto de novas instâncias com 200, 400, 600, 800 e 1000 consumidores é apresentado.

As figuras abaixo, representam os três tipos de classe C (Fig.2.4), R (Fig.2.5) e RC(Fig.2.6):

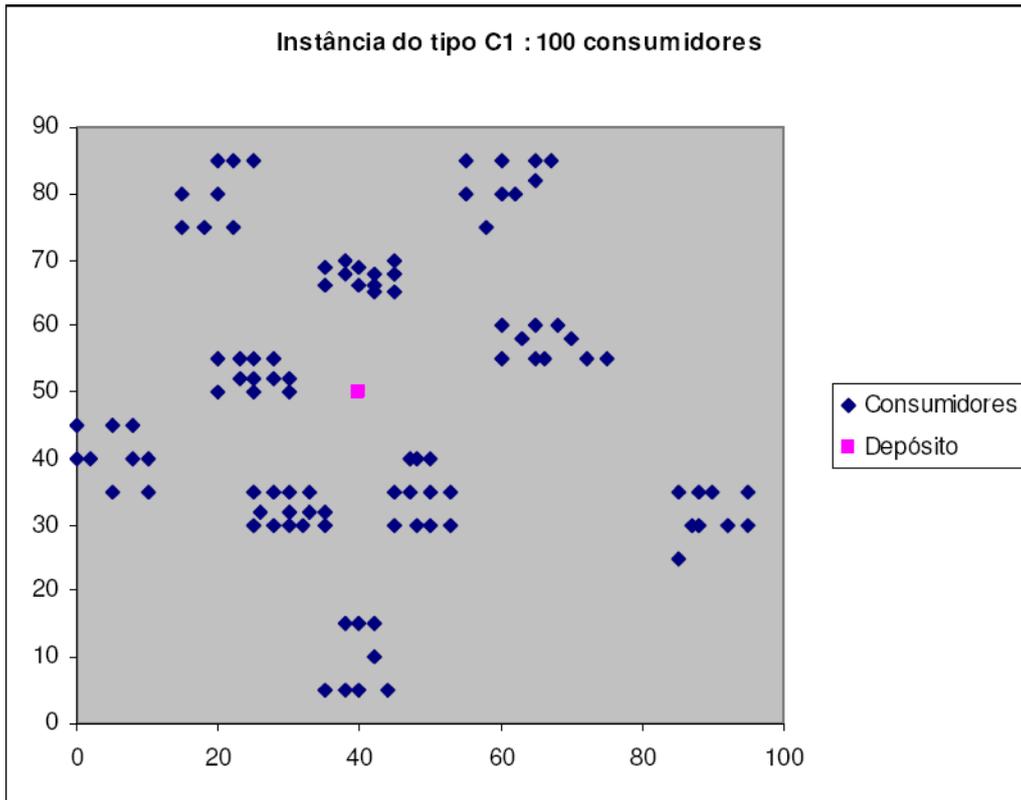


Figura 2.4: Disposição espacial dos consumidores nos problemas das classes C1 e C2.

Fonte: Alvarenga (2005)

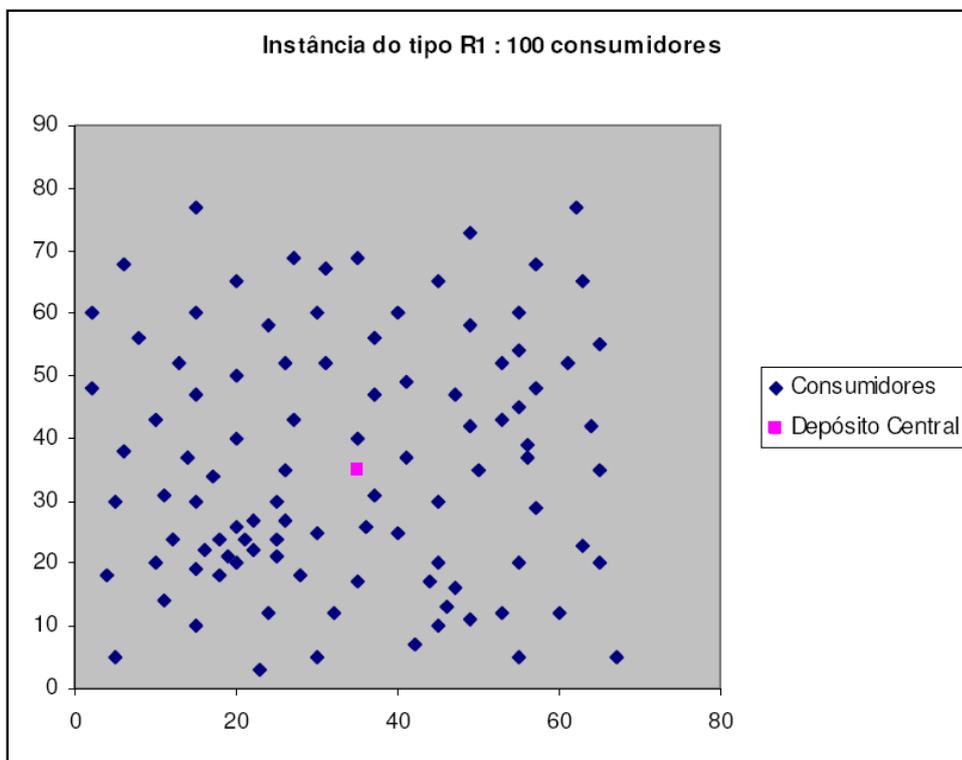


Figura 2.5: Disposição espacial dos consumidores nos problemas das classes R1 e R2.

Fonte: Alvarenga (2005)

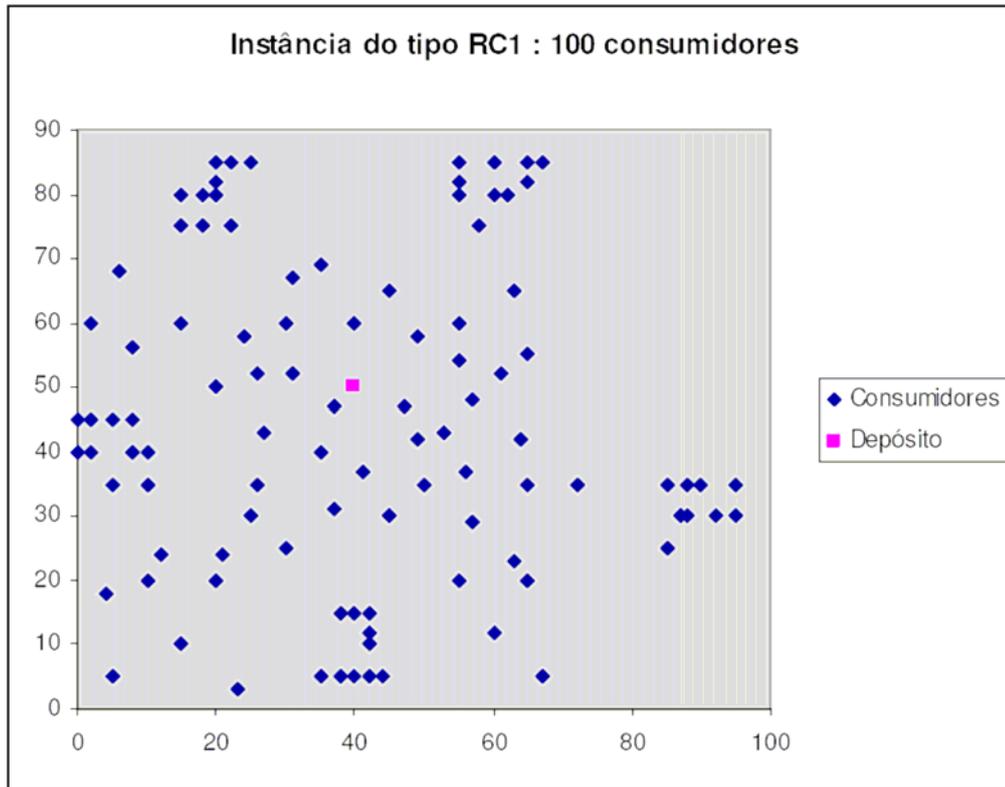


Figura 2.6: Disposição espacial dos consumidores nos problemas das classes RC1 e RC2.

