



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada
Mestrado Acadêmico

Cristiano Galafassi

Aplicação de Metaheurísticas na Abordagem do Problema de
Roteamento de Veículos Capacitado com Janelas de Tempo

São Leopoldo, 2011

CRISTIANO GALAFASSI

Aplicação de Metaheurísticas na Abordagem do Problema de Roteamento de Veículos Capacitado com Janelas de Tempo

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – Unisinos.

Área de Atuação: Ciências Exatas e da Terra

Coordenador: Prof. Dr. Cristiano André da Costa

Orientador: Prof. Dr. Arthur Tórgo Gómez

São Leopoldo

2011

Ficha catalográfica

G146a Galafassi, Cristiano

Aplicação de metaheurísticas na abordagem do problema de roteamento de veículos capacitado com janelas de tempo / por Cristiano Galafassi. – 2011.

215 f. : il., 30cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2011.

Orientação: Prof. Dr. Cristiano André da Costa ;
Coorientação: Prof. Dr. Arthur Tórgo Gómez.

1. Metaheurísticas. 2. Algoritmos genéticos. 3. Algoritmo híbrido. 4. Problema de roteamento de veículos. 5. Busca tabu. 6. Hibridização. I. Título.

CDU 004.421

Catálogo na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

CRISTIANO GALAFASSI

Aplicação de Metaheurísticas na Abordagem do Problema de Roteamento de Veículos Capacitado com Janelas de Tempo

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – Unisinos.

Aprovado em ____ de _____ de 2011.

BANCA EXAMINADORA

Prof. Dr. Arthur Tórgo Gómez

Prof. Dr. Leonardo Dagnino Chiwiacowsky

Prof. Dr. Plácido Rogério Pinheiro

São Leopoldo

2011

AGRADECIMENTOS

Em especial ao meu orientador Prof. Dr. Arthur Tórgo Gómez, não somente por ter me apresentado às heurísticas, mas também por ter acreditado em mim e me mostrado a beleza por traz da ciência.

Ao CNPQ pelo apoio financeiro.

À minha esposa pela paciência, carinho, compreensão e companheirismo.

Aos meus pais, pelo incentivo, apoio, torcida e força.

A todos meus amigos que me ajudaram nessa caminhada.

RESUMO

Este trabalho aborda o Problema de Roteamento de Veículos Capacitado com Janelas de Tempo, onde devem ser atendidas as restrições de capacidade do veículo e as janelas de tempo de atendimento do cliente. Para resolver tal problema serão utilizadas as metaheurísticas Busca Tabu e Algoritmos Genéticos, além do desenvolvimento de um Algoritmo Híbrido baseado nas duas metaheurísticas. Busca-se contribuir com o desenvolvimento de um Algoritmo Híbrido focado no Problema de Roteamento de Veículos que utilize o poder de intensificação da Busca Tabu e o poder de diversificação do Algoritmo Genético, objetivando a obtenção de soluções de boa qualidade sem comprometer o tempo computacional. Nos experimentos, no que tange a Busca Tabu, analisa-se o processo de busca da através da variação do tamanho da Lista Tabu e do número máximo de iterações sem melhora do valor da função objetivo, como critério de parada, aplicados a uma política de intensificação. Para o Algoritmo Genético, é analisada a influência e o comportamento da busca com base em três operadores de cruzamento aplicados a duas políticas de elitismo. Ainda assim, para o Algoritmo Híbrido, analisa-se o impacto do tamanho da Lista Tabu e das taxas de Mutação e Cruzamento. Por fim, os resultados obtidos são comparados com os melhores métodos heurísticos encontrados na literatura e com métodos exatos, onde o Algoritmo Híbrido mostra-se robusto, obtendo soluções ótimas para diversas instancias de problemas.

Palavras chave: metaheurísticas. busca tabu. algoritmos genéticos. algoritmo híbrido. problema de roteamento de veículos. hibridização.

ABSTRACT

This paper approaches the Capacitated Vehicle Routing Problem with Time Windows, which must obey the restrictions on vehicle capacity and time windows for customer service. To solve this problem will be used two metaheuristics, Tabu Search and Genetic Algorithms, and are developed an hybrid algorithm based on this two metaheuristics. The aim is to contribute with the development of a Hybrid Algorithm focused on Vehicle Routing Problem that uses the Tabu Search intensification power and the Genetic Algorithms diversification power, in order to obtain good quality solutions without compromising the computational time. In the experiments, with respect to Tabu Search, we analyze the search process by varying the size of the Tabu List and the maximum number of iterations without improvement in objective function value, such as stopping criterion, applied to an intensification policy. For the genetic algorithm are analyzed the influence and the search behavior on the basis of three crossover operators, applied to two elitism policies. Still, for the hybrid algorithm, we analyze the impact of the Tabu List size and rates of mutation and crossover. Finally, the results are compared with the best heuristics in the literature and with exact methods, where the Hybrid Algorithm shows robust, getting several optimal solutions.

Keywords: metaheuristics, tabu search, genetic algorithm, hybrid algorithm. vehicle routing problem. hybridization.

LISTA DE ABREVIATURAS

AG	Algoritmos Genéticos
BL	Busca Local
BT	Busca Tabu
CA	Critério de Aspiração
DJ	Política de Elitismo estudada por De Jong
FO	Função Objetivo
Ft	<i>Fitness</i>
GE	Política de Elitismo estudada por Geiger
k-DCT	<i>k-Degree Center Tree</i>
OBX	<i>Order Based Crossover</i>
PCV	Problema do Caixeiro Viajante
PMB	Problema da Mochila Binária
PMX	<i>Partially Mapped Crossover</i>
PRV	Problema de Roteamento de Veículos
PRVC	Problema de Roteamento de Veículos Capacitado
PRVCA	Problema de Roteamento de Veículos Capacitado Assimétrico
PRVCCE	Problema de Roteamento de Veículos Capacitado com Coletas e Entregas
PRVCFR	Problema de Roteamento de Veículos Capacitado com Fretes de Regresso
PRVCFRA	Problema de Roteamento de Veículos Capacitado com Fretes de Regresso Assimétricos
PRVCFRJT	Problema de Roteamento de Veículos Capacitado com Fretes de Regresso e Janelas de Tempo
PRVCJT	Problema de Roteamento de Veículos Capacitado com Janelas de Tempo
PRVCRD	Problema de Roteamento de Veículos Capacitado com Restrições de Distancia
PRVCS	Problema de Roteamento de Veículos Capacitado Simétrico
UOBX	<i>Uniform order based crossover</i>

LISTA DE FIGURAS

Figura 2.1 – Representação gráfica de uma possível solução para o PRV	29
Figura 2.2. Variações básicas do PRV	36
Figura 3.1. Representação gráfica do método “b-cyclic, k-transfer”	54
Figura 3.2. Dinâmica do processo de Busca Tabu	63
Figura 3.3. Dinâmica do processo do Algoritmo Genético	69
Figura 3.4. Representação do método de seleção por roleta	72
Figura 3.5. Representação dos cromossomos pais	73
Figura 3.6. Indivíduos gerados	73
Figura 3.7. Cruzamento com um ponto de corte	74
Figura 3.8. Cruzamento de cromossomos com representação não binária	74
Figura 3.9. Cruzamento utilizando PMX	75
Figura 3.10. Representação de cromossomos para o operador de cruzamento OBX	75
Figura 3.11. Seleção das posições para o operador de cruzamento OBX	75
Figura 3.12. Seleção de novos genes para o operador de cruzamento OBX	75
Figura 3.13. Formação do primeiro indivíduo resultante por OBX	76
Figura 3.14. Representação de um cromossomo gerado por OBX	76
Figura 3.15. Seleção das posições para o operador de cruzamento OBX	76
Figura 3.16. Seleção de novos genes para o operador de cruzamento OBX	76
Figura 3.17. Formação do primeiro indivíduo resultante por OBX	76

Figura 3.18. Representação de um cromossomo gerado por OBX.....	76
Figura 3.19. Representação de cromossomos e máscaras para o operador de cruzamento UOBX	77
Figura 3.20. Formação do primeiro indivíduo resultante por UOBX.....	77
Figura 3.21. Representação de um cromossomo gerado por UOBX	77
Figura 3.22. Formação de um indivíduo resultante por UOBX.....	77
Figura 3.23. Representação de um cromossomo gerado por OBX.....	77
Figura 3.24. Representação de mutação em genes binários	78
Figura 3.25. Representação de mutação utilizando <i>swap mutation</i>	78
Figura 4.1. Arquitetura do modelo	85
Figura 4.2. Representação gráfica de uma matriz de distâncias.....	86
Figura 4.3. Representação da obtenção da solução inicial da Busca Tabu	88
Figura 4.4. Fluxo do processo de Busca Tabu	90
Figura 4.5. Representação da estrutura de vizinhança V1.....	92
Figura 4.6. Representação da estrutura de vizinhança V2.....	93
Figura 4.7. Representação da estrutura de vizinhança V3.....	93
Figura 4.8. Fluxo do processo do Algoritmo Genético com elitismo DJ	98
Figura 4.9. Fluxo do processo do Algoritmo Genético com elitismo GE.....	99
Figura 4.10. Fluxograma do processo do Algoritmo Híbrido	101
Figura 6.1 – Etapas do processo de experimentação	130
Figura 6.2 – BT aplicada ao problema RC207 sem política de intensificação.	134
Figura 6.3 – BT aplicada ao problema RC207 com política de intensificação.	136
Figura 6.4 – BT aplicada ao problema R205 sem política de intensificação.	138

Figura 6.5 – BT aplicada ao problema R205 com política de intensificação.	140
Figura 6.6 – BT aplicada ao problema R206 sem política de intensificação.	142
Figura 6.7 – BT aplicada ao problema R206 com política de intensificação.	144
Figura 6.8 – Comparativo de melhora da função objetivo considerando a política de intensificação	146
Figura 6.9 – Evolução da Busca Tabu com e sem intensificação para o problema RC207	147
Figura 6.10 – Evolução da Busca Tabu com e sem intensificação para o Problema RC205.....	147
Figura 6.11 – Evolução da Busca Tabu com e sem intensificação para o problema R206.....	148
Figura 6.12 – Evolução do Algoritmo Genético com operador OBX e com a política DJ para o problema RC207	152
Figura 6.13 – Evolução do Algoritmo Genético com operador PMX e com a política DJ para o problema RC207	154
Figura 6.14 – Evolução do Algoritmo Genético com operador UOBX e com a política DJ para o problema RC207	156
Figura 6.15 – Evolução do Algoritmo Genético com operador OBX e com a política GE para o problema RC207	158
Figura 6.16 – Evolução do Algoritmo Genético com operador PMX e com a política GE para o problema RC207.....	160
Figura 6.17 – Evolução do Algoritmo Genético com operador UOBX e com a política GE para o problema RC207.....	161
Figura 6.18 – Comparativo de melhora do para o problema RC207	162
Figura 6.19 – Evolução do Algoritmo Genético com operador OBX e com a política DJ para o problema R206	164

Figura 6.20 – Evolução do Algoritmo Genético com operador PMX e com a política DJ para o problema R205	166
Figura 6.21 – Evolução do Algoritmo Genético com operador UOBX e com a política DJ para o problema R205	167
Figura 6.22 – Evolução do Algoritmo Genético com operador OBX e com a política GE para o problema R205	169
Figura 6.23 – Evolução do Algoritmo Genético com operador PMX e com a política GE para o problema R205	171
Figura 6.24 – Evolução do Algoritmo Genético com operador UOBX e com a política GE para o problema R205	173
Figura 6.25 – Comparativo de melhora do para o problema R205.....	174
Figura 6.26 – Evolução do Algoritmo Genético com operador OBX e com a política DJ para o problema R206	176
Figura 6.27 – Evolução do Algoritmo Genético com operador PMX e com a política DJ para o problema R206	177
Figura 6.28 – Evolução do Algoritmo Genético com operador UOBX e com a política DJ para o problema R206	179
Figura 6.29 – Evolução do Algoritmo Genético com operador OBX e com a política GE para o problema R206	181
Figura 6.30 – Evolução do Algoritmo Genético com operador PMX e com a política GE para o problema R206	182
Figura 6.31 – Evolução do Algoritmo Genético com operador UOBX e com a política GE para o problema R206	184
Figura 6.32 – Comparativo de melhora do para o problema R206.....	191

LISTA DE TABELAS

Tabela 2.1 - Relação das características dos problemas de roteamento e escalonamento	35
Tabela 3.1 – Trabalhos relacionados com metaheurísticas	56
Tabela 5.1 – Média dos valores obtidos para a série C100 com 25 clientes	106
Tabela 5.2 – Média dos valores obtidos para a série RC100 com 25 clientes	107
Tabela 5.3 – Média dos valores obtidos para a série R100 com 25 clientes	108
Tabela 5.4 – Média dos valores obtidos para a série C200 com 25 clientes	109
Tabela 5.5 – Média dos valores obtidos para a série RC200 com 25 clientes	110
Tabela 5.6 – Média dos valores obtidos para a série R200 com 25 clientes	111
Tabela 5.7 – Média dos valores obtidos para a série C100 com 50 clientes	112
Tabela 5.8 – Média dos valores obtidos para a série RC100 com 50 clientes	113
Tabela 5.9 – Média dos valores obtidos para a série R100 com 50 clientes	114
Tabela 5.10 – Média dos valores obtidos para a série C200 com 50 clientes	115

Tabela 5.11 – Média dos valores obtidos para a série RC200 com 50 clientes	115
Tabela 5.12 – Média dos valores obtidos para a série R200 com 50 clientes	116
Tabela 5.13 – Média dos valores obtidos para a série C100 com 100 clientes	117
Tabela 5.14 – Média dos valores obtidos para a série RC100 com 100 clientes	118
Tabela 5.15 – Média dos valores obtidos para a série R100 com 100 clientes	119
Tabela 5.16 – Média dos valores obtidos para a série C200 com 100 clientes	120
Tabela 5.17 – Média dos valores obtidos para a série RC200 com 100 clientes	120
Tabela 5.18 – Média dos valores obtidos para a série R200 com 100 clientes	121
Tabela 5.19 – Melhores resultados para os problemas da série RC200 com 25 clientes	122
Tabela 5.20 – Melhores resultados para os problemas da série RC200 com 50 clientes	124
Tabela 5.21 – Melhores resultados para os problemas da série RC200 com 100 clientes	126
Tabela 6.1 – Relação de valores para o nbmax	131
Tabela 6.2 – Relação de valores para o tamanho da Lista Tabu	131
Tabela 6.3 - Modelo de tabela de resultados do módulo da Busca Tabu.....	132
Tabela 6.4 – BT aplicada ao problema RC207 sem política de intensificação	133

Tabela 6.5 – BT aplicada ao problema RC207 com política de intensificação	135
Tabela 6.6 – BT aplicada ao problema R205 sem política de intensificação	137
Tabela 6.7 – BT aplicada ao problema R205 com política de intensificação	139
Tabela 6.8 – BT aplicada ao problema R206 sem política de intensificação	141
Tabela 6.9 – BT aplicada ao problema R206 com política de intensificação	143
Tabela 6.10 – Análise de busca sem política de intensificação.....	144
Tabela 6.11 – Análise de busca com política de intensificação.....	145
Tabela 6.12 – Relação de valores para a taxa de cruzamento	149
Tabela 6.13 – Relação de valores para taxa de mutação	149
Tabela 6.14 – Modelo de tabela de resultados.....	150
Tabela 6.15 – AG aplicada ao problema RC207 com operador OBX e com a política DJ.....	151
Tabela 6.16 – AG aplicada ao problema RC207 com operador PMX e com a política DJ.....	153
Tabela 6.17 – AG aplicada ao problema RC207 com operador UOBX e com a política DJ.....	155
Tabela 6.18 – AG aplicada ao problema RC207 com operador OBX e com a política GE	157
Tabela 6.19 – AG aplicada ao problema RC207 com operador PMX e com a política GE	159

Tabela 6.20 – AG aplicada ao problema RC207 com operador UOBX e com a política GE	161
Tabela 6.21 – AG aplicada ao problema R205 com operador OBX e com a política DJ.....	163
Tabela 6.22 – AG aplicada ao problema R205 com operador PMX e com a política DJ.....	165
Tabela 6.23 – AG aplicada ao problema R205 com operador UOBX e com a política DJ.....	166
Tabela 6.24 – AG aplicada ao problema R205 com operador OBX e com a política GE	168
Tabela 6.25 – AG aplicada ao problema R205 com operador PMX e com a política GE	170
Tabela 6.26 – AG aplicada ao problema R205 com operador UOBX e com a política GE	172
Tabela 6.27 – AG aplicada ao problema R206 com operador OBX e com a política DJ.....	175
Tabela 6.28 – AG aplicada ao problema R206 com operador PMX e com a política DJ.....	176
Tabela 6.29 – AG aplicada ao problema R206 com operador UOBX e com a política DJ.....	178
Tabela 6.30 – AG aplicada ao problema R206 com operador OBX e com a política GE	180
Tabela 6.31 – AG aplicada ao problema R206 com operador PMX e com a política GE	181
Tabela 6.32 – AG aplicada ao problema R206 com operador UOBX e com a política GE	183
Tabela 6.33 – Análise de busca com política DJ e operador OBX.....	185

Tabela 6.34 – Análise de busca com política DJ e operador PMX.....	186
Tabela 6.35 – Análise de busca com política DJ e operador UOBX	187
Tabela 6.36 – Análise de busca com política GE e operador OBX	188
Tabela 6.37 – Análise de busca com política GE e operador PMX	189
Tabela 6.38 – Análise de busca com política GE e operador UOBX.....	190
Tabela 6.39 – Relação de valores para a taxa de cruzamento	192
Tabela 6.40 – Relação de valores para a taxa de mutação	192
Tabela 6.41 – Relação de valores para o tamanho da Lista Tabu	192
Tabela 6.42 – Modelo de tabela de resultados.....	192
Tabela 6.43 – Algoritmo Híbrido aplicada ao problema RC207.....	193
Tabela 6.44 – Algoritmo Híbrido aplicada ao problema R205	194
Tabela 6.45 – Algoritmo Híbrido aplicada ao problema R206	195

SUMÁRIO

1 Introdução.....	21
2 O Problema de Roteamento de Veículos.....	25
2.1 Visão Geral do Problema de Roteamento de Veículos.....	25
2.2 Conceitos Básicos do Problema de Roteamento de Veículos.....	28
2.3 Formulação do PRVC.....	32
2.4 Variações do Problema de Roteamento de Veículos.....	35
2.4.1 PRVC com Restrições de Distância.....	38
2.4.2 PRVC com Janelas de Tempo.....	38
2.4.3 PRVC com Fretes de Regresso.....	39
2.4.4 PRVC com Coletas e Entregas.....	41
3 Métodos para Resolução do PRV.....	44
3.1 Métodos Exatos.....	47
3.2 Heurísticas.....	51
3.2.1 Algoritmo de Clarke e Wright.....	52
3.2.2 Algoritmo de Mole e Jamenson.....	53
3.2.3 Algoritmo de Gillet e Miller.....	53
3.2.4 Heurística de Melhoria <i>Intra-Route</i>	54
3.2.5 Heurística de Melhoria <i>Inter-Route</i>	54
3.2.6 Heurística do Vizinho mais Próximo.....	55
3.2.7 Procedimentos de Inserção.....	55

3.3 Metaheurísticas	56
3.3.1 Trabalhos Relacionados	58
3.3.2 Busca Tabu.....	62
3.3.2.1 Espaço de Soluções.....	65
3.3.2.2 Estrutura de Vizinhança.....	65
3.3.2.3 Lista Tabu	66
3.3.2.4 Critério de Aspiração	67
3.3.2.5 Critério de Parada	67
3.3.2.6 Política de Intensificação	68
3.3.2.7 Política de Diversificação	68
3.3.3 Algoritmos Genéticos	68
3.3.3.1 Codificação	70
3.3.3.2 População Inicial.....	71
3.3.3.3 Critério de Parada	71
3.3.3.4 Condição de Cruzamento e Mutação	72
3.3.3.5 Métodos de Seleção	72
3.3.3.6 Cruzamento	74
3.3.3.7 Mutação.....	79
3.3.3.8 Nova População	80
4 O Modelo.....	82
4.1 Definição do Modelo	82

4.2	Formulação Matemática.....	83
4.3	Arquitetura do modelo	85
4.3.1	Geração das Redes	87
4.3.2	Geração das Soluções Iniciais.....	87
4.3.2.1	Solução Inicial para o Módulo da Busca Tabu	88
4.3.2.2	População Inicial para os Módulos do Algoritmo Genético e Algoritmo Híbrido	90
4.3.3	Módulo da Busca Tabu	90
4.3.3.1	Representação da Solução.....	92
4.3.3.2	Estrutura de Vizinhanças	92
4.3.3.3	Lista Tabu	95
4.3.3.4	CrITÉrio de AspiraÇão	95
4.3.3.5	IntensificaÇão	95
4.3.3.6	CrITÉrio de Parada	96
4.3.4	Módulo do Algoritmo Genético.....	96
4.3.4.1	CodificaÇão	96
4.3.4.2	Operadores de RecombinaÇão Genética	97
4.3.4.3	CrITÉrio de Parada	97
4.3.4.4	Elitismo	98
4.3.5	Módulo do Algoritmo Híbrido.....	101
4.3.5.1	Parâmetros e Definições do Método.....	103
5	ValidaçãO	106

6 Resultados Obtidos e Análise dos Dados	130
6.1 Busca Tabu.....	131
6.2 Algoritmo Genético	149
6.3 Algoritmo Híbrido	192
7 Conclusão	198
8. Referências Bibliográficas.....	202

1 Introdução

Atualmente, grandes esforços vêm sendo realizados no sentido de desenvolver algoritmos eficientes que possam ser utilizados na resolução de problemas de Otimização Combinatória. Esses esforços se devem a crescente utilização de pacotes de otimização baseados em Pesquisa Operacional e técnicas de Programação Matemática (Toth e Vigo, 2002).

A área da Otimização Combinatória abrange diversos problemas que abordam aplicações práticas e que necessitam de algoritmos eficientes, para que possam ser tratados de maneira aceitável computacionalmente. Dentre esses problemas, podem ser citados o Problema do Caixeiro Viajante, o Problema da Mochila e o Problema da Coloração de Grafos. Osman e Laporte (1996) ressaltam que os problemas de Otimização Combinatória, geralmente, são simples de serem enunciados, porém, de difícil resolução. Isso se deve à grande quantidade de possíveis soluções que devem ser avaliadas para garantir que a melhor solução seja obtida. Visto isso, os métodos exatos podem não ser a melhor opção, pois exigem um enorme esforço computacional. Por outro lado, surgem as heurísticas. Heurísticas são métodos desenvolvidos de modo a não percorrer todo o espaço de soluções, diminuindo o tempo necessário para obter uma solução. Essa característica faz com que não seja possível garantir que a solução ótima seja obtida, contudo, muitas vezes é preferível trabalhar com uma solução boa, proporcionada pelas heurísticas, do que aceitar o elevado tempo computacional demandado pelos métodos exatos para obter a solução ótima.

Nas áreas da logística, transporte e distribuição, frequentemente surgem situações que podem ser tratadas como sendo problemas de Otimização Combinatória (Ho e Haugland, 2004; Xu e Kelly, 1996). Um exemplo disso baseia-se no fato de depósitos terem de suprir pontos de demanda, através do transporte de insumos ou serviços. Neste caso, faz-se necessário a criação de um roteiro que distribua e ordene os clientes a serem atendidos, de modo que seja realizado da

maneira mais econômica possível. Essa situação caracteriza o Problema de Roteamento de Veículos (PRV), o qual será abordado neste trabalho.

O ano de 2009 marcou o aniversário de 50 anos da publicação do primeiro artigo sobre o Problema de Roteamento de Veículos sob o título de “*The truck dispatching problem*” (Dantzig e Ramser, 1959; Laporte, 2009). Muitos progressos foram obtidos quanto à proposição de métodos eficientes para resolução do Problema de Roteamento de Veículos. Apesar disso, devido às dificuldades intrínsecas deste problema, classificado como NP-Difícil (Toth e Vigo, 2002; Iori et al, 2007; Laporte, 2009), a maior parte dos métodos estudados para sua resolução baseia-se em heurísticas.

Este trabalho tem o objetivo de desenvolver um modelo para abordar o Problema de Roteamento de Veículos Capacitado com Janelas de Tempo, utilizando Busca Tabu, Algoritmos Genéticos e um Algoritmo Híbrido. Esse Algoritmo Híbrido combina o poder de intensificação, proporcionado pela Busca Tabu, e o poder de diversificação, proporcionado pelo Algoritmo Genético, permitindo que ele obtenha resultados de boa qualidade para problemas de pequena, média e larga escala.

A principal contribuição deste trabalho se refere ao desenvolvimento de um Algoritmo Híbrido focado na resolução do Problema de Roteamento de Veículos Capacitado com Janelas de Tempo. Identificou-se que o Algoritmo Híbrido obteve diversas soluções ótimas, além de se mostrar mais eficiente que outros métodos heurísticos já desenvolvidos. Além disso, realizaram-se estudos que avaliaram o processo de exploração do espaço de soluções realizado pela Busca Tabu e pelo Algoritmo Genético. A Busca Tabu é avaliada através da variação do tamanho da Lista Tabu e do número máximo de iterações sem melhora no valor da função objetivo, aplicados a uma política de intensificação. Desse modo, desenvolveu-se um algoritmo robusto e que obteve soluções de boa qualidade. O Algoritmo Genético é analisado através da variação das taxas de cruzamento e mutação, aplicadas a três operadores de cruzamento, *Order Based Crossover*, *Partially Mapped Crossover* e *Uniform Order Based Crossover*. Ainda assim, analisam-se, separadamente, duas políticas de elitismo de modo a desenvolver um algoritmo eficiente, do ponto de vista da diversificação, e que obtenha soluções de boa qualidade.

O trabalho está organizado no que segue: o capítulo 2 contextualiza o Problema de Roteamento de Veículos, trazendo conceitos básicos, principais variações e sua formulação matemática. O capítulo 3 apresenta as formas de resolução do PRV, assim como expõem alguns trabalhos realizados na área. O capítulo 4 apresenta o modelo desenvolvido, trazendo a formulação matemática, a arquitetura do modelo desenvolvido, as definições dos métodos que serão utilizados e o planejamento dos experimentos. O capítulo 5 apresenta a validação do modelo. O capítulo 6 apresenta os experimentos e a análise dos resultados obtidos e o capítulo 7 apresenta as considerações finais.

2 O Problema de Roteamento de Veículos

Este capítulo apresenta uma visão geral sobre o Problema de Roteamento de Veículos, alguns conceitos básicos, nomenclaturas, definições e suas principais variações. Além disso, é apresentada sua formulação matemática para um problema genérico considerando as restrições de capacidade, distância percorrida e prioridades de entrega.

2.1 Visão Geral do Problema de Roteamento de Veículos

A logística pode ser definida como a provisão de bens e serviços de pontos de oferta a pontos de demanda (Bodin, Golden e Assad, 1983). Para Ballou (2006) o custo com transporte, normalmente, representa o elemento mais importante em termos de custos logísticos, sendo responsável por um a dois terços dos custos logísticos totais, constituindo uma das maiores preocupações do setor. Desse modo, busca-se aumentar a eficiência por meio da utilização máxima dos equipamentos e pessoal.

Nesse contexto, Ballou (2006) aponta, como um problema frequente, a redução dos custos de transporte e a melhora dos serviços, descobrindo, por exemplo, os melhores roteiros para os veículos ao longo de uma rede rodoviária, ferroviária, hidroviária ou rotas de navegação aérea. Portanto, devido à representatividade do custo da atividade de transporte, muitas empresas consideram o transporte como a atividade de maior importância dentro da logística (Ballou, 2006; Barbarosoglu e Ozgur, 1999).

Visto que o PRV possui grande aplicação, a maior parte dos problemas no domínio da distribuição de bens pode ser vista como um Problema de Roteamento de Veículos (Mester e Bräysy, 2005). Abaixo, uma relação de aplicações práticas do problema encontrada na literatura:

- Transporte Escolar (Newton e Thomas, 1974);
- Distribuição de Jornais (Golden et al, 1977);
- Sistemas de transportes coletivos urbanos (Ceder e Stern, 1981);
- Distribuição de Alimentos (Bartholdi *et al*, 1983);
- Distribuição de Manufaturados (Perl e Daskin, 1985);
- Entrega de correspondência bancária (Malmborg e Simons, 1989);
- Roteamento de helicópteros (Timlin e Pulleyblank, 1990);
- Recolhimento de lixo (Kulcar, 1996);
- Distribuição de produtos eletrônicos (Barbarasolgu e Ozgur, 1999);
- Problema de Roteamento de Locomotivas (Vaidyanathan et al, 2008);
- Roteamento dinâmico de Linhas Aéreas (Jiang e Barnhart, 2009).

Toth e Vigo (2002) definem os principais componentes do PRV com sendo: rede viária, clientes, veículos e motoristas. Esses componentes são detalhados nos próximos parágrafos.

A rede viária, utilizada para o transporte de mercadorias, em geral, é descrita através de um grafo, cujas arestas representam as seções de caminho a serem percorridas na rede viária e os vértices representam os clientes e o depósito. As arestas podem ser direcionadas, caso elas possam ser percorridas em um único sentido (por exemplo, devido à presença de ruas de mão única, típica de redes urbanas ou autoestrada), ou podem ser não direcionadas, caso possam ser percorridas em ambas as direções. A cada arco está associado um custo e um tempo de viagem, que é possivelmente dependente do tipo de veículo ou do período do dia durante o qual a aresta é percorrida.

Os atributos típicos dos clientes são:

- Vértice do grafo que representa a localização do cliente;
- Quantidade demandada de produtos;

- Janelas de tempo durante a qual o cliente pode ser servido (isso pode ocorrer, pois existem períodos específicos do dia que o cliente está aberto ou o local pode ser alcançado, devido às limitações de tráfego);
- Os tempos necessários para entregar ou coletar a mercadoria no local do cliente (tempos de descarga ou carga, respectivamente), possivelmente dependente do tipo de veículo;
- Subconjunto dos veículos disponíveis que podem ser usados para servir o cliente (por exemplo, devido às limitações de acesso possível ou necessidades de carga e descarga).

O transporte de mercadorias é realizado utilizando uma frota de veículos, cuja composição e tamanho podem ser fixos ou podem ser definidos de acordo com as exigências dos clientes.

Os atributos típicos dos veículos são:

- Depósito de origem;
- Possibilidade de finalização de um serviço em um depósito diferente do depósito de origem;
- Capacidade: expressa como o peso máximo, ou volume, ou número de *palets* que o veículo pode transportar;
- Subdivisão do veículo em compartimentos, cada um, caracterizado por sua capacidade e por tipos de bens que possam ser carregados;
- Dispositivos disponíveis para as operações de carga e descarga;
- Subconjunto de arestas que pode ser percorrido pelo veículo;
- Custos associados com a utilização do veículo (por exemplo, por unidade de distância, por unidade de tempo, por rota).

Os motoristas dos veículos devem satisfazer várias restrições estabelecidas por contratos e regulamentos da empresa. Por exemplo, períodos de trabalho durante o dia, o número e a duração das pausas durante o serviço, a duração

máxima dos períodos de condução, as horas extras. Por fim, as restrições impostas aos condutores são associadas aos veículos correspondentes.

As rotas iniciam e são finalizadas em um ou mais depósitos. Neste trabalho é considerado somente um único depósito. Cada depósito é caracterizado pelo número e tipos de veículos e pelo montante global de bens que podem lidar. Em algumas aplicações do mundo real, os clientes são, *a priori*, divididos entre os depósitos e os veículos têm de regressar ao seu depósito de origem no final de cada percurso. Nestes casos, o PRV global pode ser decomposto em vários problemas independentes, cada um associado a um depósito diferente (Toth e Vigo, 2002). Ainda assim, as rotas devem respeitar diversas restrições operacionais, que dependem da natureza das mercadorias transportadas, da qualidade do nível de serviço, e das características dos clientes e dos veículos. Algumas restrições operacionais típicas são as seguintes (Toth e Vigo, 2002):

- Ao longo de cada percurso, a carga total carregada não pode exceder a capacidade do veículo;
- Os clientes podem ser servidos apenas em sua janela de tempo e os períodos de trabalho dos condutores.

As restrições de precedência podem ser aplicadas no momento em que os clientes são visitados. Nesse contexto, algumas restrições de precedência exigem que um determinado cliente seja atendido na mesma rota que um dado subconjunto de outros clientes e que o cliente deve ser visitado antes (ou depois) dos clientes pertencentes ao subconjunto associado. Esse é o caso dos problemas de coleta e entrega, onde as rotas podem realizar a coleta e entrega das mercadorias e as mercadorias recolhidas em um cliente devem ser entregues para outro cliente pelo mesmo veículo.

2.2 Conceitos Básicos do Problema de Roteamento de Veículos

Problema de Roteamento de Veículos é o nome dado a uma classe de problemas que envolve a visita de clientes por veículos e no qual um custo mínimo

para esse atendimento deve ser buscado (Bodin, Golden e Assad, 1983; Tarantilis, Ioannou, Prastacos, 2005). Toth e Vigo (2002) apontam alguns conceitos básicos sobre a versão clássica do PRV que são considerados para formulação do modelo que é abordado neste estudo.

A versão clássica do PRV é conhecida como o Problema de Roteamento de Veículos Capacitado (PRVC) (Laporte, 1992; Thangiah e Petrovic, 1997; Ralph *et al*, 2001; Toth e Vigo, 2002; Tarantilis *et al*, 2005), no qual todos os clientes que correspondem às entregas e as demandas são conhecidos de antemão. Os veículos são idênticos, baseados em um único depósito central e somente a restrição de capacidade é imposta. O objetivo é minimizar o custo total para servir todos os clientes.

O PRVC pode ser descrito como um problema de grafos (Christofides *et al*, 1979; Toth e Vigo, 2002; Simas, 2007). Seja $G = (V, A)$ um grafo completo, onde $V = \{0, \dots, n\}$ é o conjunto de vértices e A é o conjunto de arestas. Os vértices v_i , $i = 1, \dots, n$, correspondem aos clientes, enquanto o vértice v_0 corresponde ao depósito. Em alguns trabalhos, o depósito está associado, também, ao vértice v_{n+1} . A Figura 2.1, mostra um exemplo de solução para o PRVC. Nesse exemplo, o depósito é representado ao centro, pelo vértice v_0 , enquanto que as cidades são representadas pelos vértices v_i , onde $i = 1, \dots, 11$. A solução obtida é composta por três rotas. A primeira rota interliga o depósito às cidades 1, 2 e 3, a segunda rota interliga o depósito às cidades 4, 5, 6 e 7 e a terceira rota interliga o depósito às cidades 8, 9, 10 e 11.

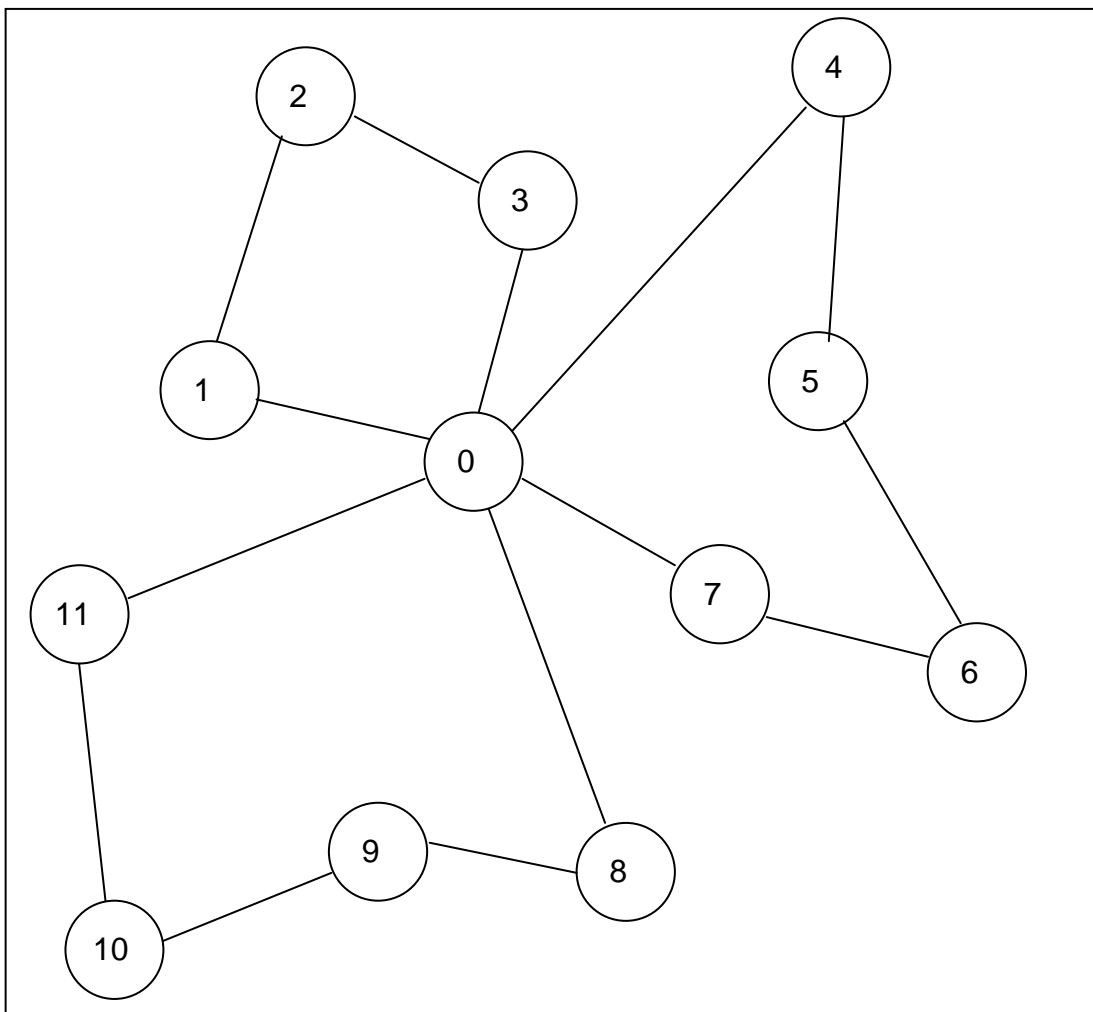


Figura 2.1 – Representação gráfica de uma possível solução para o PRV

Um custo não negativo, C_{ij} , está associado a cada aresta $(i, j) \in A$ e representa o custo de viagem gasto para ir do vértice i ao vértice j . Geralmente, o uso dos arcos do laço, (i, i) , não é permitido. Esta regra é imposta uma vez que $c_{ii} = +\infty$ para todo $i \in V$. Se G é um grafo orientado, a matriz de custos C é assimétrica, então o problema correspondente é chamado de PRVC Assimétrico (PRVCA). Vale ressaltar que o PRVCA aborda muitas aplicações práticas, como por exemplo, em problemas de distribuição de bens e sequenciamentos (Vigo, 1994). Caso contrário, tem-se $c_{ij} = c_{ji}$ para todo $(i, j) \in A$, então o problema é chamado PRVC Simétrico (PRVCS).

O grafo G deve ser conexo e, geralmente, assume-se que seja completo. Desse modo, os elementos do conjunto de vértices, $v_i \in V$, são conectados por um conjunto de arestas orientadas $a = (i, j) \in A$. O conjunto de arestas que sai do vértice

v_i (correspondendo a uma estrutura *forward star*) é denotado por $\Delta^+(i)$, e o conjunto de arestas que chega ao vértice v_i (*backward star*) é denotado por $\Delta^-(i)$.

Ainda assim, Toth e Vigo (2002) explicam que, em vários casos práticos, o custo da matriz satisfaz a desigualdade triangular conforme equação (2.1).

$$c_{ik} + c_{kj} \geq c_{ij} \text{ para todo } i, j, k \in V \quad (2.1)$$

Ou seja, pode não ser conveniente desviar a ligação direta entre dois vértices i e j . A presença da desigualdade do triângulo é, algumas vezes, exigida pelos algoritmos para PRVC e isso pode ser obtido de maneira simples, adicionando uma quantidade suficientemente grande, M , positiva para o custo de cada aresta. Com isso, quando o custo de cada arco do grafo é igual ao custo do menor caminho entre o seu terminal, o custo do desvio correspondente satisfaz a desigualdade triangular.