Fonte: Alvarenga (2005)

3. PROPOSTA DESTE TRABALHO

3.1. Algoritmo Evolutivo

Este trabalho tem como objetivo propor uma solução para o Problema de Roteamento de Veículos Dinâmico com Janela de Tempo, tratando tanto da adição como da remoção de pedidos. Esses pedidos podem ser tanto de coleta como de entrega de mercadorias.

A proposta tem como base o Algoritmo Evolutivo, que pode ser representado da seguinte maneira:

Algoritmo Evolutivo Simplificado

- 1: $P \leftarrow$ População Inicial;
- 2: **enquanto** condição não satisfeita **faça**
- 3: $P \leftarrow$ Seleção (P);
- 4: $P \leftarrow$ Mutações (P);
- 5: **fim enquanto**
- 6: Solução \leftarrow Melhor Indivíduo (P)

Observação: Pode-se notar que no Algoritmo Evolutivo não há o uso de operações de cruzamento, sendo por isso, diferente do Algoritmo Genético, descrito anteriormente.

3.2. Função Objetivo

Diferentemente da maioria das heurísticas para o Problema de Roteamento de Veículos, a função objetivo desta proposta não é a redução dos veículos, e sim a minimização da distância total, ficando a redução de veículos e o tempo total gasto, como critérios secundários.

Nas instâncias de Solomon, é estabelecido que o tempo para percorrer uma unidade do espaço euclidiano é igual a uma unidade de tempo, por isso, a diferença entre a distância total percorrida e o tempo total gasto, em números, será dada através da espera da abertura de alguma janela de tempo, quando esta existir, e também pelo tempo de serviço em cada cliente.

3.3. Cálculo da Distância e Janela de Tempo

Antes da execução do Algoritmo Evolutivo, é calculada a distância entre todos os clientes conhecidos *a-priori*. Essa informação é armazenada em uma matriz.

No caso de dois clientes terem janelas de tempo incompatíveis, esses não poderão ser incluídos na mesma rota. Entende-se por janelas de tempo incompatíveis, quando no par de consumidores em questão, a soma da abertura da janela de tempo do primeiro com o tempo de serviço e com o tempo de trajeto para o segundo cliente, for maior que o término da janela de tempo do segundo cliente.

Como o problema é dinâmico, novos consumidores podem ser acrescentados ou removidos nessa matriz construída.

3.4. Representação dos Indivíduos

O depósito é representado pelos números 0 e $[(\text{número de clientes}) + 1]$, já os clientes são representados pelos números inteiros contidos nesse intervalo citado acima.

A representação de um veículo é um vetor com os clientes incluídos na rota deste, mais a distância total a ser percorrida para atender tais clientes.



Figura 3.1: Representação de um veículo com a sua rota

Um indivíduo é formado por um vetor de veículos, ou seja, pelo conjunto das rotas necessárias para uma solução do PRVDJT. Chamado, por isso, de *indivíduo solução* do problema.

Veículo 1:	1	9	14	19	11	15	180	Distância Percorrida por Veículo
Veículo 2:	4	7	13	5	24	2	250	
Veículo 3:	8	3	16	20	23	17	200	
Veículo 4:	6	10	12	18	21	22	210	
Clientes							840	Distância Total Percorrida

Figura 3.2: Representação de uma Solução Indivíduo

Já a população pode ser considerada um vetor de indivíduos, ou seja um conjunto de soluções do problema.

3.5. População Inicial

Para a geração da população inicial necessária para a execução do Algoritmo Evolutivo é usada a heurística *PFIH* (*Push-Forward Insertion Heuristic*) original e mais duas modificações dessa heurística.

Essas modificações são necessárias pelo motivo da heurística original ser determinística, ou seja, para uma dada instância, toda população encontrada será idêntica. Sendo assim, contraria a necessidade do Algoritmo Evolutivo de conter em sua população inicial uma boa diversidade para encontrar soluções viáveis e de boa qualidade para o PRVDJT.

A primeira modificação é a *PFIH* Estocástica, em que não é utilizada a equação (2.14). Ou seja, os consumidores são escolhidos aleatoriamente, e o custo de cada inserção sendo relativa somente ao aumento da distância total percorrida na rota a ser construída.

A segunda modificação é um *PFIH* com uma variação sobre α , β e γ , que na equação (2.14) são constantes. Essa variação é dada através de valores aleatórios no intervalo de 0 a 1, para α , β e γ .

O *PFIH* que possui maior probabilidade de gerar a maior parte da população é o que possui uma variação sobre α , β e γ . Enquanto o *PFIH* original e o estocástico possuem probabilidades menores, tendo uma contribuição menor sobre a população gerada.

3.6. Cálculo de Aptidão

Depois de gerada uma população inicial, deve-se avaliar cada indivíduo, em busca dos mais promissores.

Para calcular a aptidão de cada indivíduo utiliza-se a distância total percorrida pelos seus veículos. Ou seja, aqueles que possuem uma distância total menor são considerados os mais aptos, e por isso com maiores chances de continuarem nas próximas etapas do algoritmo.

3.7. Seleção

Após a avaliação, os indivíduos passam por uma fase de seleção, em que os melhores indivíduos possuem uma maior tendência de serem selecionados.

Nessa proposta, duas estratégias de seleção são utilizadas: Torneio Múltiplo e Roleta Giratória.

No torneio múltiplo, um grupo de M indivíduos é selecionado aleatoriamente da população. Esses indivíduos são comparados através de suas adaptabilidades, vencendo aquele que apresentar o melhor valor de aptidão.

Já na roleta giratória, os indivíduos recebem fatias da roleta proporcionais ao valor do seu grau de aptidão. Um giro na roleta seleciona uma dessas fatias. Os indivíduos mais aptos tendem a ser frequentemente selecionados.