Em alguns casos, os vértices estão associados a pontos do plano, sendo representados por suas coordenadas. Desta forma, o custo c_{ij} para cada aresta $(i, j) \in A$ pode ser definido como a distância euclidiana entre os dois pontos correspondentes aos vértices i e j . Neste caso, a matriz custo é simétrica e satisfaz a desigualdade triangular, equação (2.1), e o problema resultante é chamado de PRVCS Euclidiano. Esse conceito é aplicado no modelo desenvolvido neste estudo.

Cada cliente i ($i = 1, \dots, n$) está associado a uma demanda conhecida não negativa, d_i , a ser atendida, enquanto o depósito tem uma demanda fictícia $d_0 = 0$. Dado um conjunto de vértices $S \subseteq V$, então $d(S) = \sum_{i \in S} d_i$ denota a demanda total do conjunto. Um conjunto de k veículos idênticos, cada um com capacidade Q , está disponível no depósito. Para garantir a viabilidade assume-se que $d_i \leq Q$ para cada $i = 1, \dots, n$. Cada veículo pode executar no máximo uma rota, e assume-se, também, que k não é menor que k_{\min} , onde k_{\min} é o número mínimo de veículos necessários para atender a todos os clientes. O valor do k_{\min} pode ser determinado pela resolução do Problema da Mochila Binária (PMB) associado ao PRVC, que tende para a determinação do número mínimo de caixas, cada uma com capacidade Q , necessário para carregar todos os n itens, cada um com peso não negativo d_i , $i = 1, \dots, n$. Embora o PMB seja NP-difícil (Toth e Vigo, 2002), as ocorrências com centenas de itens pode ser perfeitamente resolvido de forma ótima (Martello e Toth, 1990).

O PRVC consiste em encontrar um conjunto de exatos R circuitos simples (cada um correspondendo a rota de um veículo), com um custo mínimo, definido como a soma dos custos dos arcos que pertencem aos circuitos, e de tal forma que:

- Cada circuito visite o vértice do depósito;
- Cada cliente seja visitado somente por um único circuito;
- O somatório das demandas de cada vértice visitado pelo circuito não pode exceder a capacidade do veículo, Q .

Segundo Toth e Vigo (2002), duas variações básicas são consideradas na literatura quanto à utilização dos veículos. Na primeira variação, o número de k veículos disponíveis é maior que k_{\min} , pode ser possível deixar alguns veículos sem utilização. Neste caso, custos fixos são geralmente associados ao uso dos veículos. Na segunda variação, os veículos disponíveis são diferentes, ou seja, têm diferentes atributos, como por exemplo, capacidades.

O PRVC é NP-difícil (Toth e Vigo, 2002; Iori et al, 2007) e generaliza o Problema do Caixeiro Viajante (PCV), apelando para a determinação de um custo mínimo de circuito simples, visitando todos os vértices de G (circuito hamiltoniano) e que surge quando $Q \geq d(V)$ e $K = 1$. Portanto, todas as propostas para a abordagem do PCV são válidas para o PRVC (Toth e Vigo, 2002).

2.3 Formulação do PRVC

Nesta seção é apresentada uma modelagem genérica, para o PRVC com o objetivo de aplicar os conceitos básicos, vistos seção 2.2. Essa modelagem segue a ideia de um PRV com restrições de capacidade do veículo, tempo total de trabalho e uma prioridade associada às entregas. Com isso, busca-se ter uma visão alternativa para a abordagem utilizada na formulação do modelo desenvolvido. Além disso, são citadas três abordagens básicas encontradas na literatura, conforme apresentado por Christofides et. al. (1979) e Toth e Vigo (2002).

A primeira abordagem é conhecida como formulação de Fluxo de Veículos, na qual se utiliza variáveis numéricas inteiras, associadas a cada aresta do grafo, que

contam o número de vezes que a aresta é percorrida por um veículo. Estes modelos são mais utilizados para as versões básicas do PRV (Christofides et. al., 1979). Eles são particularmente adequados para os casos em que o custo da solução pode ser expresso como a soma dos custos associados às arestas, e quando os mais relevantes dizem respeito a restrições de transição direta entre os clientes dentro da rota. Ainda assim, os modelos de fluxo de veículos não podem ser usados para lidar com muitas questões práticas, por exemplo, quando o custo de uma solução depende da sequência de vértices em geral ou sobre o tipo de veículo designado para uma rota. Além disso, a relaxação de modelos de programação linear de fluxo de veículos pode ser muito fraca, quando as restrições operacionais adicionais estão apertadas.

A segunda abordagem é baseada na formulação conhecida como Fluxo de Mercadorias. Neste tipo de modelo, variáveis inteiras adicionais são associadas aos arcos e representam o fluxo de mercadorias ao longo dos caminhos percorridos pelos veículos. Só recentemente os modelos deste tipo foram usados como base para a solução exata do PRVC (Toth e Vigo, 2002).

A terceira abordagem possui um número exponencial de variáveis binárias, cada um associado a um circuito viável diferente. O PRV é então formulado como um Problema de Particionamento de Conjuntos, para a determinação de um conjunto de circuitos com o mínimo custo, que atende a cada cliente uma vez e, se possível, satisfaça às restrições adicionais. A principal vantagem deste tipo de modelo é que o cálculo do custo de seus percursos é extremamente genérico, dependendo, basicamente, da sequência dos arcos e do tipo de veículo. Como resultado, muitas vezes eles podem ser substituídos por um conjunto compacto de desigualdades. Isso produz uma formulação de programação linear cujo relaxamento é, normalmente, muito mais apertado do que nos modelos anteriores. Apesar disso, esses modelos geralmente necessitam lidar com um número muito grande de variáveis, o que pode acarretar problemas de desempenho.

Os modelos citados anteriormente podem ser encontrados em Christofides et. al. (1979) e Toth e Vigo (2002) para um maior aprofundamento. A seguir, é apresentada a formulação matemática para um PRV, apresentada por Christofides et. al. (1979), onde o objetivo é minimizar a distância total percorrida (assumindo que

todos os clientes serão atendidos), sendo que existe uma prioridade δ_i , onde $i = \{1, \dots, n\}$, associada a cada cliente i . O modelo apresentado a seguir é uma abordagem alternativa ao modelo desenvolvido neste trabalho.

Considera-se o PRV para um dado período de tempo T . Seja $V = \{v_i \mid i = 1, \dots, n\}$ o conjunto de n clientes (vértices), v_0 o depósito e $K = \{k_l \mid l = 1, \dots, m\}$ o conjunto de m veículos estacionados no depósito.

Um cliente v_i tem os seguintes requisitos:

- Uma quantidade d_i de algum produto para ser entregue por um veículo qualquer ao cliente i ;
- Uma prioridade δ_i associada a um cliente i ;
- Um custo de viagem c_{ij} para ir do cliente i até o cliente j ;
- Um tempo de viagem t_{ij} para ir do cliente i até o cliente j .

Um veículo possui as seguintes características:

- Uma capacidade total Q_k para carregar os produtos associada a cada veículo k ;
- Um tempo s_{ik} necessário para o veículo k descarregar a quantidade d_i associada ao cliente i ;
- Um tempo máximo de trabalho de T_k associado ao veículo k ;
- Um custo fixo F_k associado ao veículo k .

O objetivo é minimizar o custo total da rota maximizando as prioridades de entrega dos clientes. Desse modo, sendo $x_{ijk} = 1$ se o veículo k visitou o cliente j imediatamente após visitar o cliente i e $x_{ijk} = 0$ caso contrário, a formulação matemática segue abaixo.

$$\min f = \alpha \sum_{i=0}^n \sum_{j=0}^n \left(c_{ij} \sum_{k=1}^m x_{ijk} \right) + \beta \sum_{k=1}^m \left(F_k \sum_{j=1}^n x_{0jk} \right) + \gamma \sum_{i=1}^n \left(\delta_i \sum_{j=0}^n \sum_{k=1}^m x_{ijk} \right) \quad (2.3)$$

Sujeito a

$$(2.4)$$

$$\sum_{i=0}^n \sum_{k=1}^m x_{ijk} \leq 1 \quad j = 1, \dots, n$$

$$\sum_{i=0}^n x_{ipk} - \sum_{j=0}^n x_{pj k} = 0 \quad k = 1, \dots, m; \quad p = 0, \dots, n$$

$$\sum_{i=1}^n \left(d_i \sum_{j=0}^n x_{ij k} \right) \leq Q_k \quad k = 1, \dots, m$$

$$\sum_{i=0}^n \sum_{j=0}^n t_{ij} x_{ij k} + \sum_{i=1}^n \left(s_i \sum_{j=0}^n x_{ij k} \right) \leq T_k \quad k = 1, \dots, m$$

$$\sum_{j=1}^n x_{0jk} \leq 1 \quad k = 1, \dots, m$$

$$x_{ijk} \in \mathbb{Z}, \text{ para todo } i, j, k$$

$$x_{ijk} \in \{0, 1\}, \text{ para todo } i, j, k$$

A função objetivo é expressa pela equação (2.3). Cada variável do modelo está associada a um peso. O custo total da viagem está associado ao peso α , o custo fixo dos veículos está associado ao peso β e as prioridades de entrada estão associadas ao peso γ . Desse modo, podem-se analisar, separadamente, cada variável do modelo ou definir qual possui maior importância na estratégia da empresa. A inequação (2.4) define que um cliente pode ser visitado somente uma vez. A inequação (2.5) define que se um veículo visitar o cliente i , ele só poderá partir desse cliente i para outro cliente j . As inequações (2.6) e (2.7) são as limitações de capacidades e tempo de trabalho do veículo. A inequação (2.8) define que um veículo pode ser usado somente uma única vez. A expressão (2.9) é a condição de eliminação de sub-rotas onde Z é definido pela inequação (2.11).

$$Z = \left\{ (X_{ijk}) : \sum_{i \in B} \sum_{j \in B} X_{ijk} \leq |B| - 1 \text{ para } B \subseteq \frac{V}{\{0\}}, |B| \geq 2 \right\}$$

2.4 Variações do Problema de Roteamento de Veículos

Além do PRVC clássico, apresentado anteriormente na seção 2.3, são apresentadas algumas características e diferentes restrições que podem ser utilizadas para compor as variações do PRV, conforme Tabela 2.1.

Tabela 2.1 - Relação das características dos problemas de roteamento e escalonamento

Características	Possíveis Opções
Tamanho da frota disponível	Um veículo e múltiplos veículos.
Tipo de frota disponível	Homogênea e heterogênea.
Domicílio dos Veículos	Único depósito e múltiplos depósitos.
Natureza das demandas	Estocástica e determinística.
Localização das demandas	Nos nodos e nos arcos.
Tipo de rede	Não-direcionada, direcionada e euclidiana.
Restrições quanto à capacidade dos veículos	Capacidades iguais, diferentes e sem capacidade.
Tempo Máximo de duração da rota	O mesmo tempo para todas as rotas, diferentes para rotas diferentes e tempo não imposto.
Operação	Entrega, recolhimento e ambos.
Características	Possíveis Opções
Custos	Variáveis por rota e fixos.
Objetivos	Minimizar custo total das rotas, minimizar a soma dos custos fixos e variáveis e minimizar número de veículos requeridos.

Fonte: Bodin, Golden e Assad, (1983).

A partir da Tabela 2.1, observa-se várias possibilidades de tipos de PRV, conforme as características e opções consideradas. Dentre as possíveis

combinações que podem ser realizadas tendo como base a Tabela 2.1, Toth e Vigo (2002) apontam algumas variações básicas do PRV, através da Figura 2.2.

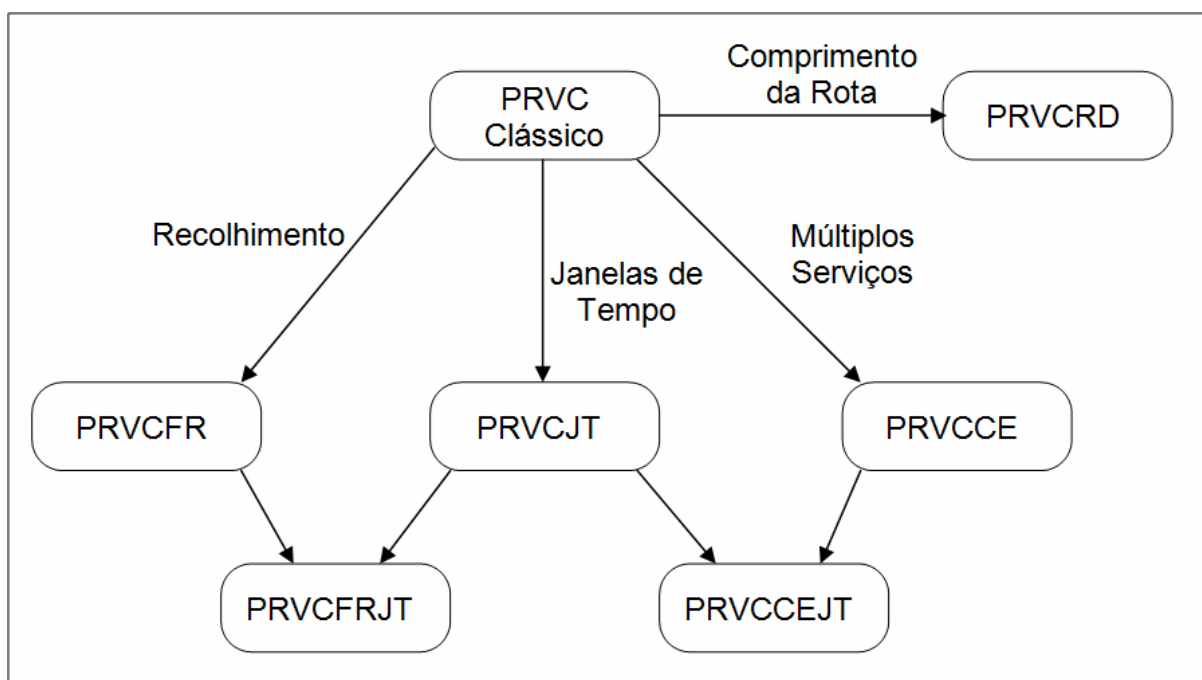


Figura 2.2. Variações básicas do PRV

A Figura 2.2, mostra o problema de roteamento de veículos capacitado clássico como sendo a origem par a criação dos demais problemas. A partir dele, surgem as variações: Problema de Roteamento de Veículos Capacitado com Restrições de Distancia (PRVCRD), onde se considera o comprimento da rota, o Problema de Roteamento de Veículos Capacitado com Fretes de Regresso (PRVCFR), onde se considera o recolhimento de materiais na volta para o depósito, o Problema de Roteamento de Veículos Capacitado com Janelas de Tempo (PRVCJT), onde se considera janelas de tempo em que os clientes podem ser atendidos, o Problemas de Roteamento de Veículos Capacitado com Coletas e Entregas (PRVCCE), onde se consideram múltiplos serviços, coletas e entregas, na mesma rota. Ainda assim, a partir dessas variações, novas acabam surgindo como no caso do Problema de Roteamento de Veículos Capacitado com Fretes de Retorno e Janelas de Tempo (PRVCFRJT) e o Problema de Roteamento de Veículos Capacitado com Coletas e Entregas e Janelas de Tempo (PRVCCEJT).

A seguir, são apresentadas algumas características dos problemas PRVCRD, PRVCFR, PRVCJT e PRVCCE mostrados na Figura 2.2. Os problemas PRVCFRJT

e PRVCCEJT não serão apresentados, pois unem os conceitos apresentados pelos demais. Essa seção tem como objetivo realizar uma breve apresentação sobre as diferenças existentes em cada variação do problema, sendo que o PRVCJT é aplicado neste estudo.

2.4.1 PRVC com Restrições de Distância

O PRVC com Restrição de Distância é uma variação do PRVC (Toth e Vigo, 2002), onde cada rota passa a ter a restrição de um comprimento máximo, além da restrição de capacidade. Em particular, um comprimento não negativo, L_{ij} está associado a cada aresta $(i, j) \in A$, ligando o cliente i ao cliente j . O comprimento total das arestas de cada rota não pode exceder o comprimento máximo de rota, R . Se os veículos são diferentes, então os comprimentos de percurso máximo são R_k , onde $k = 1, \dots, m$ e m é o número total de veículos disponíveis. Além disso, quando os comprimentos dos arcos representam os tempos de viagem, um tempo de serviço s_i , pode ser associado a cada cliente i , denotando o período de tempo que o veículo necessita para descarregar o produto. Alternativamente, os tempos de serviço podem ser adicionados ao tempo de viagem das arestas, ou seja, definindo t_{ij} como sendo o tempo de viagem para cada aresta (i, j) que liga o cliente i ao cliente j .

Geralmente, as matrizes de custo e duração coincidem, ou seja, $c_{ij} = t_{ij}$ para todos $(i, j) \in A$ (ou $c_v = t_v$ para todos os $v \in V$). Portanto, o objetivo do problema é minimizar a distância total das rotas, ou sua duração, quando o tempo de serviço está incluído no tempo de viagem das arestas. Por fim, o PRVRD é considerado NP-difícil (Franceschi et. al., 2006; Lysgaard, 2004).

2.4.2 PRVC com Janelas de Tempo

O PRVC com Janelas de Tempo (PRVCJT), que será abordado neste estudo, é uma extensão do PRVC no qual restrições de capacidade são impostas e cada

cliente i está associado a um intervalo de tempo $[a_i, b_i]$, denominado de janela de tempo (Toth e Vigo, 2002). Desse modo, existe uma janela de tempo associada a cada cliente e um veículo que necessite visitar um determinado cliente não pode chegar depois do fim da janela de tempo (Ropke e Cordeau, 2009).

Nesse caso, o instante em que os veículos deixam o depósito, o tempo de viagem, t_{ij} , para cada aresta $(i, j) \in A$ e um tempo adicional de serviço s_i para cada cliente i , são todos conhecidos *a priori*, sendo assim, determinísticos. Ainda assim, normalmente, as matrizes de custo e tempo de viagem coincidem.

O serviço de cada cliente deve começar dentro de sua janela de tempo sendo que, em caso de chegada antecipada no local do cliente i , o veículo, geralmente, pode esperar até que o serviço possa começar.

Com isso, as janelas de tempo induzem a uma orientação implícita para cada rota mesmo que as matrizes originais sejam simétricas, Portanto, o PRVCJT é normalmente modelado como um problema assimétrico. O PRVCJT consiste em encontrar um conjunto de exatamente R circuitos simples, de custo mínimo, de tal forma que:

- Cada circuito inicie e finalize no vértice de depósito;
- Cada vértice de cliente (vértice) é visitado por exatamente um circuito;
- A soma das demandas dos vértices visitados por um circuito não exceda a capacidade Q do veículo k ;
- Em cada cliente i o veículo pára por s_i instantes de tempo dentro da janela de tempo $[a_i, b_i]$ para descarregar o produto.

O PRVCJT é NP-difícil (Toth e Vigo, 2002; Ropke e Cordeau, 2009; Dell'Amico et al, 2007), uma vez que generaliza o PRVC, que surge quando $a_i = 0$, $b_i = +\infty$, para cada $i \in V$.

2.4.3 PRVC com Fretes de Regresso

O PRVCFR é outra variação do PRVC. Liu e Chung (2009) abordam que o aumento dos custos dos materiais fez com que as empresas considerassem a reutilização de materiais recicláveis, gerando uma preocupação com o agendamento das coletas desses resíduos. Com isso, o PRVC com Fretes de Regresso (PRVFR) tem sido muito discutido nos últimos anos (Liu e Chung, 2009; Ropke e Pisinger, 2006; Bianchessi e Righini, 2007). Na literatura são apresentadas diversas aplicações reais para o PRVCFR, dentre as quais podem ser citadas empresas que vendem extintores de incêndio (Liu e Chung, 2009). Nesse caso, a empresa passa entregando os extintores novos e recolhendo os usados. Além desse exemplo, Brandão (2006) apresenta o caso de uma empresa que distribui café para seus clientes e recolhe os produtos vencidos. Brandão (2006) também divide o PRVCFR em três tipos:

- As coletas podem ser realizadas depois das entregas;
- As coletas podem ser realizadas antes das entregas e os clientes podem ser tanto pontos de coletas como de entregas;
- Em uma rota, primeiro são servidos os pontos de entrega e depois os pontos de coleta. Nenhuma rota é permitida contendo somente coletas, mas uma rota pode conter somente entregas.

O PRVCFR é a extensão do PRVC (Toth e Vigo, 2002) em que o conjunto de clientes N é particionado em dois subconjuntos. O primeiro subconjunto, L , contém n clientes de entrega, cada um exigindo uma quantidade determinada de produto a ser entregue. O segundo subconjunto, B , contém h clientes de coleta, onde uma determinada quantidade de produto deve ser retirada. Os clientes são numerados de modo que $L = \{1, \dots, n\}$ e $B = \{n + 1, \dots, n+h\}$.

No PRVCFR, podem existir restrições de precedência entre os pontos de entrega e coleta. Uma rota pode, por exemplo, servir dois tipos de clientes, clientes a coletar e clientes a entregar. Desse modo, todos os pontos de entrega devem ser servidos antes que qualquer ponto de coleta possa ser servido. A demanda não negativa, d_i , a ser entregue ou recolhida, dependendo de seu tipo, é associada com cada cliente i , e o depósito está associado a uma demanda fictícia $d_0 = 0$. Quando a matriz de custo é assimétrica, o problema é chamado de PRVC com Fretes de

Regressos Assimétricos (PRVCFRA). O PRVCFR, assim como o PRVCFRA, consiste em encontrar um conjunto de exatamente k circuitos simples com um custo mínimo, e de tal forma que:

- I. Cada circuito inicie e finalize no vértice de depósito;
- II. Cada cliente (vértice) é visitado por exatamente um circuito;
- III. Os clientes, considerando as operações, são visitados por um circuito de modo que a demanda total de entrega e coleta não exceda, em separado, a capacidade Q do veículo k ;
- IV. Em cada circuito, todos os pontos de entrega devem preceder os pontos de coleta, se houver.

Circuitos contendo somente os clientes de coleta, geralmente não são permitidos. Além disso, a restrição de precedência induz uma orientação nas rotas "mistas" dos veículos, ou seja, as rotas em que existe a necessidade de visitar ambos os tipos de clientes, de entrega e coleta.

K_L e K_B representam o número mínimo de veículos necessários para atender a todos os pontos de entrega e de coleta, respectivamente. Esses valores podem ser obtidos resolvendo os casos do Problema da Mochila Binária (Toth e Vigo, 2002) associado aos subconjuntos de clientes correspondentes. Para garantir a viabilidade, pode-se, por exemplo, assumir que K não é menor do que o número mínimo de veículos necessários para atender a todos os clientes, ou seja, $K \geq \max \{K_L, K_B\}$.

PRVCFR e PRVCFRA são NP-difíceis (Toth e Vigo, 2002; Liu e Chung, 2009; Brandão, 2006), uma vez que generalizam as versões básicas de PRVCS e PRVCA, respectivamente, que surge quando $B = 0$. O caso do PRVCFR, onde as Janelas de Tempo estão presentes, tem sido estudado na literatura e é chamado de PRVC com Fretes de Regressos e Janelas de Tempo (PRVCFRJT).

2.4.4 PRVC com Coletas e Entregas

O PRVC com Coletas e Entregas (PRVCCE), também é uma variação do PRVC onde existe um pedido associado a duas localizações: uma origem, onde uma dada demanda necessita ser coletada e um destino onde a demanda coletada precisa ser entregue. Desse modo, cada rota deve satisfazer um par de restrições de precedência, ou seja, para o pedido, a origem necessita preceder o destino e ambas as localidades devem ser visitadas pelo mesmo veículo (Ropke e Cordeau, 2009).

Com isso, Toth e Vigo (2002) explicam que no PRVCCE, cada cliente i está associado com duas quantidades d_i e p_i , que representam a demanda de produtos homogêneos para ser entregue e recolhido no cliente i , respectivamente. Para cada cliente i , O_i indica o vértice que está na origem da demanda e D_i representa o vértice de destino da demanda recolhida, conforme equação.

O PRVCCE consiste em encontrar um conjunto de exatamente R circuitos simples com um custo mínimo, e de tal forma que:

- Cada circuito inicie e finalize no vértice de depósito;
- Cada vértice de cliente seja visitado por exatamente um circuito;
- A carga corrente do veículo ao longo do circuito não deve ser negativa e nunca pode exceder a capacidade do veículo Q ;
- Para cada cliente i , quando o vértice de origem O_i for diferente do depósito, o vértice de destino D_i deve ser servido no mesmo circuito e depois do vértice O_i ;
- Para cada cliente i , quando o vértice de destino D_i for diferente do depósito, o vértice O_i deve ser servido no mesmo circuito e antes do vértice D_i .

Muitas vezes, a origem ou o destino das demandas são comuns (por exemplo, eles estão associados com o depósito, como no PRVC e PRVCFR). Neste caso, não há necessidade de indicá-los explicitamente. Esse problema é conhecido como o PRVC com Coletas e Entregas Simultâneas (PRVCCES).

PRVCCE e PRVCCES são NP-difíceis (Toth e Vigo, 2002; Ropke e Cordeau, 2009; Nowak et al, 2008) uma vez que generalizam o PRVC, resultantes quando $O_i = D_i = 0$ e $p_i = 0$ para cada $i \in V$. O caso do PRVCCE em que as Janelas de Tempo estão presentes tem sido estudado na literatura e é chamado de PRVC com Coletas e Entregas e Janelas de Tempo (PRVCCEJT).

3 Métodos para Resolução do PRV

Muitos métodos para resolução do PRV foram propostos durante seus mais de 50 anos de estudo, como pode ser visto em Laporte (2009). Esses métodos podem ser classificados em três grupos distintos: Métodos Exatos, Heurísticas e Metaheurísticas.

O PRV pertence à classe de problemas NP-Difíceis (Lenstra e Rinoooy Kan, 1981; Toth e Vigo, 2002; Christofides et al, 1979) o que o torna de difícil resolução por métodos exatos, em função do alto custo computacional.

Segundo Kytöjoki *et al* (2005) e Cordeau *et al* (2002), os métodos exatos têm sido usados para resolver problemas que envolvem em torno de 50 clientes (Laporte, 2009). Dentre eles, vale ressaltar o método de relaxação lagrangeana, onde o problema original tem algumas restrições relaxadas, ou seja, flexibiliza-se algumas restrições. Essas restrições relaxadas são penalizadas através de multiplicadores de Lagrange, os quais são gradativamente ajustados ao longo das iterações. Desse modo, a sequência de soluções gerada cada vez mais se aproxima da solução ótima do problema. Uma essa técnica pode ser analisada em Fisher (1985).

Ainda assim, pode-se citar as técnicas aproximativas. Os algoritmos 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). Segundo Grag e Hochbaum (1997), a busca do equilíbrio entre estas situações conflitantes é o grande paradigma dos algoritmos aproximativos.

Além disso, Laporte e Norbert (1980) e Toth e Vigo (2002) explicam que os métodos exatos propostos podem ser classificados em três tipos: métodos de árvore de busca direta, programação dinâmica e programação linear inteira. Contudo, devido ao alto custo computacional gerado pela execução dos métodos exatos, a maior parte dos métodos propostos para a resolução do PRV baseia-se em heurísticas.

Osman e Laporte (1996) definem heurística como sendo uma técnica que procura boas soluções com um custo computacional razoável, entretanto sem ser capaz de garantir a otimalidade. Segundo Laporte *et al* (2000) e Toth e Vigo (2002) esses métodos realizam uma exploração relativamente limitada no espaço de busca, mas acabam gerando soluções de boa qualidade com um baixo custo computacional.

Ao longo do tempo, diversas heurísticas têm sido propostas para a resolução do PRV. Laporte *et al* (2000) dividem as heurísticas em duas classes principais: heurísticas clássicas e Metaheurísticas. Alguns trabalhos (Christofides *et al*, 1979; Laporte e Semet, 1998; Marinakis e Migdalas, 2002; Toth e Vigo, 2002) classificam as heurísticas clássicas em três tipos: métodos construtores, métodos de duas fases e métodos de melhoramento. Também é comum encontrar trabalhos (Gendreau, Hertz e Laporte, 1994; Rego, 1998; Breedam, 1995) em que consta, também um quarto método: os métodos de otimização incompleta. As heurísticas clássicas vêm sendo utilizadas como método para resolução do PRV durante os últimos 40 anos, entretanto, somente a partir das últimas duas décadas diversas metaheurísticas começaram a ser utilizadas como solução para este problema (Simas, 2007; Laporte, 2009).

O estudo das metaheurísticas foi intensificado a partir dos anos 90 e desde então elas vêm sendo aplicadas com sucesso na resolução de problemas de otimização combinatória (Toth e Vigo, 2002; Laporte, 2009). Glover e Kochenberger (2003) definem metaheurística como sendo um método de solução que orquestra uma interação entre procedimentos de melhoria local e estratégias de alto nível para criar um processo capaz de escapar de ótimos locais e produzir uma busca robusta no espaço de soluções. Em outras palavras, metaheurística pode ser definida como um processo iterativo que guia uma heurística subordinada, através da combinação inteligente de diferentes conceitos provenientes de fenômenos naturais e exploração do espaço de busca, usando estratégias de aprendizado para estruturar as informações com o objetivo de se obter soluções eficientes perto da solução ótima (Osman e Laporte, 1996). A grande vantagem das metaheurísticas é que elas conseguem deixar um ótimo local, através da execução de movimentos que causem uma deterioração da função objetivo (Breedam, 2001; Glover e Kochenberger, 2003), a fim de obter uma exploração aprofundada nas regiões mais promissoras do

espaço de solução. Estes métodos, normalmente, combinam regras sofisticadas de busca em vizinhança, estruturas de memória, e recombinações das soluções. Segundo Glover e Kochenberger (2003) e Toth e Vigo (2002), a qualidade das soluções produzidas por estes métodos é muito melhor do que a obtida pelas heurísticas clássicas, contudo ocorre o aumento do tempo computacional. Além disso, os procedimentos geralmente são dependentes do contexto do problema e exigem o ajuste correto de uma série de parâmetros, o que pode tornar a sua extensão a outras situações mais difícil.

Diversas metaheurísticas têm sido propostas para a resolução do PRV. Entre as metaheurísticas que Gendreau *et al* (1999) destacam como aplicáveis na resolução do PRV estão: 1) *Simulated Annealing* (Osman, 1993; Breedam, 1995; Chiyoshi e Galvão, 2000); 2) *Deterministic Annealing* (Dueck e Scheurer, 1990; Dueck, 1993); 3) Busca Tabu (Osman, 1993; Taillard, 1993; Gendreau, Hertz e Laporte, 1994; Rochat e Taillard, 1995; Xu e Kelly, 1996; Rego e Roucairol, 1996; Barbarasopglu e Ozgur, 1999; Cordeau *et al*, 2002; Toth e Vigo, 2003; Brandão, 2006; Montané e Galvão, 2006; Bolduc *et al*, 2010); 4) Algoritmos Genéticos (Breedam, 1995; Geiger, 2008; Nallusamy, 2009; Dridi *et al*, 2010; Hu e Di Paolo, 2007); 5) Colônia de Formigas (Kawamura *et al*, 1998; Chen e Ting, 2005; Chen e Ting, 2009) e 6) Redes Neurais (Matsuyama, 1991; Nallusamy, 2009). Breedam (2001), Baker e Ayechev (2003) destacam *Simulated Annealing* e Busca Tabu como as metaheurísticas mais eficientes na resolução do PRV e entre elas Cordeau *et al* (2002), Toth e Vigo (2003), Prins (2004), Xu e Kelly (1996) afirmam que a Busca Tabu é a Metaheurística de maior destaque nessa área.

Para uma visão geral da evolução do PRV é sugerida a leitura de Laporte (2009) onde é apresentado um *survey* dos mais de 50 anos de estudos do PRV. Para uma revisão mais aprofundada sobre metaheurísticas recomenda-se a leitura de Osman e Laporte (1996), Rayward-Smith *et al* (1996), Aarts e Lenstra (1997) e Glover e Kochenberger (2003). Ainda assim, são sugeridos, para uma revisão sobre as metaheurísticas aplicadas ao PRV, os trabalhos de Gendreau, Laporte e Potvin (1999), Cordeau *et al* (2002), Toth e Vigo (2002) e Tarantilis *et al* (2005).

A seguir, as três principais abordagens, métodos exatos, heurísticas e metaheurísticas, serão mais bem explanadas, de modo a prover uma visão geral de

alguns métodos e um aprofundamento em técnicas abordadas neste estudo, como as heurísticas de melhoria *Intra-route* e *Inter-route* e vizinho mais próximo e as metaheurísticas Busca Tabu e Algoritmos Genéticos.

3.1 Métodos Exatos

Nos últimos 50 anos, os métodos exatos para o PRV têm envolvido desde os esquemas básicos de *branch-and-bound* até técnicas de engenharia altamente sofisticadas baseadas em programação matemática (Toth e Vigo, 2002). Neste trabalho é feita uma breve apresentação de alguns métodos exatos propostos para a resolução do PRV. Para uma revisão mais detalhada é sugerida a leitura de Christofides et al (1979), Laporte e Nobert (1980), Laporte (1992a) e Toth e Vigo (2002).

Segundo Laporte e Norbert (1980) e Toth e Vigo (2002), os métodos exatos propostos podem ser classificados em três grandes grupos: métodos de árvore de busca direta, programação dinâmica e programação linear inteira. São comuns trabalhos da literatura (Laporte, 1992; Rego e Roucairol, 1996; Xu e Kelly, 1996; Cordeau *et al*, 2002; Tarantilis, Ioannou e Prastacos, 2005; Iori et al, 2007; Archetti et al, 2007; Laporte, 2009) afirmarem que os métodos exatos são indicados apenas para apenas poucas centenas de clientes, pois tendem a tornar-se ineficientes à medida que a quantidade de clientes aumenta.

Mesmo com os problemas de desempenho, na literatura sobre o PRV, é possível encontrar diversos trabalhos que utilizam esses métodos. Laporte e Nobert (1980), Laporte (1992a) apresentam um *survey* com diversos métodos exatos para resolução do PRV baseados principalmente em algoritmos de *branch-and-bound*, relaxações diversas, programação dinâmica e algoritmos para formulações de fluxo de veículos. Christofides e Elion (1969) apresentam um dos primeiros algoritmos de *branch-and-bound* para o PRV, onde o problema é formulado como PCV e tem seus limites inferiores calculados a partir de árvores de expansão mínima. Christofides, Mingozzi e Toth (1981) apresentam um algoritmo de *branch-and-bound* utilizando relaxações do tipo *k-degree center tree* e q-rotas, Fisher (1994) apresenta outro

algoritmo de *branch-and-bound* utilizando relaxação do tipo *k-trees* e Hadjicostantinou *et al* (1995) também apresentam um algoritmo de *branch-and-bound* utilizando relaxação do tipo particionamento de conjuntos.

O trabalho de Ralph *et al* (2001) apresenta um algoritmo de *branch-and-cut* construído a partir de uma metodologia de separação baseada em decomposição para as restrições de capacidade. Toth e Vigo (2002) descrevem detalhadamente o PRV capacitado no caso simétrico e assimétrico. No caso assimétrico, o problema é modelado através de formulações de fluxo de veículos com limites inferiores calculados a partir de modelos de designação, arborescências e fluxo de custo mínimo, e no caso simétrico, é apresentada uma formulação de fluxo de veículos com duplo índice e relaxações baseadas em árvore de expansão mínima, *b-matching* e no problema de partição de conjuntos. Lysgaard *et al* (2004) apresentam um algoritmo de *branch-and-cut* para o PRVC onde, segundo os autores, conseguem resolver, de forma ótima, três instâncias de problemas. Martinhon, Lucena e Maculan (2004) apresentam um algoritmo de *relax-and-cut* para resolver o PRV formulado como uma relaxação da *k-tree*, estendendo o trabalho de Fisher (Fisher, 1994). Archetti *et al* (2007) apresentam um modelo de Programação Inteira Mista para determinar a quantidade a ser entregue e a melhor rota para realizar essa entrega utilizando uma técnica de *branch-and-cut* para resolver o problema de forma ótima.

Segundo Toth e Vigo (2002) e Taha (2007), o primeiro algoritmo de *branch and bound* foi apresentado em 1960 por A. Land e G. Doig para os problemas de programação inteira e programação inteira mista e são, basicamente, procedimentos de enumeração. Seu funcionamento se dá pela seguinte forma: supondo que exista uma função objetivo que se deseje minimizar e um limite superior para esta função esteja disponível, o primeiro passo é dividir o conjunto de todas as soluções viáveis em diversos subconjuntos e para cada um deles obter um limite inferior. Os subconjuntos cujos limites ultrapassam o limite superior são excluídos. Dos subconjuntos restantes, um deles é escolhido e novamente dividido em diversos subconjuntos. Este procedimento continua até que seja encontrada uma solução viável tal que o valor correspondente da função objetivo não seja maior que o limite inferior para qualquer subconjunto (Simas, 2007).

A seguir, é apresentada uma série de aplicações de algoritmos exatos. O algoritmo Christofides e Elion (1969) usa relaxação baseada em árvores de expansão mínima para calcular os limites inferiores nos nodos e antes do corte, verifica se qualquer uma das 3 condições é verdadeira: (1) a carga total do veículo excede a capacidade, (2) a distância total do veículo excede o limite ou (3) a capacidade restante dos veículos não é capaz de atender às cidades ainda não visitadas. Christofides, Mingozzi e Toth (1981) apresentam um algoritmo de *branch-and-bound* baseado nas técnicas de árvore de expansão mínima e q-rotas associados ao uso de relaxação lagrangeana para gerar os limites inferiores. Para calcular os limites inferiores baseados em árvore de expansão mínima, os autores modelam o PRV através de sua generalização, o *m*-PCV, sugerindo que uma solução para o *m*-PCV seria um limite inferior para o PRV. Os autores denominam de "*k-degree center tree*" (k-DCT) uma solução para o *m*-PCV. Lysgaard et al (2004) apresentam uma técnica de *branch-and-cut* totalmente baseada na chamada formulação de dois índices para o PRVC Simétrico. O modelo consiste em x_{ij} representando o número de vezes que um veículo passa entre os vértices i e j . $V_c \subseteq V$ denota um conjunto de clientes. Dado um conjunto $S \subseteq V_c$, seja $q(S)$ denotado por $\sum_{i \in S} q_i$, $\delta(S)$ denotado pelo conjunto de arestas em G com exatamente um fim nos vértices de S . $E(S)$ denotado por um conjunto de arestas em G com todos os fins em vértices de S , e $r(S)$ denotado pelo número de vértices requeridos para servir os clientes em S onde $r(S)$ é a solução ótima do PMB com capacidade Q e o tamanho dos itens dados pela demanda dos clientes em S . Por fim, dado um $F \subseteq E$ arbitrário e $x(F)$ irão denotar $\sum_{e \in F} x_e$. O modelo de programação inteira é dado por:

$$\sum_{e \in E} c_e x_e \tag{3.1}$$

Sujeito a:

$$x(\delta(i)) = 2 \quad i = 1, \dots, n \tag{3.2}$$

$$x(\delta(S)) \geq 2r(S) \quad S \subseteq V_c; \quad |S| \geq 2 \tag{3.3}$$

$$x_{ij} \in \{0, 1\} \quad 1 \leq i < j \leq n \tag{3.4}$$

$$x_{ij} \in \{0, 1, 2\} \quad i = 0; \quad j = 1, \dots, n \tag{3.5}$$

A equação (3.1) representa a função objetivo. A equação (3.2) garante que cada cliente seja visitado exatamente uma vez. A inequação (3.3) impõe que a restrição de capacidade do veículo e as rotas estejam conectadas. As inequações (3.4) e (3.5) são as condições de integralidade.

O algoritmo de *branch-and-cut* apresentado funciona, através de uma fase de separação, onde os nodos são dispostos em forma de árvore e uma segunda parte que consiste na ramificação.