Seleção por Roleta Giratória

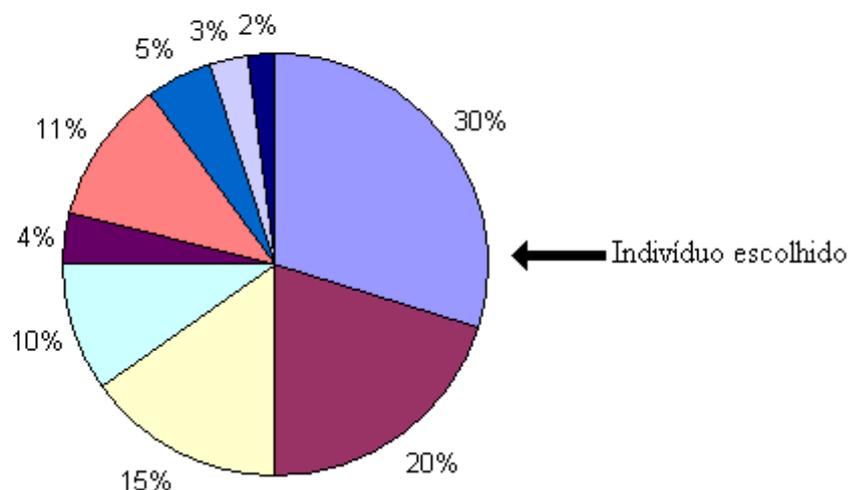


Figura 3.3: Representação de uma Roleta Giratória utilizada na Seleção de Indivíduos

Devem-se usar ambas as estratégias, a fim de diminuir a convergência para mínimos locais. Diminuindo assim, a chance de ocorrer uma variabilidade pobre da população, que tornaria os indivíduos muito parecidos no decorrer das gerações.

3.7.1. Elitismo

Para não perder a melhor solução encontrada, utiliza-se o processo de elitismo, ou seja, a cópia integral dessa solução na geração sucessora. Como a melhor solução ainda não passou pela fase das mutações do Algoritmo Evolutivo, ela ainda pode ser perdida por alterações causadas pelos conjuntos de operadores dessa fase. Para não ter a possibilidade da perda da melhor solução, duas cópias são armazenadas. Na primeira cópia é possível realizar o conjunto de operações existente na fase de mutação, para que uma possível melhora possa ser testada. A segunda cópia é conservada mesmo depois da fase de mutação, como garantia de sua existência na próxima geração.

3.8. Mutações

As mutações são essenciais para o Algoritmo Evolutivo, pois é através delas que se obtém uma diversificação e uma intensificação no espaço de busca.

Para a proposta deste trabalho, são especificados os tipos de mutação utilizados em Alvarenga (2005):

1- Mutação de Eliminação de Rotas: onde uma rota é escolhida aleatoriamente e todos os seus consumidores são inseridos em outras rotas. Cada consumidor, a partir do primeiro, é inserido na primeira posição viável de alguma das outras rotas. Se algum consumidor não consegue mudar de rota, continua na rota de origem. A mutação termina quando todos os consumidores da rota escolhida foram testados.

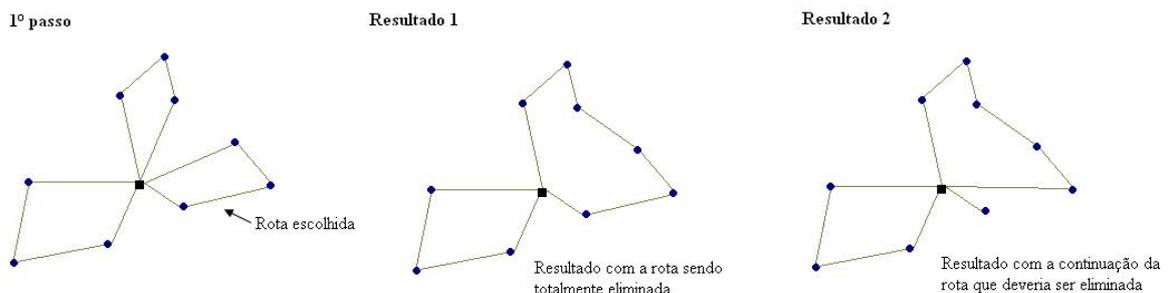


Figura 3.4: Diagrama esquemático da Mutação de Eliminação de Rotas

2- Mutação de Retirada de Consumidor: mutação totalmente aleatória com o objetivo de aumentar a diversificação. Um consumidor é escolhido aleatoriamente na tentativa de encaixá-lo em uma outra rota. O veículo destino também é escolhido aleatoriamente. A posição de inserção no novo veículo é definida pelo menor custo adicional obtido naquele veículo, embora a solução como um todo possa ter piorado. Se o veículo escolhido não suportá-lo, outro veículo é sorteado até esgotar todos os veículos possíveis. O operador falha, se nenhum veículo for capaz de receber o consumidor em questão.

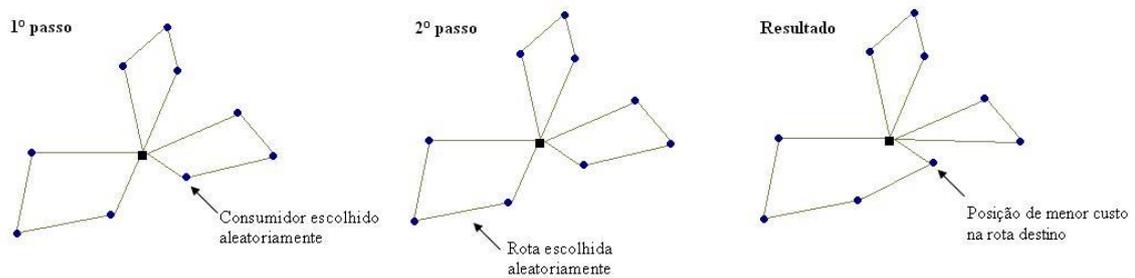


Figura 3.5: Diagrama esquemático da Mutação de Retirada de Consumidor

3- Mutação de Inserção de Consumidor: nesse operador, uma rota escolhida aleatoriamente (rota receptora) irá receber um consumidor. Depois que é definida a rota receptora, todos os consumidores de outras rotas são testados um a um, em todas as posições possíveis da rota escolhida. O consumidor que resultar na menor alteração da distância percorrida na rota receptora será retirado de sua rota de origem e inserido na melhor posição da rota receptora.

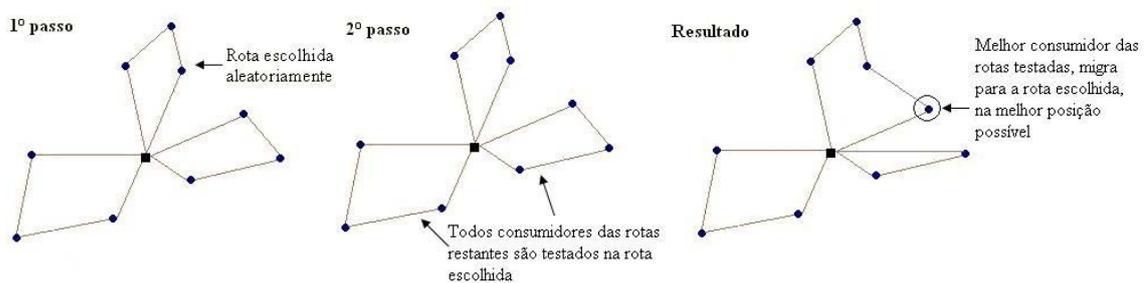


Figura 3.6: Diagrama esquemático da Mutação de Inserção de Consumidor

4- Mutação por Troca Simples de Consumidores em Rotas Distintas: operador que realiza uma troca aleatória de dois consumidores de duas rotas diferentes, em que

ocorrendo a troca cada um assume a posição que era do seu substituto. A troca só é realizada se a solução encontrada for viável.

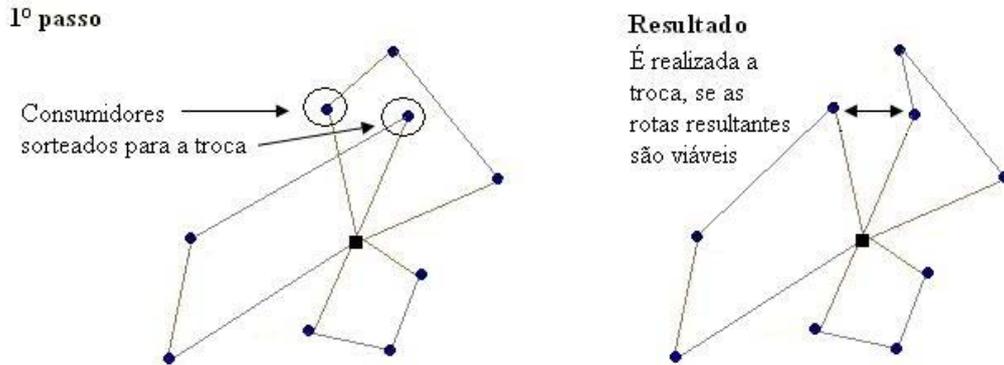


Figura 3.7: Diagrama esquemático Mutação por Troca Simples de Consumidores em Rotas Distintas

5- Mutação por Troca de Consumidores em Rotas Distintas: operador que realiza uma troca aleatória de dois consumidores de duas rotas diferentes, em que ocorrendo a troca cada um não precisa assumir necessariamente a posição que era do seu substituto, e sim a melhor posição nas rotas destino. A troca só é realizada se houver viabilidade dos consumidores no destino.

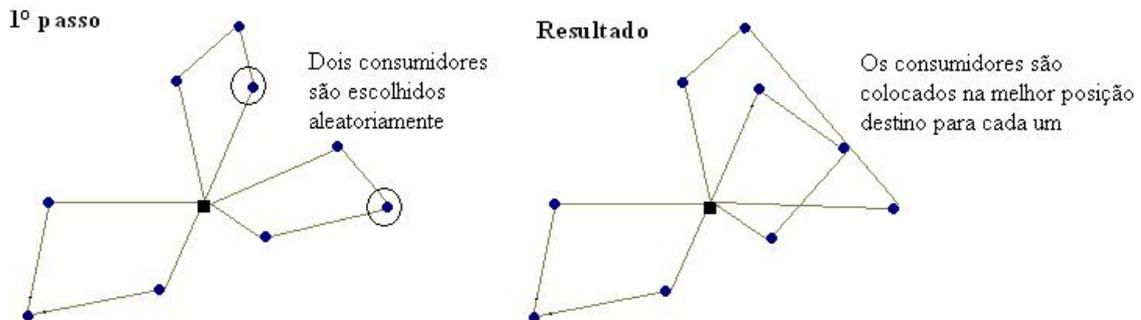


Figura 3.8: Diagrama esquemático da Mutação por Troca de Consumidores em Rotas Distintas

6- Mutação λ -interchange: dada uma solução representada por $s = \{r_1, \dots, r_p, \dots, r_q, \dots, r_k\}$ onde r_p é um conjunto de consumidores servidos por um veículo p , uma operação λ -interchange ocorre entre um par de rotas. A operação λ -interchange entre r_p e r_q é uma troca do conjunto de consumidores r'_p que está contido ou é igual a r_p de dimensão $|r'_p| =$

λ , por outro r'_q que está contido ou é igual a r_q de dimensão $|r'_q| = \lambda$, entre as rotas r_p e r_q . A operação só é realizada se houver uma redução da distância total.

7- Mutação por Inversão de Consumidores: operador que inverte a ordem de atendimento de uma seqüência de consumidores de uma rota.

8- Mutação por Troca Simples de Consumidores com Redução da Distância Total Percorrida: operador que realiza uma troca aleatória de dois consumidores de duas rotas diferentes, em que ocorrendo a troca cada um assume a posição que era do seu substituto. A troca só é realizada se houver um ganho na distância total percorrida global da solução.

9- Mutação por Troca de Consumidores com Redução da Distância Total Percorrida: operador que retira dois consumidores aleatoriamente de duas rotas diferentes. Verifica-se a possibilidade de troca dos consumidores entre as duas rotas, caso seja viável essa troca e ainda houver ganho global na distância percorrida a operação é realizada com sucesso.

10- Mutação *PFIH* parcial: nessa mutação, um veículo é escolhido randomicamente, todos os outros são esvaziados e sobre eles é realizado um dos processos de geração da população, ou seja, o *PFIH* estocástico.

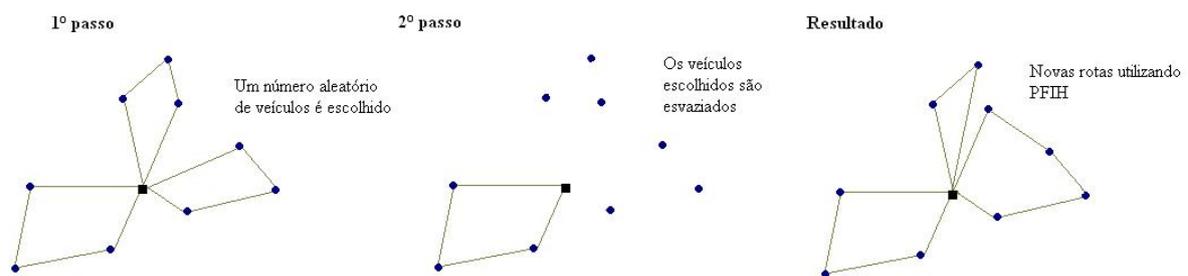


Figura 3.9: Diagrama esquemático da Mutação *PFIH* parcial

11- Mutação por Reinserção de Consumidores com ganho em Distância Total: essa mutação retira um consumidor aleatório por vez de um veículo e testa todas as posições possíveis no mesmo veículo em que o consumidor foi retirado. Se há alguma nova posição que minimiza a distância total, sua posição é atualizada.

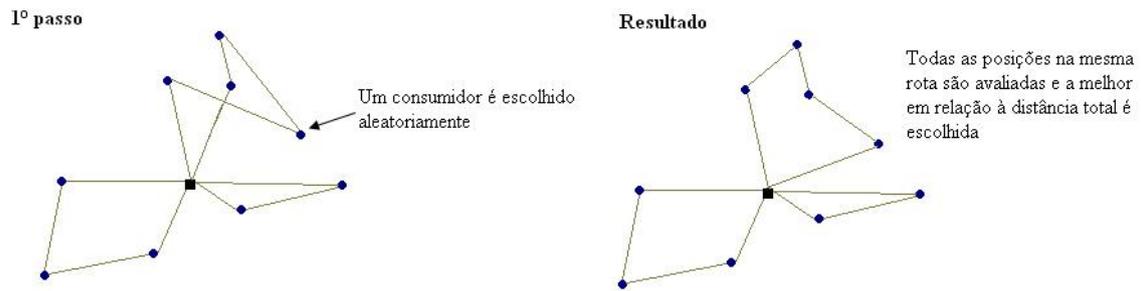


Figura 3.10: Diagrama esquemático da Mutação por Reinscrição de Consumidores com ganho em Distância Total

12- Mutação de Busca Exaustiva por Troca com Ganho de Custo: realiza a busca exaustiva por troca simples de consumidores entre duas rotas distintas. Todas as trocas possíveis entre consumidores das duas rotas são testadas. A cada troca com ganho na distância total percorrida, a seqüência de avaliações é interrompida, reiniciando com as rotas diferentes na próxima execução.