Caso a primeira parte não obtenha uma solução inteira, realiza-se a ramificação, a qual consiste em selecionar vários conjuntos de candidatos $S \subseteq V_c$ para cada $x^*(\delta(S))$, através de uma heurística gulosa e ordenando-os em ordem crescente, conforme a equação (3.6), gerando uma lista de conjuntos S_1, S_2, \dots ordenados. Depois disso, para cada conjunto S_i , analisados individualmente, calcula-se os dois limites inferiores, LB_i^1 e LB_i^2 , dos quais deveriam ser obtidos os dois nós descendentes caso for utilizado S_i para ramificação. Se cada nó descendente puder ser descoberto, S_i é escolhido imediatamente. Caso contrário, aplica-se a seguinte regra: seja $LB_i^- = \min\{LB_i^1, LB_i^2\}$ e $LB_i^+ = \max\{LB_i^1, LB_i^2\}$. Opta-se S_i a S_j se $LB_i^- > LB_j^-$. Nos casos de laços, opta-se S_i a S_j se $\lceil LB_i^+ \rceil > \lceil LB_j^+ \rceil$.

$$\frac{|x^*(\delta(S)) - z|}{q(S)}$$

(3.6)

Na seleção dos nós, o nodo com o menor limite inferior é sempre processado na sequência e a sistemática de cortes utilizada nesse algoritmo consiste em armazenar somente os cortes que poderão gerar, futuramente, uma possibilidade de reconstrução da programação linear.

Com esse algoritmo, os autores puderam provar a otimalidade de três problemas pela primeira vez, além de um comparativo com diversos outros autores (Lysgaard, 2004). Para maiores informações, sugere-se a leitura do trabalho original.

3.2 Heurísticas

Toth e Vigo (2002) e Laporte (2009) abordam que diversas heurísticas têm sido propostas para o PRV e que estas podem ser divididas em duas classes principais: heurísticas clássicas que se desenvolveram, sobretudo, entre 1960 e 1990 e metaheurísticas, cujo crescimento tem ocorrido nas últimas décadas. Muitos dos métodos de construção e melhoria em uso, atualmente, pertencem à primeira classe. Como comentado, anteriormente, esses métodos realizam uma exploração relativamente limitada no espaço de busca, mas acabam gerando soluções de boa qualidade com um baixo custo computacional. Além disso, a maioria deles pode ser facilmente estendida para abordar diversas restrições encontradas no contexto da vida real, fazendo com que ainda sejam amplamente utilizados em pacotes comerciais (Toth e Vigo, 2002).

As heurísticas para o PRV podem ser classificadas em três categorias: Heurísticas Construtoras, Heurísticas de Duas Fases e em Métodos de Melhoramento. As Heurísticas Construtoras, gradualmente, constroem uma solução viável enquanto avaliam o custo da solução, contudo não possuem uma fase de melhoria, são, por exemplo, os algoritmos de Clarke e Wright (1964), Mole e Jamenson (1976). As Heurísticas de Duas Fases decompõem o problema em dois componentes, agrupamento dos vértices em rotas viáveis e construção das rotas, com possibilidades de ciclos entre os dois estágios. As Heurísticas de Duas Fases são divididas em duas classes. Na primeira classe, os vértices são primeiramente organizados em agrupamentos viáveis e então, uma rota é criada para cada um deles. Na segunda classe, uma rota é criada com todos os vértices e após são agrupados de modo a gerar diversas rotas viáveis. Dentre as Heurísticas de Duas Fases podem ser encontradas as heurísticas de Gillet e Miller (1974) e Fisher e Jaikumar (1981). Por fim, as Heurísticas de Melhoramento, buscam evoluir qualquer solução viável através da troca ou exportação de clientes entre as rotas. Este tipo de método pode ser usado para melhoramento de uma única rota (*intra-route*) ou de várias rotas simultaneamente (*inter-route*).

Segundo Laporte e Semet (1998), Laporte *et al* (2000) e Toth e Vigo (2002) a maioria dos métodos de melhoramento *intra-route* podem ser descritos em termos

dos mecanismos λ -opt de Lin (1965). Quanto aos trabalhos de melhoramento *inter-route*, Breedam (1995) faz uma análise e considera quatro tipos possíveis de operações a serem feitas: *String Cross*, *String Exchange*, *String Relocation* e *String Mix*. Outras referências deste tipo de heurística podem ser encontradas no trabalho de Thompson e Psaraftis (1993). As heurísticas de melhoramento são também conhecidas como Heurísticas de Busca Local (Marinakis e Migdalas, 2002; Thangiah e Petrovic, 1997; Laporte, 2009).

A seguir, são apresentadas as principais heurísticas clássicas segundo Toth e Vigo (2002) e Christofides (1979). Dentre elas, as heurísticas de melhoramento *intra-route* e *inter-route* e do vizinho mais próximo, são utilizadas no modelo que será abordado neste estudo.

3.2.1 Algoritmo de Clarke e Wright

Esta, talvez, seja a heurística mais conhecida para resolução do PRV (Thangiah e Petrovic, 1997; Laporte *et al*, 2000; Christofides *et al*, 1979; Toth e Vigo, 2002). O algoritmo é baseado na noção de economias. Quando duas rotas $(0, \dots, i, 0)$ e $(0, j, \dots, 0)$ podem ser unidas em uma única rota viável $(0, \dots, i, j, \dots, 0)$, uma economia de distância $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ é obtida. Devido as suas características, esse algoritmo é naturalmente aplicado a problemas nos quais o número de veículos é uma variável de decisão, e funciona igualmente bem para problemas direcionados e não direcionados. O algoritmo pode ser sequencial ou paralelo (Toth e Vigo, 2002) e baseia-se em calcular as economias, conforme equação (3.7).

$$s_{ij} = c_{i0} + c_{0j} - c_{ij} \quad \text{para } i, j = 1, \dots, n; i \neq j \quad (3.7)$$

Com isso, as economias calculadas são ordenadas de forma decrescente e os clientes i e j que têm as maiores economias são unidos enquanto que a restrição de capacidade não seja ultrapassada. O algoritmo pára quando nenhuma união pode mais ser realizada.

3.2.2 Algoritmo de Mole e Jamenson

O algoritmo de Mole e Jamenson é um procedimento de inserção sequencial de dois parâmetros λ e μ para expandir uma rota em construção. A equação (3.8), define o critério de proximidade, onde c_{ik} representa o custo entre os nodos i e k , c_{kj} o custo entre k e j e c_{ij} o custo entre i e j .

$$\alpha(i, k, j) = c_{ik} + c_{kj} - \lambda c_{ij} \quad (3.8)$$

Com a equação (3.9), seleciona-se o vértice que está mais próximo da rota atual. O critério de economia visa selecionar qual o melhor local para se inserir o vértice selecionado.

$$\beta(i, k, j) = \mu c_{0k} - \alpha(i, k, j) \quad (3.9)$$

onde c_{0k} representa a distância entre o depósito e o vértice k . O local a ser selecionado é o que apresentar a maior economia.

3.2.3 Algoritmo de Gillet e Miller

Também conhecida como o algoritmo das varreduras (Thangiah e Petrovic, 1997), esta heurística procura obter a solução em duas etapas distintas. Primeiro os pontos de demanda são agrupados segundo algum critério de proximidade, formando assim *clusters*, depois solucionar cada *cluster*, independentemente, criando uma rota para estes clientes (Goldbarg e Luna, 2000).

Esta heurística usa, além das distâncias entre os clientes, os ângulos das coordenadas polares para construir as rotas. Os clientes são organizados numa ordem crescente (ou decrescente) dos seus ângulos em coordenadas polares. O algoritmo inicialmente forma os *clusters* unindo os clientes sem violar a restrição da capacidade. Num segundo momento são criadas rotas para cada um destes *clusters*. A criação destas rotas pode ser feita através de algoritmos para resolução do PCV ou usando-se algoritmos exatos (Thangiah e Petrovic, 1997).

3.2.4 Heurística de Melhoria *Intra-Route*

Este tipo de heurística é aplicado às rotas individualmente. Estes métodos fazem uma mudança na ordem em que os clientes são visitados dentro de cada rota (Thangiah e Petrovic, 1997), na tentativa de diminuir a distância ou tempo da rota (Breedam, 1995). Segundo Laporte *et al* (2000) e Laporte e Semet (1998) a maioria dos métodos existentes podem ser descritos em termos dos mecanismos λ -opt de Lin (1965). Nesse caso, λ arestas são removidas do *tour* e as λ restantes são reconectadas de todas as possíveis maneiras. Se algumas destas novas conexões proporcionar um custo menor, ou seja, se melhorar a rota, este movimento é executado. Para Thangiah e Petrovic (1997) os métodos mais comuns são os métodos 2-opt e 3-opt. O método 2-opt remove duas arestas de uma rota inicial e então cria duas novas arestas diferentes ligando os clientes que ficaram desconectados. O método 3-opt segue o mesmo procedimento, entretanto 3 arestas são removidas.

3.2.5 Heurística de Melhoria *Inter-Route*

Este tipo de heurística considera possíveis trocas ou realocações de clientes entre duas rotas (Breedam, 1995), sempre respeitando as restrições associadas.

Thompson e Parafitis (1993) apresentam um esquema “*b-cyclic, k-transfer*” no qual uma permutação circular de b rotas é considerada e k clientes de cada rota são transferidos para a próxima rota do ciclo da permutação (Laporte *et al*, 2000). Breedam (1995) classifica os movimentos como *String Cross*, *String Exchange*, *String Relocation* e *String Mix* e, segundo Laporte *et al* (2000) e Laporte e Semet (1998) estes movimentos podem ser vistos como casos especiais da troca *2-cyclic*.

Os algoritmos “*b-cyclic, k-transfer*” são algoritmos de duas fases. Na primeira é realizado um agrupamento e na segunda são realizadas as mudanças nas ligações dos vértices. A Figura 3.1, apresenta um exemplo dessa técnica.

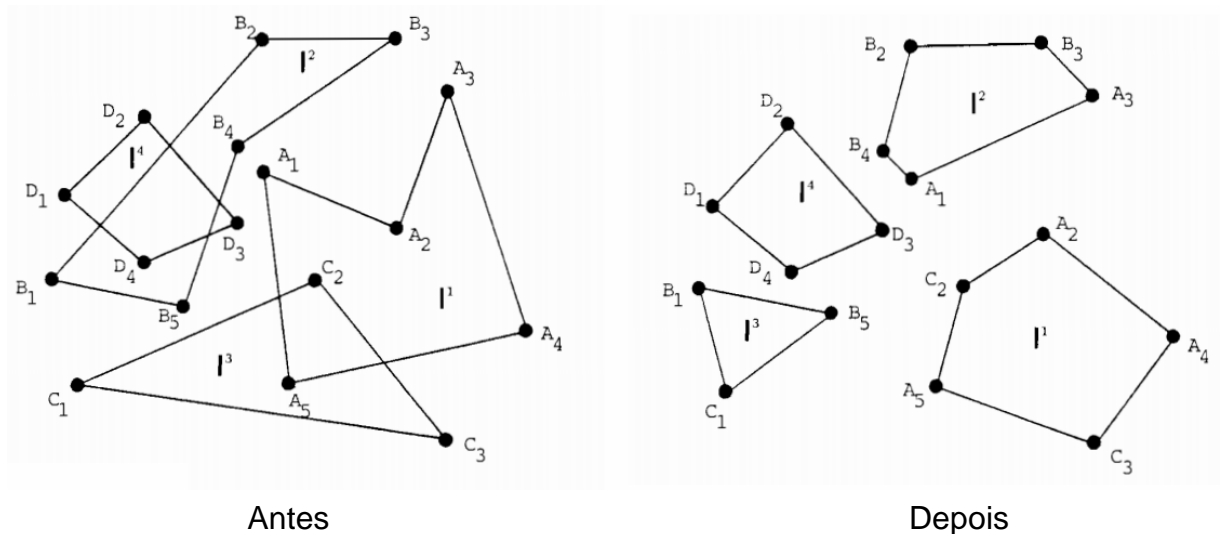


Figura 3.1. Representação gráfica do método “*b-cyclic, k-transfer*”
 Fonte: Thompson e Psaraftis (1993)

3.2.6 Heurística do Vizinho mais Próximo

O trabalho de Tyagi (1968) apresenta o método do Vizinho mais Próximo para a resolução do PRV. Este procedimento deve, a partir de um dado vértice, visitar o vértice mais próximo que ainda não foi visitado e retornar ao vértice inicial quando todos os outros tiverem sido visitados. Segundo Cook *et al* (1998), a vantagem deste método é que ele é fácil de ser implementado, rápido, e geralmente produz *tours* de qualidade razoável.

3.2.7 Procedimentos de Inserção

Este método inicia com uma sub-rotas unindo k vértices, então adiciona os vértices restantes um a um conforme a política escolhida, sendo que esta pode assumir diversas variações, como por exemplo:

- Inserção do Mais Distante: neste caso, inicia-se um *tour* com dois vértices que estão no final de uma aresta de alto custo. Para cada vértice não inserido v , é calculado o custo mínimo entre v e qualquer vértice no *tour* construído. Então é escolhido como o próximo a ser

inserido aquele para o qual o custo for máximo (Cook *et al*, 1998; Reinelt, 1994);

- Inserção do Mais Próximo: inicia-se o subgrafo com apenas um vértice, encontra-se outro para formar o *subtour* $i-k-i$, cujo custo c_{ik} seja mínimo. Dado este *subtour* seleciona-se o próximo vértice como sendo o mais próximo a qualquer vértice no *subtour* e o insere entre $i-k$ (Cook *et al*, 1998);
- Inserção do Mais Barato: semelhante à inserção do mais próximo, com exceção de que o próximo vértice a ser inserido não precisa ser o mais próximo a qualquer um do *subtour* já existente (Cook *et al*, 1998; Reinelt, 1994);
- Inserção Arbitrária: também inicia com um subgrafo, porém o próximo vértice a ser inserido é escolhido aleatoriamente (Bodin, Golden, Assad e Ball, 1983).

3.3 Metaheurísticas

As metaheurísticas podem ser classificadas em busca local, busca em população e mecanismos de aprendizagem, sendo que grande parte das metaheurísticas propostas para o PRV são do primeiro tipo (Laporte, 2009; Toth e Vigo, 2002).

Como é citado, metaheurísticas podem ser definidas como sendo um método de solução que orquestra uma interação entre procedimentos de melhoria local e estratégias de alto nível para criar um processo capaz de escapar de ótimos locais e produzir uma busca robusta no espaço de soluções (Glover e Kochenberger, 2003). Devido a essas características, as melhores metaheurísticas conhecidas, desenvolvidas para o PRV, normalmente identificam melhores ótimos locais do que as heurísticas anteriores, mas elas também tendem a consumir mais tempo.

Muitos trabalhos podem ser encontrados na literatura apresentando a aplicação de metaheurísticas na abordagem do PRV. Na Tabela 3.1 podem ser analisados trabalhos relacionados com as metaheurísticas utilizadas.

Tabela 3.1 – Trabalhos relacionados com metaheurísticas

Metaheurística	Trabalhos Relacionados
Busca Tabu	Osman (1993), Taillard (1993), Rochat e Taillard (1995), Xu e Kelly, (1996), Rego e Roucairol (1996), Gendreau <i>et al</i> (1998), Barbarosoglu e Ozgur (1999), Cordeau <i>et al</i> (2002), Toth e Vigo (2003), Desaulniers <i>et al</i> (2008), Easwaran e Uster (2009), Bolduc <i>et al</i> (2010), Brandão (2010), Benjamin (2010), Paul (2010).
Têmpera Simulada (<i>Simulated Annealing</i>)	Robusté <i>et al</i> (1990), Gendreau <i>et al</i> (1998), Alfa <i>et al</i> (1991), Breedam (1995)
Colônia de Formigas	Chen e Ting (2005), Montemanni <i>et al</i> (2007), Reyes <i>et al</i> (2008) e Chen e Ting (2009).
Algoritmos Genéticos	Baker e Ayechev (2003), Prins (2004), Geiger (2008), Dridi <i>et al</i> (2010).

Laporte (2009) explica que, essencialmente, a busca local explora o espaço de soluções movendo-se, a cada iteração, da solução atual para outra solução em sua vizinhança. Exemplos clássicos incluem Busca Tabu, em inglês *Tabu Search* (Glover, 1986), Tempera Simulada, em inglês *Simulated Annealing* (Kirkpatrick *et al*, 1983), Tempera Determinista, em inglês *Deterministic Annealing* (Dueck and Scheurer, 1990; Dueck, 1993), Busca em Vizinhança Variável, em inglês *Variable Neighbourhood Search* (Mladenovic and Hansen 1997), Busca em Vizinhança Muito Grande, em inglês *Very Large Neighbourhood Search* (Ergun *et al*, 2006) e Busca em Grande Vizinhança Adaptativa, em inglês *Adaptive Large Neighbourhood Search* (Ropke e Pisinger, 2006). Os ingredientes principais da busca local são as regras empregadas para definir a vizinhança de uma solução e os mecanismos criados para explorá-lo. Normalmente, as heurísticas de busca local executam movimentos *inter-route* com reotimizações ocasionais via *intra-route*.

Busca em população trabalha com uma população de soluções. O exemplo mais conhecido desse paradigma são os Algoritmos Genéticos (Holland, 1975). A

cada iteração de um algoritmo genético, algumas soluções pai são extraídas da população inicial e recombinadas para criar uma nova prole, a qual irá substituir os piores indivíduos da população. Ainda podem ser aplicados mecanismos de diversificação, chamados de Mutação, na prole antes de serem incluídas na população de soluções.

Mecanismos de aprendizagem, Segundo Laporte (2009), são bem representados por duas técnicas: Redes Neurais Artificiais e Otimização por Colônia de Formigas. Redes Neurais Artificiais são derivadas de conceitos oriundos da inteligência artificial. Elas podem aprender de experiências e, de modo incremental, ajustar seus pesos de forma iterativa. A aplicação limitada desse conceito para o PRV (Ghaziri, 1991; Schumann e Retzko, 1995) tem encontrado resultados divergentes. A Otimização por Colônia de Formigas busca imitar o comportamento de formigas buscando por alimento, depositando feromônio pelo caminho. Com o tempo, o feromônio se acumula rapidamente nos caminhos mais curtos, os quais são os mais seguidos pela maioria das formigas. Em um algoritmo de Otimização por Colônia de Formigas, a ideia traduz-se em, gradualmente, dar mais peso às arestas que aparecem frequentemente em boas soluções.

A seguir são apresentados alguns trabalhos que utilizam metaheurísticas, visando analisar a forma de implementação das técnicas. Em seguida, é apresentado um estudo das técnicas de Busca Tabu e Algoritmos Genéticos de modo a destacar componentes importantes em cada método.

3.3.1 Trabalhos Relacionados

Segundo Gendreau *et al* (1998), as duas primeiras implementações de *Simulated Annealing* foram os trabalhos de Robusté, Daganzo e Souleyrette (1990) e Alfa, Heragu e Chen (1991). O primeiro trabalho define como estrutura de vizinhança uma combinação de diversos mecanismos tais como trocas de clientes dentro da mesma rota e trocas de clientes entre rotas. O segundo, utiliza a implementação de uma heurística *route first, cluster-second* para gerar uma solução inicial e mecanismos de 3-opt (Lin, 1965) para gerar as vizinhanças.

O estudo de Osman (1993) foi uma das primeiras implementações de sucesso da Busca Tabu para o PRV, pois apresentava uma série de características que aumentaram a qualidade do algoritmo. Dentre essas características pode-se citar: uso de uma melhor solução inicial (com a heurística de Clark e Wright), alguns parâmetros podem ser ajustados, geração de vizinhança mais rica baseada em mecanismos de λ -*interchange* que incluía movimentos 2-opt, alocação de vértices a diferentes rotas e troca de vértices entre rotas. Além disso, Osman (1993) propõe duas versões do algoritmo. Em uma versão, depois que toda a vizinhança foi explorada, o melhor movimento é aceito, na segunda versão, nem toda a vizinhança precisa ser explorada uma vez que é considerado como aceito o primeiro movimento encontrado.

O algoritmo de Taillard (1993) apresenta algumas das características existentes no Taburoute (Gendreau e Potvin, 2005), como a duração randômica do atributo tabu e utilização de técnicas de diversificação. A inovação apresentada por este trabalho se refere à decomposição do problema principal em subproblemas (Laporte *et al*, 2000; Cordeau *et al*, 2002) e cada subproblema é então resolvido independentemente em processadores paralelos. A estrutura de vizinhança é definida através de mecanismos de λ -*interchange*, com $\lambda=1$. A reotimização de rotas individuais é feita através de algoritmos exatos construídos para o PCV.