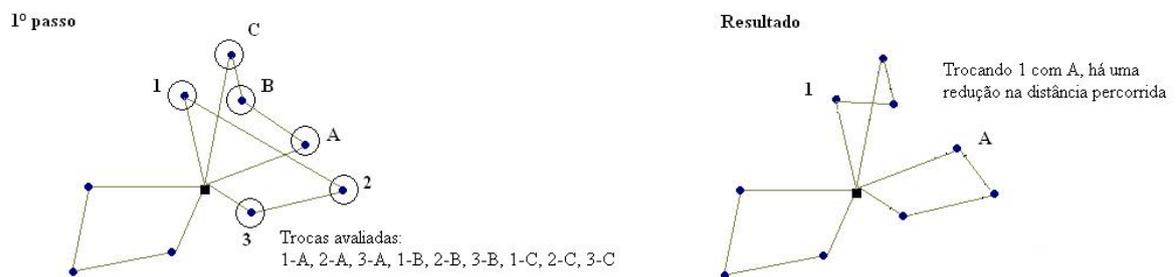


Figura 3.11: Diagrama esquemático da Mutação de Busca Exaustiva por Troca com Ganho de Custo

13- Mutação de Retirada e Troca de Consumidores seguida do *PFIH*: um veículo é escolhido aleatoriamente, uma quantidade randômica de consumidores consecutivos menor que 15 é retirada deste veículo, as partir de um consumidor escolhido aleatoriamente. Em seguida, aplica-se a Mutação de Busca Exaustiva por Troca com Ganho de Custo. Posteriormente os consumidores retirados no começo são reinseridos na solução utilizando o *PFIH*.

3.9. Resultados com Instâncias Estáticas

Todas as etapas, descritas anteriormente nesse capítulo, foram implementadas. O algoritmo implementado pode ser considerado estático, ou seja, sem as modificações necessárias para tratar da dinamicidade proposta, que serão vistas nas seções seguintes.

Para os testes foram utilizadas as instâncias de Solomon. O computador utilizado para a coleta dos resultados foi um Pentium IV, 1,50GHz, com 256MB de RAM, utilizando o sistema operacional *Windows XP Professional*. A implementação foi feita utilizando a linguagem C#.

As tabelas, a seguir, mostram os resultados obtidos com o algoritmo implementado para as 56 instâncias, além de mostrar também os melhores resultados conhecidos na literatura. Os melhores resultados da literatura para o problema estático foram retirados do site: <http://www.sintef.no/static/am/opti/projects/top/vrp/bknown.html>.

Para cada instância, o algoritmo foi executado três vezes e tinha em sua configuração uma população de 50 indivíduos e 5000 gerações.

Tabela 3.1: Resultados obtidos na classe R1

Instância	Distância	Veículos	Melhor resultado na literatura	
			Distância	Veículos
R101	1646,72	20	1645,79	19
R102	1489,12	18	1486,12	17
R103	1227,96	14	1292,68	13
R104	1004,43	11	1007,24	9
R105	1375,68	16	1377,11	14
R106	1255,82	13	1251,98	12
R107	1097,08	12	1104,66	10
R108	963,88	10	960,88	9
R109	1184,93	14	1194,73	11
R110	1126,67	13	1118,59	10
R111	1086,58	12	1096,72	10
R112	1011,21	11	982,14	9

Tabela 3.2: Resultados obtidos na classe R2

Instância	Distância	Veículos	Melhor resultado na literatura	
			Distância	Veículos
R201	1196,09	6	1252,37	4
R202	1047,45	5	1191,7	3
R203	930,79	4	939,54	3

Continuação da Tabela 3.2

R204	793,68	3	825,52	2
R205	981,36	5	994,42	3
R206	914,75	4	906,14	3
R207	863,98	3	890,61	2
R208	730,21	3	726,75	2
R209	888,73	5	909,16	3
R210	929,02	5	939,34	3
R211	768,06	4	892,71	2

Tabela 3.3: Resultados obtidos na classe C1

Instância	Distância	Veículos	Melhor resultado na literatura	
			Distância	Veículos
C101	828,94	10	828,94	10
C102	828,94	10	828,94	10
C103	828,06	10	828,06	10
C104	824,78	10	824,78	10
C105	828,94	10	828,94	10
C106	828,94	10	828,94	10
C107	828,94	10	828,94	10
C108	828,94	10	828,94	10
C109	828,94	10	828,94	10

Tabela 3.4: Resultados obtidos na classe C2

Instância	Distância	Veículos	Melhor resultado na literatura	
			Distância	Veículos
C201	591,56	3	591,56	3
C202	591,56	3	591,56	3
C203	591,17	3	591,17	3
C204	590,6	3	590,6	3
C205	588,88	3	588,88	3
C206	588,49	3	588,49	3
C207	588,29	3	588,29	3
C208	588,32	3	588,32	3

Tabela 3.5: Resultados obtidos na classe RC1

Instância	Distância	Veículos	Melhor resultado na literatura	
			Distância	Veículos
RC101	1669,67	16	1696,94	14
RC102	1518,24	15	1554,75	12
RC103	1303,29	12	1261,67	11
RC104	1167,35	11	1135,48	10
RC105	1561,99	16	1629,44	13
RC106	1440,89	14	1424,73	11
RC107	1297,54	12	1230,48	11
RC108	1203,34	12	1139,82	10

Tabela 3.6: Resultados obtidos na classe RC2

Instância	Distância	Veículos	Melhor resultado na literatura	
			Distância	Veículos
RC201	1338,07	6	1406,91	4
RC202	1154,77	5	1365,64	3
RC203	952,43	5	1049,62	3
RC204	802,98	4	798,41	3
RC205	1192,19	6	1297,19	4
RC206	1110,71	5	1146,32	3
RC207	1011,15	5	1061,14	3
RC208	802,67	4	828,14	3

Na classe R1 e RC1, algumas instâncias conseguiram melhorar a distância em relação aos resultados já conhecidos na literatura, mas não conseguiram chegar ao mesmo número de veículos.

As classes R2 e RC2, apresentaram melhores resultados, mas também não conseguiram chegar ao mesmo número de veículos dos melhores resultados obtidos na literatura.

Já as classes C1 e C2, conseguiram alcançar tanto a distância quanto o número de veículos dos melhores resultados já encontrados anteriormente.

3.10. Adição e Remoção de Clientes

Para tornar o problema dinâmico, depois de cada geração, o algoritmo reserva um intervalo para que novos pedidos possam ser incluídos ou para excluir pedidos cancelados.

Quando o dinamismo refere-se à adição de clientes, utiliza-se o *PFIH* para inserir os novos clientes em cada indivíduo da população, para que esse não seja perdido nas próximas etapas do Algoritmo Evolutivo.

E quando o dinamismo refere-se à remoção de clientes, retira-se o cliente em questão de todas as rotas dos indivíduos da população.

3.11. Coletas e Entregas

Outro ponto que sofre modificações com o dinamismo do problema é o tratamento de coletas e entregas.

As coletas no problema dinâmico são mais fáceis de serem resolvidas, já que um veículo pode coletar mercadorias até a sua capacidade máxima sem precisar voltar ao depósito. O problema maior encontra-se na entrega, já que um novo cliente pode aparecer depois que o veículo já deixou o depósito, o que poderia causar um retorno ao depósito do veículo escolhido para atender tal cliente.

No trabalho proposto, quando um novo cliente é inserido no problema e requer uma coleta, procura-se por um veículo que possa coletar tal mercadoria sem ultrapassar sua capacidade máxima. Se nenhum veículo satisfizer tal condição, um novo veículo vindo do depósito é acrescentando no cenário.

Em caso de um novo cliente acrescentado depois da partida dos veículos do depósito central querendo uma entrega, necessita-se do retorno de algum veículo ao depósito central ou de um novo veículo.

Já no caso da exclusão de um cliente que receberia uma entrega, o veículo passa a ter em seu interior uma mercadoria que apenas está ocupando espaço.

E por último, no caso da exclusão de uma coleta, nada precisaria ser feito em relação ao veículo, já que este passa a ter uma capacidade maior do que a esperada, quando este partiu do depósito.

3.12. Esquema Geral da Proposta

A proposta começa com um cálculo das distâncias de pares de consumidores e compatibilidade de suas janelas de tempo. Logo após essa etapa, começa o Algoritmo Evolutivo.

A primeira parte do algoritmo gera uma população inicial através de 3 tipos da heurística *PFIH*, logo em seguida essa população é avaliada e os melhores indivíduos são escolhidos pela seleção, que é feita por roleta giratória ou por torneio múltiplo, além disso, a melhor solução encontrada é armazenada, para que ela possa prosseguir nas próximas etapas do algoritmo.

Os indivíduos selecionados passam pela etapa de mutação completando um ciclo. Logo após essa etapa, pode-se inserir ou retirar clientes do problema, retornando para a fase de avaliação da população, já com os indivíduos alterados.

O algoritmo termina quando atinge um tempo limite, e como resultado do problema tem-se a melhor solução (indivíduo solução) encontrada até o momento em que se alcança o tempo limite.

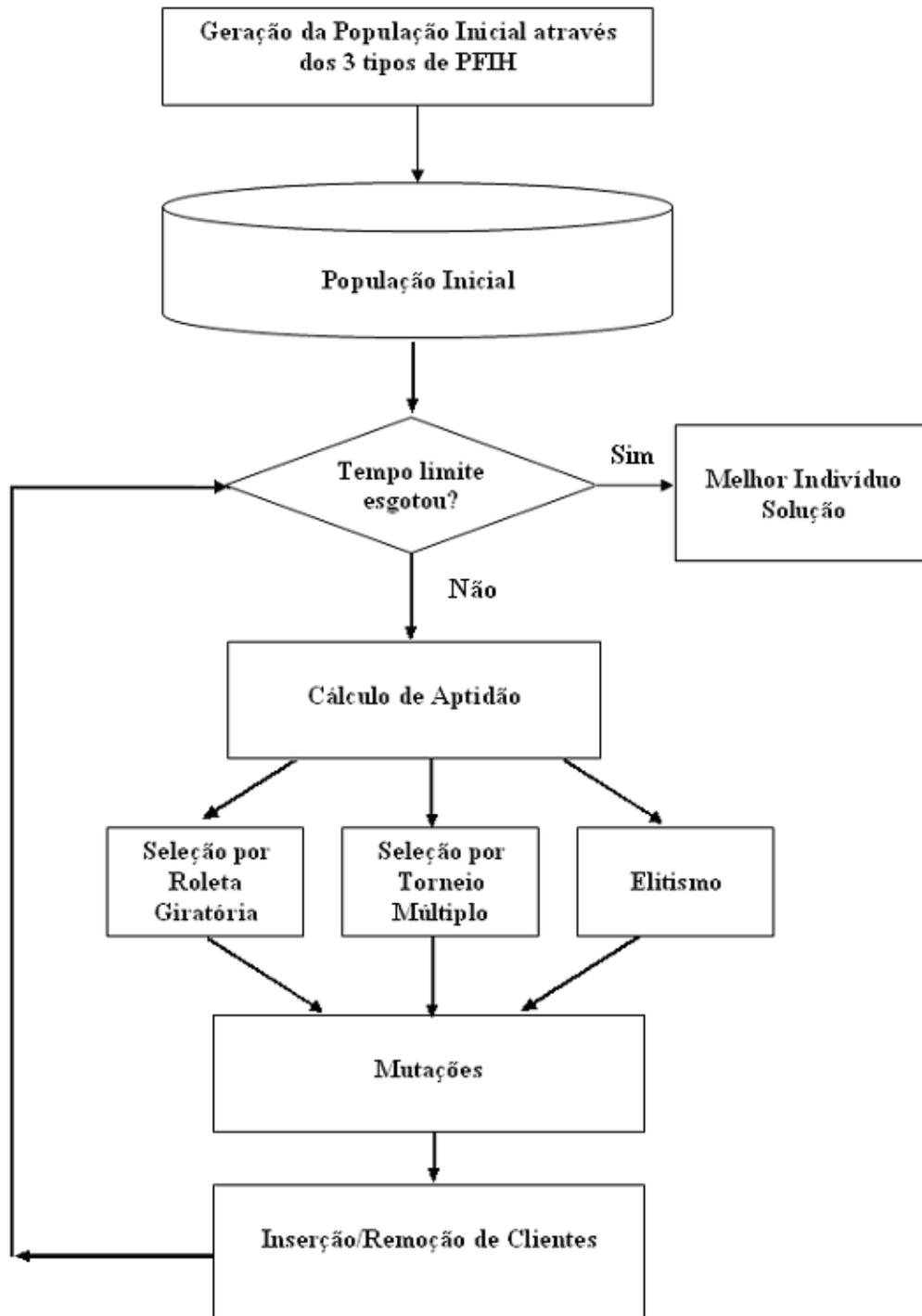


Figura 3.12: Diagrama do Algoritmo Evolutivo proposto

3.13. Instâncias de Solomon Modificadas

As instâncias de Solomon devem ser modificadas para se adaptar ao dinamismo do problema.

Uma alternativa é o uso de parte dos clientes conhecidos de determinada instância e com o decorrer do tempo, acrescentar clientes, até que o número de clientes do problema e da instância original seja igual. Desse modo, pode-se haver uma comparação entre o resultado obtido do problema dinâmico com os resultados já conhecidos do problema estático usando as instâncias de Solomon.

No caso de remover clientes, pode-se usar mais clientes do que uma instância original de Solomon possui e fazer a remoção, até que o problema fique com o número de clientes original.

Já que não existe uma instância consagrada que engloba o PRVDJT, esta continua sendo a melhor maneira de avaliar os resultados obtidos com o algoritmo proposto.