No trabalho de Gendreau, Hertz e Laporte (1994), o algoritmo denominado *Taburoute* apresenta algumas características que o tornam mais robusto. Uma das características inovadoras é a permissão do uso de soluções inviáveis com relação à capacidade do veículo ou duração da rota a fim de explorar outras regiões do espaço de busca. A maneira como os autores permitiram o uso destas soluções inviáveis é através da introdução de termos de penalidade na função objetivo. A vizinhança de x_t é definida como sendo o conjunto de todas as soluções alcançáveis a partir de x_t através da remoção do vértice v da sua rota atual e inserção dele em outra rota s contendo um dos seus vizinhos mais próximos. Como atributo tabu controla-se a quantidade de vezes que um vértice deve ficar sem ser utilizado. Esta quantidade é definida randomicamente no intervalo [5,10]. Uma estratégia de diversificação que penaliza os vértices que são frequentemente movimentados é utilizada.

O trabalho de Xu e Kelly (1996) apresenta uma estrutura de vizinhança mais sofisticada. Eles consideram trocas entre vértices de duas rotas, uma reposição global de alguns vértices em outras rotas e melhoramentos locais. A estratégia de reposição global resolve um modelo de fluxo de redes para realocar, de maneira ótima, um dado número de vértices em diferentes rotas. As políticas de intensificação e diversificação são baseadas em procedimentos de reinício de busca e em frequência de memória de longo prazo respectivamente. Para o funcionamento da estratégia de intensificação, um *pool* de melhores soluções é mantido e usado para reiniciar a busca. Quando a busca é reiniciada, a solução atual é recuperada deste *pool*, as restrições tabus são relaxadas e os parâmetros reinicializados. A estratégia de diversificação é baseada em penalidades que são associadas aos vértices em função da frequência que eles são designados a rotas específicas. O objetivo é dar preferência aos vértices que aparecem com menor frequência nas rotas.

O algoritmo de Rego e Roucairol (1996) define a estrutura de vizinhança através da utilização de cadeias de ejeção para mover-se de uma solução a outra. Segundo os autores, uma interpretação genérica do termo consiste em sucessivamente selecionar um conjunto de elementos que são designados a um novo estado, de tal maneira que a cada passo a mudança do estado é criada pelo elemento do passo imediatamente precedente. Na terminologia de cadeias de ejeção, cada passo é chamado nível e a mudança do estado de um elemento produzido por outro é denominado de uma ejeção. Desta maneira uma cadeia de ejeção pode ser vista como uma série de níveis z , cada um consistindo de três vértices (u_{s-1}, u_s, u_{s+1}) que aparecem consecutivamente em uma rota. Um z -nível da cadeia de ejeção consiste em substituir a tripla $(u_{s-1}^k, u_s^k, u_{s+1}^k)$, onde $k=0, \dots, 1$ pela tripla $(u_{s-1}^k, u_s^k, u_{s+1}^k)$, onde $k=1, \dots, z$ até que o vértice posicionado no último nível, u_s seja realocado.

Rochat e Taillard (1995) desenvolveram o conceito de Memória Adaptativa. Uma memória adaptativa é um *pool* de boas soluções produzidas por heurísticas que é dinamicamente atualizado, através da adição de novos elementos de alta qualidade e remoção de elementos de menor qualidade (Cordeau *et al*, 2002).

Periodicamente, alguns elementos das soluções são extraídos e combinados diferentemente para produzir novas soluções (Laporte *et al*, 2000).

O trabalho de Barbarosoglu e Ozgur (1999) descreve um algoritmo de Busca Tabu que não contém estratégias de diversificação e apenas soluções viáveis são examinadas. Soluções vizinhas são definidas através de um esquema de λ -*interchange*, que favorece vértices relativamente distantes do centroide de suas rotas correntes e perto do centróide de uma nova rota. Também faz a reotimização de rotas através da execução de procedimentos 2-opt (Gendreau *et al*, 1998).

Toth e Vigo (2003) desenvolveram um algoritmo chamado *Granular Tabu Search* em que a ideia principal é desconsiderar as áreas não promissoras do espaço de busca, através da remoção de arcos que provavelmente não pertençam a uma boa solução devido ao seu alto custo. Daí se origina o nome *granular*, e pode ser vista como uma implementação eficiente de lista de candidatos. Um limite associado ao tamanho máximo dos arcos esperados em uma solução de alto nível é definido e desta maneira arcos superiores a este limite são eliminados. O intuito é restringir drasticamente o espaço de soluções possíveis removendo elementos que tem baixa probabilidade de pertencerem a soluções de boa qualidade. Soluções inviáveis também são permitidas através do uso de penalidades associadas ao excesso de demanda e distância da solução.

Simas e Gómez (2007) abordam um algoritmo de Busca Tabu para o PRVC onde aplicam dois métodos de geração de vizinhança, *intra-route* e *inter-route* apresentados nas seções 3.2.5 e 3.2.6, respectivamente. Além disso, utilizam uma técnica de intensificação para obter um melhor aproveitamento da Busca Tabu, baseada em uma lista de soluções candidatas.

Baker e Ayechev (2003) propõem um algoritmo genético onde a rota que o veículo deve seguir não é especificada claramente. Dado n clientes e m veículos os cromossomos são representados por *strings* de tamanho n , e em cada gene um valor entre 1 e m , associando um veículo para cada cliente. Desta maneira as rotas podem ser consideradas implicitamente especificadas. Para tornar as rotas explícitas, uma solução para o PCV é requerida para cada veículo, e associado a cada indivíduo existe o valor da função de *fitness*. Para gerar a população inicial os autores fazem uso de dois métodos, a heurística de Gillet e Miller (1974) e a

abordagem de Fisher e Jaikumar (1981). Os cromossomos pais são escolhidos aleatoriamente. O processo de reprodução é através do cruzamento de 2 pontos, neste caso dois pontos nos cromossomos pais são escolhidos e dois novos cromossomos são gerados a partir da recombinação dos pedaços das *strings* dos pais. A política de substituição deste algoritmo mantém o tamanho da população constante, com os indivíduos inferiores sendo removidos ao mesmo tempo em que um novo indivíduo que atende ao critério de entrada é aceito. Este critério é baseado na distância total (chamada de *fitness* do indivíduo) e no excesso da capacidade do veículo e distância percorrida (chamada de *unfitness* do indivíduo). Além da operação de recombinação através do cruzamento de dois pontos, dois tipos de mutação são provocados: troca do valor entre dois genes e substituição do veículo pelo valor do veículo vizinho.

O trabalho de Prins (2004) aborda características importantes para este estudo. O autor apresenta um algoritmo genético no qual a solução do PRV Capacitado é dada como uma sequência de n vértices sem delimitação de rotas. Ou seja, o cromossomo apenas representa a ordem em que os clientes devem ser atendidos. Com isso, um procedimento para decompor de maneira ótima a sequência em rotas e gerar a solução é utilizado. Esse tipo de representação será aplicada em um dos módulos do modelo a ser desenvolvido. Ainda assim, o trabalho de Prins (2004) utiliza o algoritmo de Bellman's (1958) sobre um grafo auxiliar para encontrar um caminho de custo mínimo e calcular o custo correspondente da solução.

3.3.2 Busca Tabu

Essa seção é de suma importância nesse estudo, pois a Busca Tabu compõe um dos módulos do modelo a ser implementado e, também, por ser considerada como a melhor Metaheurística para o PRV (Xu e Kelly, 1996; Cordeau *et al*, 2002; Toth e Vigo, 2003; Prins, 2004). As abordagens baseadas em Busca Tabu são capazes de obter soluções de boa qualidade para o PRV. Ao considerar, por exemplo, o conjunto de 14 problemas de Christofides e Elion (1969), a maioria dos

melhores resultados foi obtida através de soluções baseadas na Busca Tabu (Toth e Vigo, 2002; Toth e Vigo, 2003).

O método heurístico Busca Tabu (BT) original foi proposto por Glover em 1986 (Glover, 1986) para diversos problemas de otimização combinatória e consiste em numa rotina iterativa para construir vizinhanças com ênfase na fuga de ótimos locais. Segundo Osman e Laporte (1996), o emprego de estratégias para modificar a vizinhança de uma solução durante o processo de busca permite que se ultrapasse os limites de uma simples heurística de busca local. A BT utiliza estruturas auxiliares de memória de curto e longo prazo, também chamadas de memória baseada em recorrência e frequência, respectivamente, para determinar as novas vizinhanças (Simas, 2007).

Segundo Gendreau et al (2003), a Busca Tabu é uma extensão do método clássico de Busca Local (BL) e, por conseguinte, devido as suas semelhanças com o método de BL, os dois primeiros elementos básicos de uma BT, espaço de busca e a estrutura de vizinhança, são muito semelhantes aos da BL. Contudo, a BT se diferencia em outros aspectos. Seu principal diferencial, a Lista Tabu, daí o nome Busca Tabu, confere a essa técnica a possibilidade de restringir movimentos repetidos e escapar de ótimos locais, proporcionando maior abrangência na varredura do espaço de busca e, por conseguinte, melhores resultados.

A Figura 3.2 mostra um fluxograma representando a dinâmica do algoritmo da Busca Tabu. Esse fluxo serve como base para a formulação do módulo de Busca Tabu que irá compor o modelo desenvolvido, apresentado no capítulo 4.

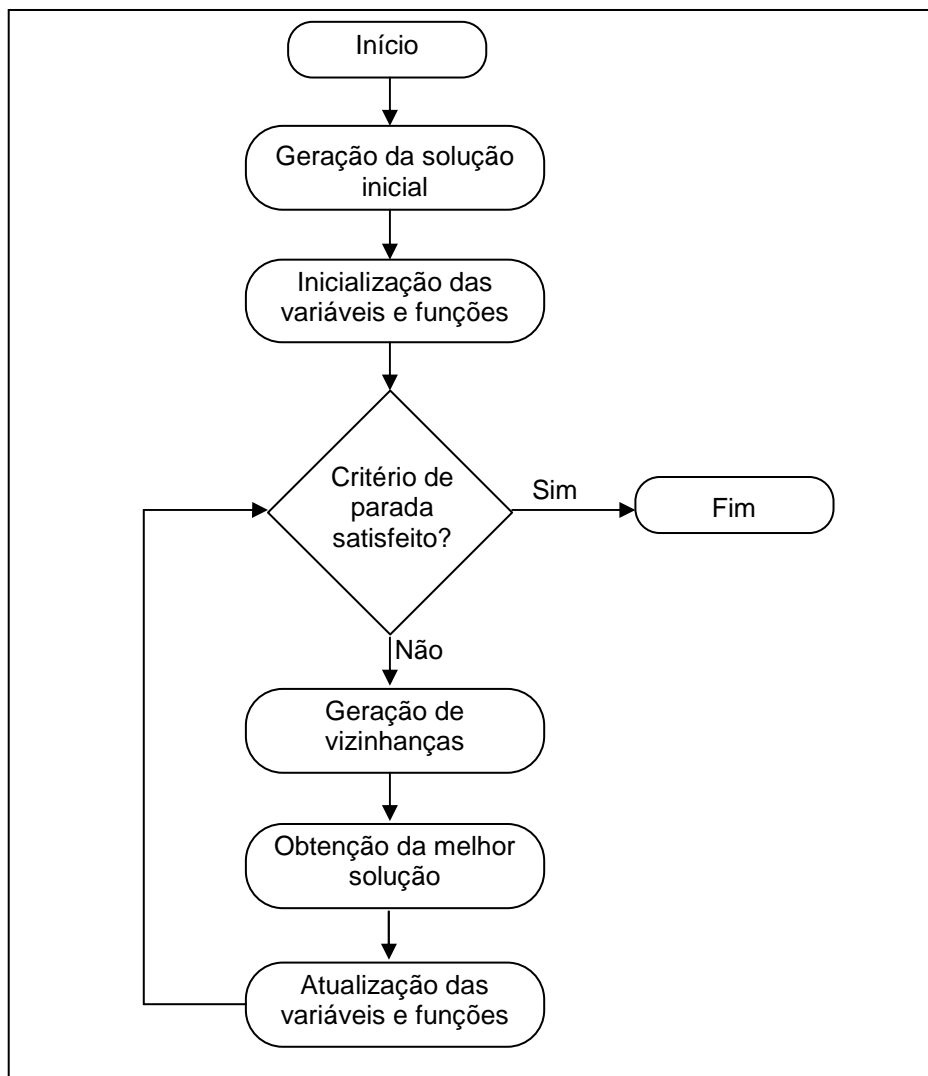


Figura 3.2. Dinâmica do processo de Busca Tabu

O processo de Busca Tabu se inicia na geração da solução inicial. Essa solução inicial deve ser uma solução viável, gerada através de alguma técnica. A inicialização trata da inicialização das variáveis e funções, como, por exemplo, a iteração atual, melhor solução encontrada até o momento e a iteração em que a melhor solução foi encontrada. O Critério de Parada tem a finalidade de determinar quando o algoritmo deve encerrar. A geração de soluções vizinhas consiste na criação de novas soluções através da estrutura de vizinhança, podendo ser uma combinação de diversas técnicas. Em seguida, deve-se obter a melhor solução vizinha, respeitando-se a Lista Tabu e o Critério de Aspiração para um movimento considerado Tabu. Por fim, a atualização das variáveis de controle tem o objetivo de definir, por exemplo, qual a melhor solução encontrada e em que iteração isso aconteceu.

Segundo Gendreau et al (2003), a BT, apresentada acima, pode algumas vezes resolver problemas difíceis de modo satisfatório, mas na maioria dos casos alguns elementos necessitam ser incluídos na estratégia de busca para melhorar a eficiência do algoritmo. Os principais elementos que compõem uma Busca Tabu serão apresentados a seguir. Alguns apontamentos, quanto à utilização de alguns elementos neste estudo, serão realizados durante a explanação desses elementos, maiores informações são apresentadas na seção 4.3.3.

3.3.2.1 Espaço de Soluções

Segundo Gendreau et al (2003), o espaço de busca de uma BT é simplesmente o espaço de todas as possíveis soluções que podem ser consideradas durante a busca. No caso do PRVC, por exemplo, o espaço de soluções pode ser definido como o conjunto de soluções viáveis do problema, onde cada ponto no espaço de soluções corresponde a um conjunto de rotas que satisfazem todas as restrições.

3.3.2.2 Estrutura de Vizinhaça

A definição de estrutura de vizinhaça está intimamente ligada a definição de espaço de soluções. A cada iteração de uma BT, as transformações locais que podem ser aplicadas na solução corrente, denotada por S , definem um conjunto de soluções vizinhas no espaço de busca, denotado por $N(S)$ (a vizinhaça de S). Formalmente, $N(S)$ é um subconjunto do espaço de busca. Um exemplo de estrutura de vizinhaça para o PRVC envolve movimentar, a cada iteração, um único cliente de sua rota atual; o cliente selecionado é inserido na mesma ou em outra rota que possua capacidade para atendê-lo. Uma importante vantagem dessa estrutura de vizinhaça está na forma de inserção do cliente na rota: o cliente pode ser inserido em uma posição aleatória ou pode ser inserido na melhor posição da rota, através de diversas técnicas (Gendreau et al, 2003). Diversas estruturas de vizinhaça

podem ser aplicadas, desde as mais simples, como 3-opt (Lin, 1965), até as mais complexas, para o PRVC, como é o caso da λ -interchange de Osman (1993), onde as soluções vizinhas podem ser obtidas através de movimentos simultâneos de clientes para diferentes rotas e a troca entre clientes nas rotas.

3.3.2.3 Lista Tabu

A Busca Tabu trabalha com estruturas de memória, entre as quais a mais usada é a Lista Tabu (LT). Nessa lista ficam armazenados os últimos movimentos reversos realizados, com a função de evitar que a busca fique presa em pontos de mínimos ou máximos locais, provocados, por exemplo, pela ocorrência de ciclos. Os movimentos armazenados na Lista Tabu ficam proibidos por um determinado número de iterações, onde, geralmente, somente uma pequena e limitada quantidade de informações são gravadas (Gendreau et al, 2003). Mesmo assim, podem ser armazenadas soluções completas, contudo essa forma exige um grande espaço de armazenamento e torna a verificação de um movimento Tabu mais demorada e custosa. Segundo Gendreau et al (2003), a maior parte da utilização da Lista Tabu envolve o armazenamento de um número limitado das últimas transformações realizadas na solução corrente e proibindo um movimento reverso. Para o PRVC pode-se citar, por exemplo, um cliente v_1 saindo da rota R_1 e indo para a rota R_2 onde o movimento de retorno, de R_2 para R_1 , pode ser armazenado formando uma tripla (v_1, R_2, R_1) na Lista Tabu. Esse tipo de representação não restringe muito o espaço de busca, possibilitando um ciclo se o cliente v_1 for de R_2 para outra rota R_3 e então retornar a R_1 . Gendreau et al (2003) reforça que uma Busca Tabu forte poderia armazenar somente uma dupla (v_1, R_1) impossibilitando que v_1 retorne a R_1 , de qualquer modo ou também armazenando somente v_1 , impossibilitando v_1 de ser movido enquanto o mesmo se encontrar na Lista Tabu.

Dentre as três formas de implementação de Lista Tabu expostas, neste estudo, utiliza-se o armazenamento de uma dupla, de modo que a Lista Tabu não seja tão restritiva, possibilitando a oscilação desse parâmetro em futuros testes, visto que também não permita facilmente a ocorrência de ciclos.

3.3.2.4 Critério de Aspiração

A Lista Tabu pode acabar sendo muito restritiva e proibir movimentos, mesmo que não ocorram ciclos ou que levem a uma estagnação da busca. Para resolver esse problema é necessária uma ferramenta que anule o *status* Tabu de um movimento. Essa ferramenta é conhecida como Critério de Aspiração (CA) e, basicamente, permite que um movimento Tabu seja realizado, desde que atenda um dos critérios pré-definidos. O Critério de Aspiração pode ser definido de diversas formas, seja, por exemplo, por simplesmente melhorar a função objetivo ou por promover uma diversificação, forçando movimentos Tabus, onde, dadas as circunstâncias, irão direcionar a busca para um espaço ainda não explorado. Neste estudo, o Critério de Aspiração somente permite que um movimento Tabu seja realizado se este melhorar a função objetivo.

3.3.2.5 Critério de Parada

Na teoria, a Busca Tabu poderia ser executada interminavelmente a não ser que se conhecesse a solução ótima *a priori*. Contudo, na prática, a busca deve ser interrompida em algum ponto, sendo abaixo apresentados os principais critérios de parada, segundo Gendreau et al (2003):

- Depois de um número fixo de iterações (ou de tempo de CPU);
- Depois de um determinado número de iterações sem melhora da função objetivo (critério mais utilizado na literatura, também conhecido como Nbmax);
- Quando a função objetivo atingir um determinado valor, determinado *a priori*.

Em um esquema de Busca Tabu mais complexo, a parada pode ser definida depois do final de todas as etapas e cada etapa pode ser finalizada, por exemplo,

por um dos critérios acima. Neste estudo, é utilizado o Nbmax como critério de parada.

3.3.2.6 Política de Intensificação

Conforme comentado anteriormente, a Busca Tabu pode necessitar de mais alguns elementos para obter melhores resultados (Gendreau et al, 2003). Um desses elementos é a intensificação. O conceito por trás de uma política de intensificação consiste em explorar profundamente algumas partes do espaço de soluções que parecem promissoras para tentar buscar a melhor solução nessa área. De modo geral, a intensificação está baseada em uma memória de médio prazo onde é registrado, por exemplo, o número de iterações que várias partes da solução estiveram presentes na solução atual sem interrupção. Uma política de intensificação é utilizada neste estudo, a qual está detalhada na seção 4.3.3.4.

3.3.2.7 Política de Diversificação

Outro elemento que Gendreau et al (2003) apontam é a política de diversificação. Um dos principais problemas de todos os métodos baseados em Busca Local, incluindo a Busca Tabu, é que eles tendem a ser “locais”, eles tendem a “gastar” muito tempo de processamento em uma parte restrita do espaço de soluções. Para lidar com esse problema, pode-se utilizar uma política de diversificação que força a busca em áreas ainda não exploradas, sendo geralmente baseada em uma memória de longo prazo (Toth e Vigo, 2002). No caso do PRVC, por exemplo, podem ser armazenadas o número de vezes que cada cliente foi movido de sua rota atual, sendo possível, nesse caso, identificar regiões do espaço de solução que ainda não foram exploradas.