Uma outra maneira de usar as instâncias de Solomon, seria utilizar um número menor de clientes no início e a cada intervalo de tempo, acrescentar ou retirar clientes do problema. A desvantagem dessa alternativa está no fato de não haver uma maneira de comparar tal resultado com o problema estático, a menos que se faça uma alteração no problema estático também.

4. CONCLUSÃO

Os problemas de roteamento de veículos no mundo real apresentam variáveis que não são retratadas no problema estático, tornando tal problema incapaz de responder sobre certas dinamicidades que podem ocorrer durante o transporte das mercadorias.

Por isso, a necessidade de uma solução que possa responder a essas variações em um tempo mínimo é de grande importância.

Uma proposta de solução usando Algoritmo Evolutivo foi apresentada neste trabalho.

Algoritmo Evolutivo Simplificado

- 1: $P \leftarrow$ População Inicial;
- 2: **enquanto** condição não satisfeita **faça**
- 3: $P' \leftarrow$ Seleção (P);
- 4: $P \leftarrow$ Mutações (P);
- 5: **fim enquanto**
- 6: Solução \leftarrow Melhor Indivíduo (P)

O Algoritmo Evolutivo tem como vantagem a sua característica de diversificação da busca por novas regiões, permitindo movimentos de forma mais completa possível para todo o espaço de busca. E também a característica de intensificação nas soluções encontradas.

E por possuir ao longo de suas gerações, soluções de boa qualidade, pode-se fazer um aproveitamento dessas soluções quando novos pedidos são inseridos ou quando há o cancelamento de pedidos, adaptando-se ao problema dinâmico, e retornando soluções de maneira mais rápida.

Espera-se encontrar, na versão dinâmica do Algoritmo Evolutivo, resultados satisfatórios iguais aos encontrados em sua versão estática.

5. TRABALHOS FUTUROS

Para trabalhos futuros poderia ter a implementação da proposta deste trabalho, assim como um maior estudo sobre as operações de mutação e até a criação de novos operadores.

Outras propostas para trabalhos futuros seria a inclusão da etapa de cruzamento, transformando o algoritmo usado, em um Algoritmo Genético. Além da criação de instâncias já elaboradas para o próprio problema dinâmico.

6. REFERENCIAL BIBLIOGRÁFICO

- ALVARENGA, G.B; MATEUS, G.R. *A Two-Phase Genetic and Set Partitioning Approach for the Vehicle Routing Problem with Time Windows*, Fourth International Conference on Hybrid Intelligent Systems (HIS04), IEEE Computer Society Press, 2004.
- ALVARENGA, G.B; *Um Algoritmo Híbrido para Problemas de Roteamento de Veículos Estático e Dinâmico com Janela de Tempo* - Tese de doutorado, Belo Horizonte, UFMG, 2005.
- ANDRADE, C.E; BATISTA, F.L.N; TOSO, R.F. *Modelo de Otimização para Transporte de Cargas em Ambientes Reduzidos*; monografia, Departamento de Ciência da Computação, Universidade Federal de Lavras, 2004.
- ARCHETTI, C.; MANSINI, R.; SPERANZA, M.G. *The Split Delivery Vehicle Routing Problem with Small Capacity*. To Appear in *Transportation Science.*, 2001.
- BACKER, E.K.; SCHAER, J.R. *Solution improvement heuristic for the vehicle routing problem with time window constraints*, 1986.
- BERGER, J.; BARKAOUI, M.; BRÄYSY, O. *A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows*". Working paper, Defense Research Establishment Valcartier, Canada, 2001.
- BERTSIMAS, D.J; VAN RYZIN, G. *A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane*. *Operations Research*, 39(4):601-615, 1991.
- BERTSIMAS, D.J; VAN RYZIN, G. *A Stochastic and Dynamic Vehicle Routing with General Demand and Interarrival Time Distributions*. *Advance Applied Probability*, 25:947-978, 1993.
- BEZERRA, O.B. *Localização De Postos De Coleta Para Apoio Ao Escoamento De Produtos Extrativistas*. Disponível em 07/07/05 em http://www.eps.ufsc.br/disserta/oneida/cap3/cp3__one.htm.
- BIANCHI, L. *Notes on Dynamic Vehicle Routing - The State of Art*. Technical Report, IDSIA-05-01, IDSIA, Switzerland, 2000.
- BRÄYSY, O. *Genetic Algorithms for the Vehicle Routing Problem with Time Windows*. *Arpakannus* 1, 33-38 (special issue on Bioinformatics and Genetic Algorithms), 2001.
- CHEN, H. K.; HSUEH, C. F.; CHANG, M.S. *The Real-Time Time Dependent Vehicle Routing Problem*. *Transportation Research Part E*. To Appear, 2005.
- CHIANG, W.-C; RUSSELL, R.A. *Simulated annealing metaheuristics for the vehicle routing problem with time windows*. *Annals of Operations Research*, 63:3-27, 1996.

- CHRISTOFIDES, N.; MINGOZZI, A; TOTH, P. *The vehicle routing problem*, in: Combinatorial Optimization, John Wiley, 1979.
- CHRISTOFIDES, N; BEASLEY J. *The Period Routing Problem*. Networks, v. 14, p. 237-256, 1984.
- COOK, S.A; *The Complexity of theorem-Proving Procedures*, Symp. on Theory of computing, 1971.
- CORMEN, T.H; LEISERSON, C.E; RIVEST, R.L. *Introduction to algorithms*, The MIT Press, 1999.
- CZECH, Z.J.; CZARNAS, P. *A Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows*, Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, 376-383, 2002.
- DANTZIG G.B; RAMSER J.H. *The Truck Dispatching Problem*. Management Science, Vol.6, 1959.
- DARWIN, C.; *On The Origin of Species*, 1st edition, Harward University Press, MA, 1859.
- DORIGO, M; MANIEZZO, V; COLORNI, A. *The Ant System: Na Autocatalytic Optimizing Process*, Technical Report No. 91-016 Revised, Piltecnico di Milano, Italy, 1991.
- DORIGO, M. Optimization, *Learning and Natural Algorithms*. Ph.D.Thesis, Politecnico di Milano, Italy, 1992.
- DORIGO, M; CARO, G. DI - *The Ant Colony Optimization Meta-Heuristic*. In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization, McGraw-Hill, p.11-32, 1999.
- DORIGO, M; CARO, G.DI; CAMBARDELLA, L.M. *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, 1999.
- DROR, M; LAPORTE, G; TRUDEAU, P. *Vehicle Routing with Split Deliveries*. Discrete Applied Mathematics, v.50, n. 3, p. 229-254, 1994.
- FISHER, M.; JAIKUMAR, R.;WASSENHOVE, V. *A Generalized Assignment Heuristic for Vehicle Routing*. Networks, v.11, p.109-124, 1981.
- FISHER, M. L.. *An Application Oriented Guide to Lagrangian Relaxation*. In Interfaces 15:2 pages10-21. Department of Decision Sciences, The Wharton School,University of Pennsylvania, Philadelphia, April 1985.
- GENDREAU, M; HERTZ, A.;LAPORTE, G. *A Tabu Search Heuristic for the Vehicle Routing Problem*, Management Science, 1994.