3.3.3 Algoritmos Genéticos

O termo Algoritmo Genético (AG) foi inicialmente apresentado por Holland (1975) no livro *Adaptation in Natural and Artificial Systems*. A motivação original do AG foi uma análise biológica. Na reprodução seletiva de animais e plantas, por exemplo, a prole possui certas características desejáveis, dependendo do nível genético proveniente da combinação resultante dos cromossomos de seus pais. No caso do AG, uma população de indivíduos é utilizada, esses indivíduos são referenciados na literatura como cromossomos (Reeves, 2003). A recombinação é realizada fazendo uma analogia com a genética, utilizando operadores de cruzamento e mutação, e a busca é guiada pela avaliação da função objetivo de cada indivíduo da população. Com base nessa sistemática, os indivíduos melhor avaliados, com melhor *fitness*, podem ser identificados e, assim, terem mais chances de se reproduzir. O funcionamento de um Algoritmo Genético está representado por um fluxograma na Figura 3.3.

O processo do Algoritmo Genético apresentado na Figura 3.3 consiste na geração da população inicial e avaliação da aptidão dos seus indivíduos. O Critério de Parada irá garantir que o algoritmo pare de executar, dada alguma regra. Depois da avaliação da aptidão dos indivíduos da população atual, é realizada a seleção de indivíduos para o cruzamento. Esses indivíduos serão cruzados e mutados, caso atenderem aos critérios de cruzamento e mutação e de uma forma definida *a priori*. O novo indivíduo gerado é avaliado e a nova população que está sendo criada é atualizada. Ainda assim, é necessário garantir que a nova população esteja completa antes de voltar a avaliar o critério de parada do algoritmo.

Um AG irá compor a arquitetura do modelo a ser gerado através desse estudo, sendo responsável por um dos três módulos aplicados à resolução do problema (seção 4.3.4). Com isso, seus componentes serão apresentados a seguir, bem como possíveis apontamentos da forma de utilização no modelo a ser desenvolvido.

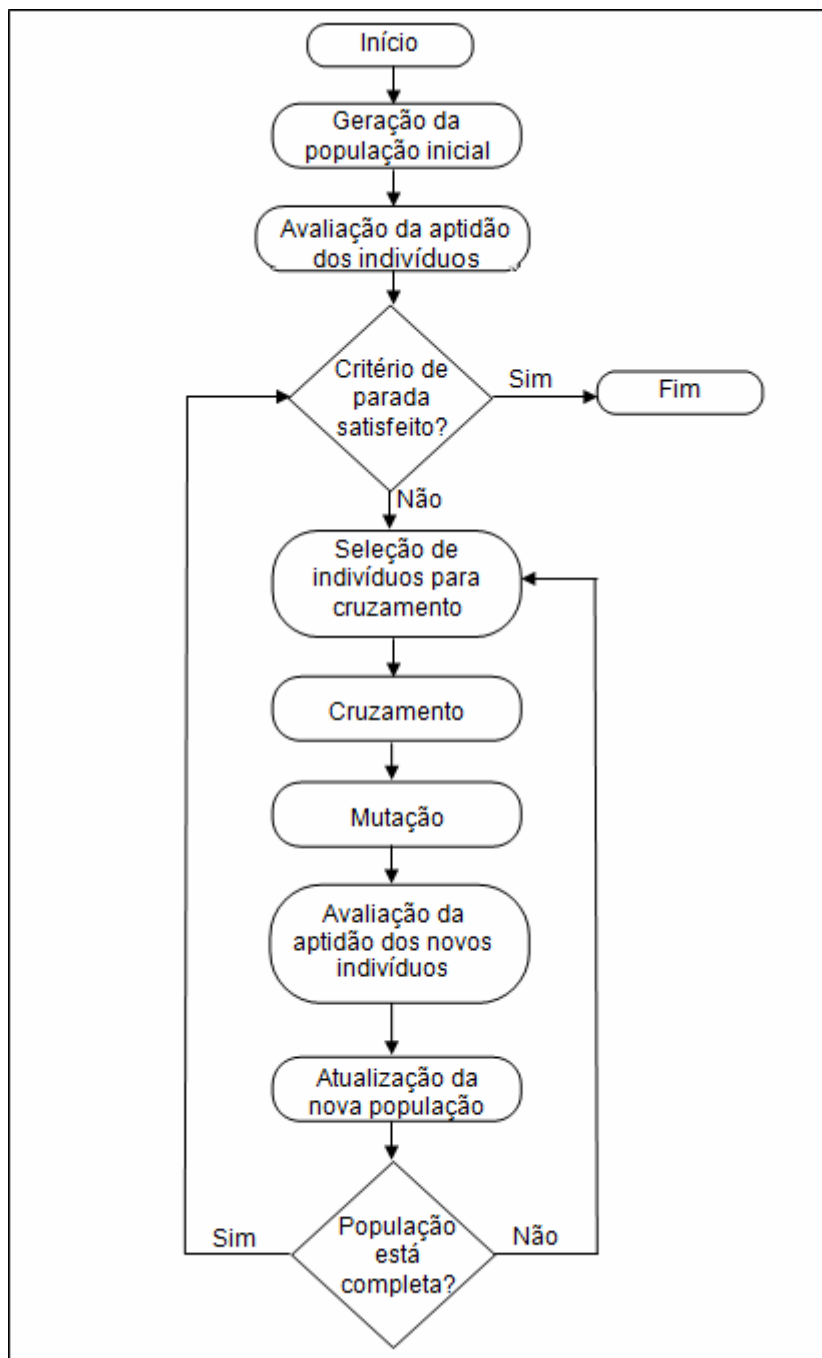


Figura 3.3. Dinâmica do processo do Algoritmo Genético

3.3.3.1 Codificação

Os genes podem ser representados de diversas formas. Reeves (2003) e Hu e Di Paolo (2007) apresentam diferentes possibilidades de representação. Uma das possíveis representações pode ser binária, onde os cromossomos podem ser

compostos de 0s e 1s. Em outros casos, podem ser necessários valores não binários, inteiros ou reais. Nesse caso, pode-se, por exemplo, transformar, *a priori*, os números em binários para após aplicar os operadores. Ainda assim, outra possibilidade de representação consiste na utilização de k-vetores para representar os cromossomos. Por fim, Reeves (2003) e Hu e Di Paolo (2007) abordam o fato de existirem problemas em que a escolha óbvia de representação seria uma permutação e não um conjunto. Esse é o caso do PRV, assim como uma série de outros problemas. Nesses casos, podem ser gerados problemas com alguns operadores de cruzamento. Esse tipo de codificação será aplicado no módulo do Algoritmo Genético, apresentado na seção 4.3.4.

3.3.3.2 População Inicial

O Algoritmo genético parte da geração da população inicial, onde, normalmente, assume-se ser gerada de modo aleatório. Apesar disso, a geração de uma população aleatória não garante necessariamente uma cobertura uniforme do espaço de busca, provendo variabilidade genética. Para tal, segundo Reeves (2003), podem ser utilizadas técnicas mais sofisticadas como, por exemplo, o *Latin Hypercube*. Ainda assim, soluções conhecidas de boa qualidade podem ser inseridas na população inicial, possibilitando que o Algoritmo Genético encontre melhores soluções mais rapidamente do que simplesmente utilizando uma geração aleatória.

3.3.3.3 Critério de Parada

O critério de parada é necessário para definir quando o algoritmo deve encerrar, sendo geralmente utilizados um limite no número de avaliações do *fitness* da população, tempo computacional ou analisar a trajetória da diversidade da população, parando quando esse índice atingir determinado nível (Reeves, 2003; Dridi et al, 2010).

3.3.3.4 Condição de Cruzamento e Mutação

A condição de cruzamento geralmente é implementada através de uma regra de aleatoriedade (Reeves, 2003). Define-se um valor, também conhecido como taxa de cruzamento, e sorteia-se um número de modo aleatório que irá definir se o cruzamento será ou não realizado. Não obstante, além do operador de cruzamento, pode ser utilizado um operador de mutação, o qual também necessita de um parâmetro para definir a taxa de mutação. Davis (1991) e Dridi et al (2010) sugerem que, inicialmente, as taxas de cruzamento sejam altas e as taxas de mutação baixas, porém, quando a população começar a convergir, diminuindo a diversidade da população, a taxa de mutação seja aumentada, atualizada em tempo de execução.

3.3.3.5 Métodos de Seleção

Os métodos de seleção definem quais indivíduos serão utilizados no cruzamento. Dentre todos os métodos de seleção, os mais utilizados são a roleta, a classificação e o torneio (Reeves, 2003; Dridi et al, 2010). Esses três métodos serão mostrados.

A ideia básica do método de seleção por roleta consiste em dividir uma roleta, de valores entre 0 e 1, pelo número total de indivíduos e o tamanho ocupado na roleta é definido pelo seu *fitness* em relação aos demais. Com isso, o indivíduo é escolhido através do sorteio de um número aleatório. Nesse caso, os indivíduos mais aptos tendem a ser utilizados com mais frequência. Um exemplo desse método pode ser analisado na Figura 3.4, a seguir.

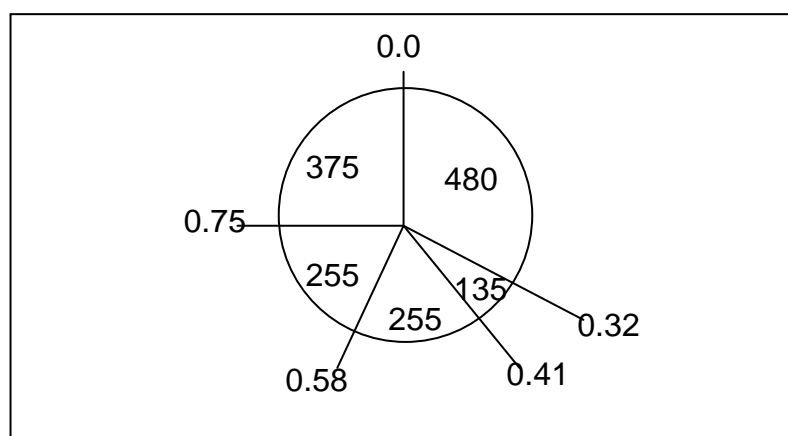


Figura 3.4. Representação do método de seleção por roleta

A Figura 3.4, mostra a representação gráfica de uma roleta de seleção para cinco indivíduos com *fitness* {480, 135, 255, 255, 375}, respectivamente. Neste caso, o somatório total dos *fitness* resulta 1500. Uma vez que a probabilidade de seleção do indivíduo está diretamente ligada ao seu *fitness*, as probabilidades de seleção desses cinco indivíduos são de 32%, 9%, 17%, 17% e 25%, respectivamente. Essas probabilidades são dispostas na roleta, como pode ser visto, onde o primeiro indivíduo será selecionado se um número sorteado for maior ou igual a 0 e menor do que 0,32, o segundo indivíduo será selecionado se um número sorteado for maior ou igual a 0,32 e menor do que 0,41, e assim sucessivamente para os demais indivíduos.

O método de classificação consiste em ordenar os indivíduos de uma população, possibilitando a escolha dos mesmos da forma que for mais conveniente. Para tal, supõe-se que a probabilidade de seleção de um indivíduo classificado na posição k é dada por $P[k]$, conforme equação (3.10).

$$P[k] = \alpha + \beta k \quad (3.10)$$

Na equação (3.10), α e β são pesos. A premissa de que $P[k]$ é uma função distribuição de probabilidade, possibilita uma condição apresentada na equação (3.11), a qual deixa a livre escolha de outro parâmetro de modo a ajustar a classificação.

$$\sum_{k=1}^M (\alpha + \beta k) = 1 \quad (3.11)$$

Por fim, o método de seleção por torneio consiste em selecionar e comparar um conjunto τ de cromossomos onde o mais apto, cromossomo com melhor *fitness*, é selecionado. Uma vantagem desse método de seleção é que ele necessita classificar um par ou um conjunto de cromossomos e ainda pode atuar em situações onde não existe uma função objetivo formal. Contudo, é importante destacar que essa técnica não garante que todos os indivíduos serão selecionados para o torneio.

3.3.3.6 Cruzamento

A técnica de cruzamento é um dos operadores que permite que as soluções para o problema evoluam. Segundo Reeves (2003), o cruzamento consiste, simplesmente, em substituir uma parte dos genes de um pai, pela parte correspondente de outro pai. Por exemplo, para um caso com dois indivíduos a e b, cada um consistindo de 6 variáveis, representando duas possíveis soluções para o problema, Figura 3.5.

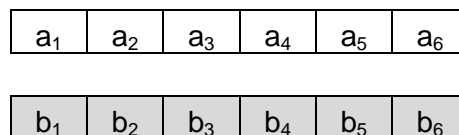


Figura 3.5. Representação dos cromossomos pais

Dois pontos de corte são escolhidos aleatoriamente. No caso de terem sido selecionados os pontos de corte 2 e 4, a prole gerada ficará como na Figura 3.6.

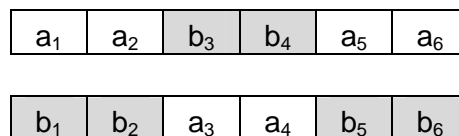


Figura 3.6. Indivíduos gerados

No caso de um cruzamento de cromossomos pais, P1 e P2, compostos por genes binários com um ponto de corte definido em 3, pode-se obter dois filhos, F1 e F2, como é mostrado na Figura 3.7.

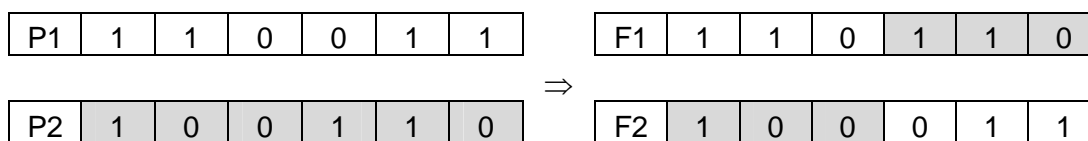


Figura 3.7. Cruzamento com um ponto de corte

Contudo, um problema muito comum pode acontecer se a representação dos cromossomos não for binária, como no caso de o espaço de soluções baseado em um espaço de permutações (Π_1) de número $1, \dots, n$, onde n é o tamanho máximo do cromossomo. Esse é um exemplo de representação para os problemas de sequenciamento e PRV, representado na Figura 3.8.

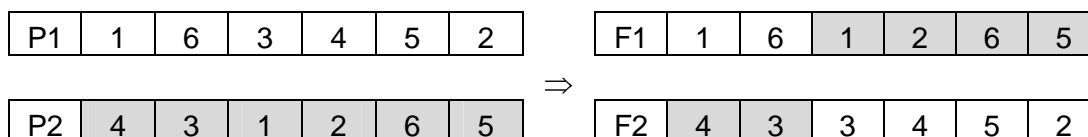


Figura 3.8. Cruzamento de cromossomos com representação não binária

Se no caso da Figura 3.8 com os pais P1 e P2 e um ponto de corte, na segunda posição, fosse a representação do PCV, o indivíduo F1 visitaria as cidades 1 e 6 duas vezes e nunca chegaria às cidades 3 e 4. Assim como, o indivíduo F2, visitaria as cidades 3 e 4 duas vezes e nunca visitaria as cidades 1 e 6. Esse tipo de cruzamento tende a gerar uma grande quantidade de soluções inviáveis (Dridi et al, 2010).

Geiger (2008) comenta que estudos apontam que os melhores operadores de cruzamento para a representação genética através de uma permutação são as técnicas conhecidas como *partially mapped crossover* (PMX) (Goldberg e Lingle, 1985), *order based crossover* (OBX) (Syswerda, 1991) e *uniform order based crossover* (UOBX) (Davis, 1991). Esses três operadores de cruzamento, PMX, OBX e UOBX, são detalhados abaixo, pois são aplicados no módulo do Algoritmo Genético, seção 4.3.4

O PMX foi uma das primeiras ideias para resolver esse problema de inviabilidade de soluções. Nessa técnica, dois pontos de cruzamentos, X e Y, são escolhidos uniformemente de modo aleatório entre 1 e n , onde n é o tamanho

máximo do cromossomo. A área entre esses dois pontos define o mapeamento de trocas. A Figura 3.9, representa esse tipo de operador de cruzamento.

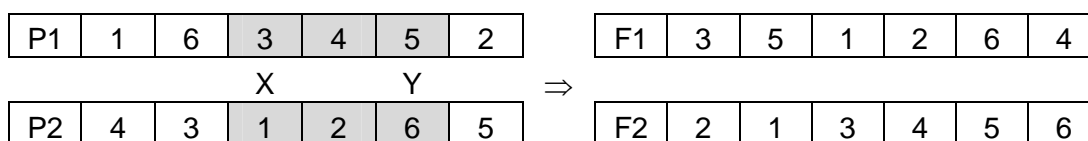


Figura 3.9. Cruzamento utilizando PMX

Os pontos de cruzamento X e Y definem o mapeamento de trocas 3 por 1, 1 por 3, 4 por 2, 2 por 4, 5 por 6 e 6 por 5 em seus respectivos indivíduos, P1 e P2, gerando dois indivíduos novos F1 e F2. Maiores informações sobre essa técnica podem ser encontradas em Goldberg e Lingle (1985).

Outra técnica apontada por Geiger (2008), OBX, tem seu funcionamento diferente da primeira. Considerando dois pais, P1 e P2, conforme Figura 3,10.

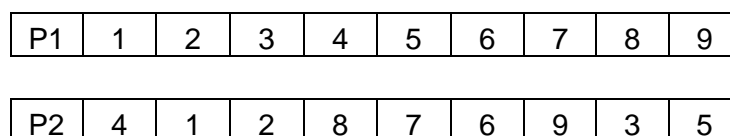


Figura 3.10. Representação de cromossomos para o operador de cruzamento OBX

Inicia-se pela seleção de algumas posições aleatórias, por exemplo, 3, 4, 6 e 9. A ordem das cidades nessas posições do pai P2 será imposta no pai P1. As cidades nessas posições no pai P2, realçadas na Figura 3.11, são 2, 8, 6 e 5.

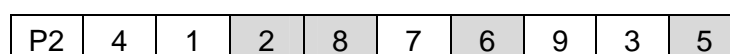


Figura 3.11. Seleção das posições para o operador de cruzamento OBX

No pai P1, as cidades 2, 8, 6 e 5 são representadas pelas posições 2, 8, 6 e 5, Figura 3.12.

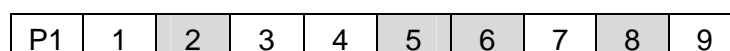


Figura 3.12. Seleção de novos genes para o operador de cruzamento OBX

No indivíduo F1, os elementos dessas posições serão reordenados para combinar a ordem dos mesmos elementos do pai P2, a ordem é 2, 8, 6 e 5. O

indivíduo F1 é uma cópia do pai P1 em todas as posições com exceção das posições 2, 5, 6 e 8, Figura 3.13.

F1	1	x	3	4	x	x	7	x	9
----	---	---	---	---	---	---	---	---	---

Figura 3.13. Formação do primeiro indivíduo resultante por OBX

Todos os outros elementos são preenchidos na ordem dada pelo pai P2, 2, 8, 6 e 5, Figura 3.14.

F1	1	2	3	4	8	6	7	5	9
----	---	---	---	---	---	---	---	---	---

Figura 3.14. Representação de um cromossomo gerado por OBX

Realizando o mesmo processo, mas agora em consideração ao pai P2, para gerar um segundo indivíduo F2. Para as posições sorteadas 3, 4, 6 e 9 são extraídos os número 3, 4, 6 e 9 do pai P1, Figura 3.15.

P1	1	2	3	4	5	6	7	8	9
----	---	---	---	---	---	---	---	---	---

Figura 3.15. Seleção das posições para o operador de cruzamento OBX

Esses números, no pai P2, representam as posições 8, 1, 6 e 7, Figura 3.16.

P2	4	1	2	8	7	6	9	3	5
----	---	---	---	---	---	---	---	---	---

Figura 3.16. Seleção de novos genes para o operador de cruzamento OBX

Copia-se os elementos do pai P2 com exceção das posições 1, 6, 7 e 8, Figura 3.17.