- GENDREAU, M.; POTVIN, J-Y. *Dynamic Vehicle Routing and Dispatching*. In: Crainic, T.G. e Laporte (eds) *Fleet Management and Logistics*, p. 115-226. Kluwer, 1998.
- GENDREAU, M.; LAPORTE, G; POTVIN, Y. *Metaheuristics for the Vehicle Routing Problem*, Technical Report, Centre de Recherche sur les transports, Université de Montréal, 1994.
- GENDREAU, M.; GUERTIN, F.; POTVIN, J.-Y; SÉGUIN, R. *Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-Ups and Deliveries*. Technical Report CRT-98-10, Centre de Recherche sur les Transports, Université de Montréal, 1998.
- GENDREAU, M.; GUERTIN, F.; POTVIN, J.-Y; TAILLARD, E. *Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching*. *Transportation Science*, 33(4):381-390, 1999.
- GUNTSCH, M., BRANKE J., HARTMURT S. *ACO Strategies for Dynamic TSP*. In Marco Dorigo, Martin Middendorf, and Stützle, editors, *Abstract Proceedings of ANTS'2000 From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms*, pages 59-62, 2000.
- GUNSTSH, M; MIDDENDORF, M. *Applying Population Based ACO to Dynamic Optimization Problems*. *Anais do ANTS 2002*, p.149-162, 2002.
- HJORRING, C.A. *The Vehicle routing problem and local search metaheuristics*. Phd Thesis. The University of Aucland, 1995
- HOLLAND, J. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.
- HOMBERGER, J.; GEHRING, H. *Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows*. *INFOR*, VOL. 37, 297-318, 1999.
- HOMBERGER, J. *Extended solomon's vrptw instances*. Available on the web at www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm, 2000.
- KAAN A.H.G.R; KOLEN, A.W.J.; TRIENEKENS, W.J.M. *Vehicle Routing Problem with Time Windows*, 1987.
- KILBY, P.; PROSSER, P.; SHAW, P. *Dynamic VRPs: a Study of Scenarios*. Technical Report APES-06-1998, University of Strathclyde, 1998.
- KIRKPATRICK, S; GELATT, C.D.; VECCHI, P.M. *Optimization by Simulated Annealing*, *Science* 220, 671-680, 1983.
- KRUMKE, S.O.; RAMBAU, J.; TORRES, L.M. *Real-time Dispatching of Guided and Unguided Automobile Service Units with Soft Time Windows*. Technical Report 01-22, Konrad-Zuse-Zentrum für Information Stechnik Berlin, 2001.

- LARSEN, A. *The Dynamic Vehicle Routing Problem*. PhD thesis, Technical University of Denmark, Lyngby, Denmark, 2000.
- LARSEN, J. . *Parallelization of the Vehicle Routing Problem with Time Windows*. Ph.D. thesis, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.
- LI, H.; LIM, A.; HUANG, J. *Local Search with Annealing-like Restarts to Solve the VRPTW*, European Journal of Operational Research 150, 115-127,2003.
- MACULAN F, N.; CAMPELLO, R.E.; LOPES, L. B. R. *Relaxação Lagrangeana em Programação Inteira*. In Relatório Técnico ES40-84 COPPE, UFRJ, 1984.
- METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTHM.N.; TELLER, A.H.; TELLER, E. *Equation of State Calculation by Fast Computing Machines*, Journal of Physical Chemistry, 21, 1087-1092, 1953.
- MIRANDA, M.N. *Algoritmos Genéticos: Fundamentos e Aplicações*. Disponível em 11/12/2007 no site: <http://www.gta.ufrj.br/~marcio/genetic.html>.
- MITCHELL M., *An Introduction to Genetic Algorithms*. MIT Press, 1996.
- MONTEMANNI, R.; GAMBARDELLA, L.M.; RIZZOLI, A.E.; DONATI, A.V. *A New Algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System*. Anais do Proceedings of ODYSSEUS 2003. Palermo, Italy, p.26-31, 2003.
- OLIVEIRA, H.C.B. *Algoritmo Evolutivo no Tratamento do Problema de Roteamento de Veículos com Janela de Tempo*, monografia, Universidade Federal de Lavras, 2004.
- OSMAN, I.H. *Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem*, Annals of Operations Research 41, 421-451, 1993.
- PAPADIMITRIOU, C.H; STEIGLITZ, K. *Combinatorial Optimization: Algorithms and Complexity*; USA, Dover Publications Inc., 1982.
- PAPASTAVROU J.D. *A Stochastic and Dynamic Routing Policy Using Branching Processes with State Dependent Immigration*. European Journal of Operational Research, 95:167-177, 1996.
- PASSOS, R.M.; BARROS, E. *Heurísticas para o Vehicle Routing Problem with Time Windows*, 2003.
- POTVIN, J.-Y.; KERVAHUT, T; GARCIA, B.L.; ROUSSEAU, J.-M. *A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows*. Publication CRT-8855, Centre de Recherche sur les Transports, Université de Montreal, 1993.
- POTVIN, J-Y; XU, Y.; BENYAHIA, I. *Vehicle Routing and Scheduling with Dynamic Travel Times*. Computers and Operations Research. To Appear, 2005.

- PSARAFITS, H.N. *Dynamic Vehicle Routing: Status and Prospects*. Operations Research, v. 61, p. 143-164.,1995.
- PUREZA V.M.; FRANÇA P.M. Vehicle Routing via Tabu Search Metaheuristic. Publication CRT-747, Centre of Recherche sur les Transports, Université de Montreal, 1991.
- RIBEIRO, G. M. E LORENA, L. A. N. *Roteamento de veículos dinâmico usando algoritmos genéticos*. XIX ANPET - Congresso de Pesquisa e Ensino em Transportes - Recife /PE - 7 a 11 de novembro de 2005.
- SAVELSBERGH M.; SOL, M. *DRIVE: Dynamic Routing of Independent Vehicles*. Operations Research, 46(4):474-490, 1991.
- SAVELSBERGH M.; SOL, M. *The General Pickup and Delivery Problem*. Transportation Science, pp. 17 -29, 1995.
- SOLOMON, M. *On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time windows constraints*, 1986.
- SOLOMON, M.; *Algorithms for the Vehicle Routing Problem and Scheduling Problem with Time window Constraints*, 1987.
- TAILLARD, E. *Parallel Iterative Search Methods for Vehicle-Routing Problems*. Networks, v.23, p.662-673, 1994.
- THANGIAH, S.R; NYGARD, K.E; JUELL, P.L. *GIDEON: A Genetic algorithm System for vehicle routing with time windows*. In: Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications, p. 322-328, Miami Beach, FL, 1991
- THANGIAH, S.R; OSMAN, I. H.; SUN, T. *Hybrid Genetic Algorithm Simulated Annealing and Tabu Search Methods for Vehicle Routing Problem with Time Windows*, Technical Report 27, Computer Science Department, Slippery Rock University, 1994.
- WILLARD, J.A.G. *Vehicle routing using r-optimal tabu search*, Master's thesis. The Management School. Imperial College. London, 1989.
- XU H., CHEN Z.-L; RAJAGOPAL S.; ARUNAPURAM S. *Solving a Practical Pickup and Delivery Problem*. Manugistics Inc., 2001.