F2	x	1	2	8	7	x	x	x	5
----	---	---	---	---	---	---	---	---	---

Figura 3.17. Formação do primeiro indivíduo resultante por OBX

Os outros elementos são preenchidos com as cidades obtidas do pai P1, 3, 4, 6 e 9, em ordem, Figura 3.18.

F2	3	1	2	8	7	4	6	9	5
----	---	---	---	---	---	---	---	---	---

Figura 3.18. Representação de um cromossomo gerado por OBX

A última técnica apontada por Geiger (2008), UOBX, consistem em, após a seleção dos pais, P1 e P2, criar uma máscara binária M, gerada aleatoriamente, do tamanho do cromossomo, para um novo indivíduo, Figura 3.19.

P1	1	2	3	4	5	6	7	8	9
P2	4	1	2	8	7	6	9	3	5
M	0	0	1	1	0	1	1	0	1

Figura 3.19. Representação de cromossomos e máscaras para o operador de cruzamento UOBX

Após a criação da máscara, copia-se do pai P1 as posições que possuem, na máscara, valor um, posições 3, 4, 6, 7 e 9, Figura 3.20.

F1	x	x	3	4	x	6	7	x	9
----	---	---	---	---	---	---	---	---	---

Figura 3.20. Formação do primeiro indivíduo resultante por UOBX

Preenche-se as posições restantes com os valor obtidos do pai P2 em ordem, removendo os valores selecionados do pai P1, 1, 2, 8 e 5, Figura 3.21.

F1	1	2	3	4	8	6	7	5	9
----	---	---	---	---	---	---	---	---	---

Figura 3.21. Representação de um cromossomo gerado por UOBX

A mesma regra deve ser aplicada para um novo indivíduo F2. Copia-se do pai P2 as posições da mascara M que possuem valor um, 2, 8, 6, 9 e 5, Figura 3.22.

F2	x	x	2	8	x	6	9	x	5
----	---	---	---	---	---	---	---	---	---

Figura 3.22. Formação de um indivíduo resultante por UOBX

Preenche-se as posições restantes com os valores obtidos do pai P1 em ordem, removendo os valores selecionados do pai P2, 1, 3, 4 e 7, Figura 3.23.

F2	1	3	2	8	4	6	9	7	5
----	---	---	---	---	---	---	---	---	---

Figura 3.23. Representação de um cromossomo gerado por OBX

3.3.3.7 Mutação

O conceito do operador de mutação é mais simples do que o conceito dos operadores de cruzamento. A mutação pode ser aplicada usando uma distribuição de Bernoulli associando-se um valor p muito baixo para cada gene, para definir se irá ocorrer mutação ou não. Essa ideia implica em um problema de desempenho quando o cromossomo for muito grande, devido a isso, em 1964 (Bremermann et al, 1964) surgiu uma nova forma da aplicação da mutação. Utilizando uma distribuição de Poisson com parâmetro λ , onde λ é o número médio de mutações por cromossomo e geralmente com valor 1. Com isso, se $\lambda = 1$ e n é o número total de genes de um cromossomo, tem-se que a taxa de mutação é definida por $\mu = 1/n$.

No caso de os genes serem binários, pode ser utilizada uma simples troca de valor, de 0 para 1 ou de 1 para 0, como na Figura 3.24, a seguir.

I1	0	0	1	1	0	1
I1 _m	0	0	1	0	0	1

Figura 3.24. Representação de mutação em genes binários

A Figura 3.24, acima, mostra o indivíduo I1 sofrendo uma mutação no quarto gene, tendo o valor de seu alelo alterado de 1 para 0, resultando no indivíduo I1_m.

Apesar de ser uma troca simples, quando existem diversas possibilidades de valores para os alelos dos genes, pode-se decidir o novo valor de modo aleatório. Ainda assim, em alguns casos, os alelos são sensíveis às variações desses valores, como no caso do PRV. Desse modo, Geiger (2008) sugere a utilização de técnica de mutação chamada de *swap mutation* que consiste na troca de valores entre dois genes selecionados aleatoriamente, como na Figura 3.25.

I1	1	4	2	5	3	6
I1 _m	5	4	2	1	3	6

Figura 3.25. Representação de mutação utilizando *swap mutation*

A Figura 3.25, acima, mostra um exemplo onde as posições 1 e 4 foram selecionadas no indivíduo I1. Aplicando a técnica de *swap mutation*, alteram-se os valores das posições entre si, gerando um novo indivíduo I1_m. Esse novo indivíduo não terá problema de viabilidade de soluções, se esses cromossomos fossem possíveis soluções para o PRV (Geiger, 2008). Essa técnica, *swap mutation*, é aplicada no módulo do Algoritmo Genético, seção 4.3.4.

3.3.3.8 Nova População

O AG original de Holland, segundo Reeves (2003), assumia que os passos de seleção, recombinação e mutação eram aplicados em uma população de M indivíduos até que um conjunto de novos M indivíduos fosse gerado, formando uma nova população. Do ponto de vista da otimização combinatória, isso pode parecer estranho, pois existe a possibilidade de combinações inapropriadas ocorrerem, descartando boas combinações obtidas até o momento. Por essa razão, De Jong (1975) introduziu o conceito de elitismo e sobreposição de população. A ideia do elitismo consistia em manter vivo o melhor indivíduo da população atual, e gerar os outros (M - 1) indivíduos, enquanto que sobreposição tinha por objetivo substituir uma fração da população por novos indivíduos.

Geiger (2008) ressalta que é importante que seja mantida a diversidade da população o maior tempo possível e que uma forma muito utilizada chama-se política de não duplicidades (Davis, 1991). Essa política consiste em não permitir que clones sejam inseridos na nova população. Isso acabou gerando um problema de desempenho, uma vez que os novos indivíduos precisavam ser comparados. Devido a isso, Davis (1991) sugeriu que antes da aplicação do cruzamento, os pais fossem avaliados para identificar possíveis pontos de corte.

4 O Modelo

O modelo desenvolvido é baseado na versão clássica do PRV com restrições de capacidade e janelas de tempo, PRVCJT. Nesta versão do problema assume-se que todos os veículos têm capacidade homogênea e estão localizados em um único depósito. Existe uma demanda e uma janela de tempo conhecidas para cada cliente. Todas as rotas devem iniciar e terminar no depósito. A demanda total de uma rota não pode exceder a capacidade do veículo e o veículo não pode chegar ao cliente depois do fim da janela de tempo. O objetivo é criar um conjunto de rotas que atenda a todos os clientes uma única vez, minimizando os custos destes atendimentos. Uma vez que o PRVCJT é NP-difícil optou-se por desenvolver o modelo utilizando as metaheurísticas Busca Tabu e Algoritmos Genéticos além de um algoritmo híbrido envolvendo as duas técnicas.

Todos os experimentos computacionais foram realizados em um computador com oito núcleos de processamento, com 3.0 Ghz cada, e 24 GBs de memória randômica. Na Busca Tabu, toda geração de vizinhança e avaliação das soluções são realizadas em paralelo, utilizando os oito núcleos para tal. No Algoritmo Genético, um núcleo de processamento seleciona os indivíduos para cruzamento ou mutação enquanto que os demais núcleos realizam as tarefas em paralelo. Essas mesmas características são utilizadas no Algoritmo Híbrido.

4.1 Definição do Modelo

Partindo dos conceitos básicos sobre o PRV, apresentados na seção 2.2, o PRVCJT é definido por uma rede $R = (V, A, M)$ composta por um conjunto $V = \{v_i \mid i = 1, \dots, n\}$ de n clientes, um conjunto de arestas A ligando um cliente i a um cliente j e um conjunto de magnitudes M associado ao conjunto de arestas A . O depósito é representado por dois vértices, v_0 e v_{n+1} , de modo que todas as rotas viáveis correspondem a caminhos em R iniciando no vértice v_0 e terminando no vértice v_{n+1} .

Uma janela de tempo também é associada aos vértices v_0 e v_{n+1} . Por exemplo, $[a_0, b_0] = [a_{n+1}, b_{n+1}] = [E, L]$, onde E e L representa a primeira possível partida do depósito e a última possível chegada no depósito, respectivamente. Uma demanda e tempos de serviços nulos são definidos para esses dois vértices, ou seja, $d_0 = d_{n+1} = s_0 = s_{n+1} = 0$. Ainda assim, um conjunto $K = \{k_l \mid l = 1, \dots, m\}$ de m veículos é considerado, onde cada veículo k está associada a uma capacidade máxima Q_k . Soluções viáveis existem somente se atenderem às restrições (4.1) e (4.2), definidas a seguir.

$$d_0 = E \leq \min_{i \in V} b_i - t_{0i} \quad (4.1)$$

$$b_{n+1} = L \geq \min_{i \in V} a_i + s_i + t_{i0} \quad (4.2)$$

Por fim, é importante salientar que uma aresta $(i, j) \in A$ pode ser eliminado se uma restrição de tempo, equação (4.3), ou de capacidade, equação (4.4), for violada.

$$a_i + s_i + t_{ij} > b_j \quad (4.3)$$

$$d_i + d_j > C \quad (4.4)$$

A seguir, é apresentada uma formulação de programação matemática para o PRVCJT envolvendo dois tipos de variáveis: variáveis de fluxo x_{ijk} , $(i, j) \in A$, $k \in K$, igual a 1 se a aresta (i, j) é utilizado pelo veículo k e 0 caso contrário, e variáveis de tempo w_{ik} , $i \in V$, $k \in K$, especificando o início do serviço no nodo i quando servido pelo veículo k . Não existem limitantes no número de veículos.

4.2 Formulação Matemática

O PRVCJT pode ser formalmente descrito seguindo o modelo de fluxo de rede de multiprodutos com restrições de janelas de tempo e capacidade apresentado por (Cordeau et al, 2003)

$$\sum_{(i,j) \in A} \sum_{k \in K} c_{ij} x_{ijk} \quad (4.5)$$

sujeito a:

$$\sum_{i \in K} \sum_{j \in C^+} x_{ij} = 1, \quad \forall i \in C^+ \quad (4.6)$$

$$\sum_{i \in C^+} \sum_{j \in C^+} x_{ij} = 1, \quad \forall k \in C^+ \quad (4.7)$$

$$\sum_{i \in C^+} \sum_{j \in C^+} x_{ij} - \sum_{i \in C^+} \sum_{j \in C^+} x_{ji} = 0, \quad \forall k \in C^+ \quad (4.7)$$

$$\sum_{i \in C^+} \sum_{j \in C^+} x_{ij} = 1, \quad \forall k \in C^+ \quad (4.8)$$

$$x_{ij} (w_{ijk} + s_i + t_{ij} - w_{jlk}) \leq 0, \quad \forall k \in C^+, (i,j) \in A$$

$$a_i \sum_{j \in C^+} x_{ij} \leq s_{ik} \leq b_i \sum_{j \in C^+} x_{ij}, \quad \forall k \in C^+, i \in C^+ \quad (4.9)$$

$$E \leq w_{ijk} \leq L, \quad \forall k \in C^+, i \in \{0, n+1\}$$

$$\sum_{i \in C^+} \sum_{j \in C^+} x_{ij} \leq Q_k, \quad \forall k \in C^+ \quad (4.10)$$

$$x_{ij} \geq 0, \quad \forall k \in C^+, (i,j) \in A \quad (4.11)$$

$$x_{ij} \in \{0, 1\}, \quad \forall k \in C^+, (i,j) \in A \quad (4.12)$$

A função objetivo (4.5) expressa o custo total. A restrição (4.6) restringe os clientes a serem atendidos por exatamente um veículo, onde C^+ representa o conjunto de arestas que podem ser percorridas a partir do vértice i . As restrições (4.7)-(4.10) caracterizam o fluxo no caminho a ser seguido pelo veículo k e C^+ representa o conjunto de arestas de chegada no vértice i . Ainda, as restrições (4.11), (4.12) e (4.13) garantem a viabilidade do sequenciamento com respeito ao tempo e à capacidade. Observa-se que para um dado veículo k , a restrição (4.12) força $w_{ik} = 0$ sempre que o cliente i não é visitado pelo veículo k . Finalmente, a restrição (4.15) impõe uma condição binária para as variáveis de fluxo.

A condição binária (4.15) permite que a condição (4.11) seja linearizada, conforme equação (4.16).

$$w_{ijk} + s_i + t_{ij} - w_{jlk} \leq (1 - x_{ij}) M_{ij}, \quad \forall k \in C^+, (i,j) \in A \quad (4.16)$$

onde M_{ij} são constantes enormes. Além disso, M_{ij} pode ser substituída por $\max\{b_i + s_i + t_{ij} - a_j, 0\}$, $(i, j) \in A$, e as restrições (4.11) e (4.16) precisam simplesmente ser trocadas para arcos $(i, j) \in A$ tal que $M_{ij} > 0$; caso contrário, quando $\max\{b_i + s_i + t_{ij} - a_j, 0\} = 0$, essas restrições são satisfeitas para todos os valores de w_{ik} , w_{jk} e x_{ijk} .

4.3 Arquitetura do modelo

Para implementar tal modelo, foi definida uma arquitetura composta por cinco módulos: módulo de geração das redes, módulo de geração da solução inicial, módulo da Busca Tabu, módulo do Algoritmo Genético e módulo do Algoritmo Híbrido. A representação gráfica desta arquitetura pode ser vista na Figura 4.1.

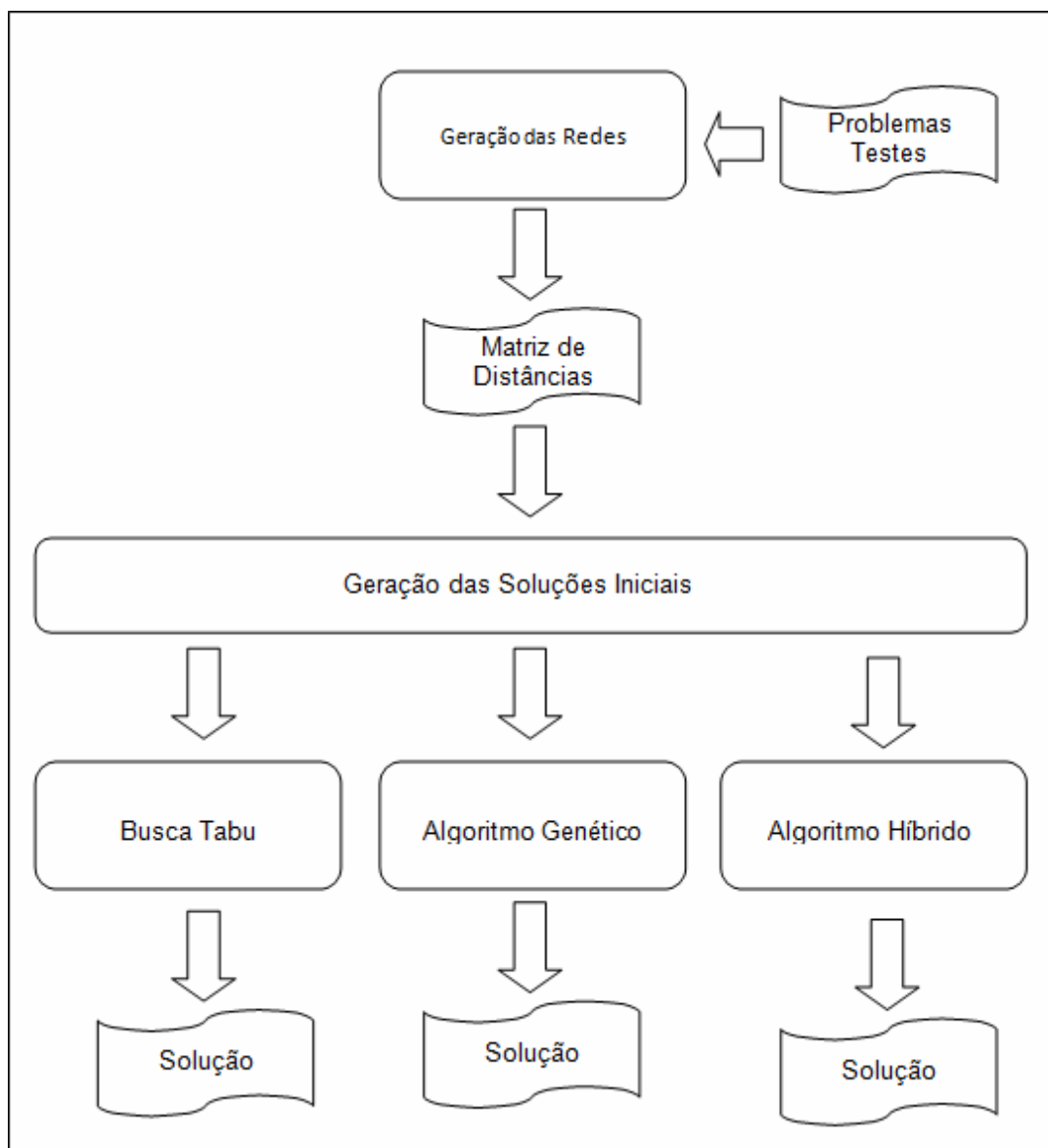


Figura 4.1. Arquitetura do modelo

O módulo de Geração das Redes, através dos dados obtidos a partir de problemas testes seleccionados (vértices, coordenadas e demandas), gera uma matriz de distância entre todos os vértices da rede. A matriz de distância é utilizada no módulo de Geração das Soluções Iniciais para gerar as soluções iniciais utilizadas no módulo da Busca Tabu, Algoritmo Genético e Algoritmo Híbrido. No módulo da Busca Tabu, a partir de uma solução inicial viável, espaço de busca é percorrido através da geração de soluções vizinhas e escolha da melhor solução de cada vizinhança. No módulo do Algoritmo Genético, a partir da geração da população inicial pelo módulo de Geração das Soluções Iniciais, são utilizados operadores de cruzamento e mutação para explorar o espaço de busca. No módulo

do Algoritmo Híbrido o objetivo será combinar as grandes vantagens que cada metaheurística possui, intensidade e variabilidade.

4.3.1 Geração das Redes

Este módulo tem o objetivo de gerar uma matriz com as distâncias euclidianas entre os vértices. Com base em instâncias clássicas do PRVCJT, encontradas na literatura, obtêm-se as coordenadas (x_1, y_1) e (x_2, y_2) de modo que através da fórmula $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ (Cook et al, 1998) as distâncias possam ser calculadas, gerando uma matriz de distâncias simétrica, representada na Figura 4.2, a seguir.

	1	2	3	4	5	6	...
1		d12	d13	d14	d15	d16	
2	d21		d23	d24	d25	d26	
3	d31	d32		d34	d35	d36	
4	d41	d42	d43		d45	d46	
5	d51	d52	d53	d54		d56	
6	d61	d62	d63	d64	d65		
...							...

Figura 4.2. Representação gráfica de uma matriz de distâncias

4.3.2 Geração das Soluções Iniciais

Esse módulo utilizará a matriz de distâncias, gerada pelo módulo de Geração das Redes, para gerar a solução inicial para o módulo da Busca Tabu e a população Inicial para o módulo do Algoritmo Genético e do Algoritmo Híbrido.

4.3.2.1 Solução Inicial para o Módulo da Busca Tabu

A solução inicial para a Busca Tabu será obtida, através da heurística do Vizinho mais Próximo. Nesta heurística, o próximo vértice a ser inserido na rota é determinado pelo último que foi inserido, uma vez que, segundo esta heurística, o próximo vértice selecionado é sempre o mais próximo do último vértice inserido. Como as rotas devem iniciar e terminar no depósito, naturalmente, a primeira aresta escolhida será uma aresta que liga (v_0, v_i) em quem v_0 representa o depósito e v_i outro vértice qualquer em que o valor desta aresta seja mínimo. Dado que a capacidade do veículo e o tempo são limitantes, toda vez que a inserção de um próximo cliente à rota violar a restrição da capacidade ou da janela de tempo, este cliente não é inserido e retorna-se ao depósito. Se ainda houver clientes não atendidos, inicia-se a partir do depósito, a formação de uma nova rota, desprezando-se os vértice e arestas já utilizados. A solução inicial gerada neste módulo é representada por uma rede que interliga todos os vértices representando os clientes através de rotas que iniciam e terminam no vértice que representa o depósito. A Figura 4.3, apresenta essa dinâmica.

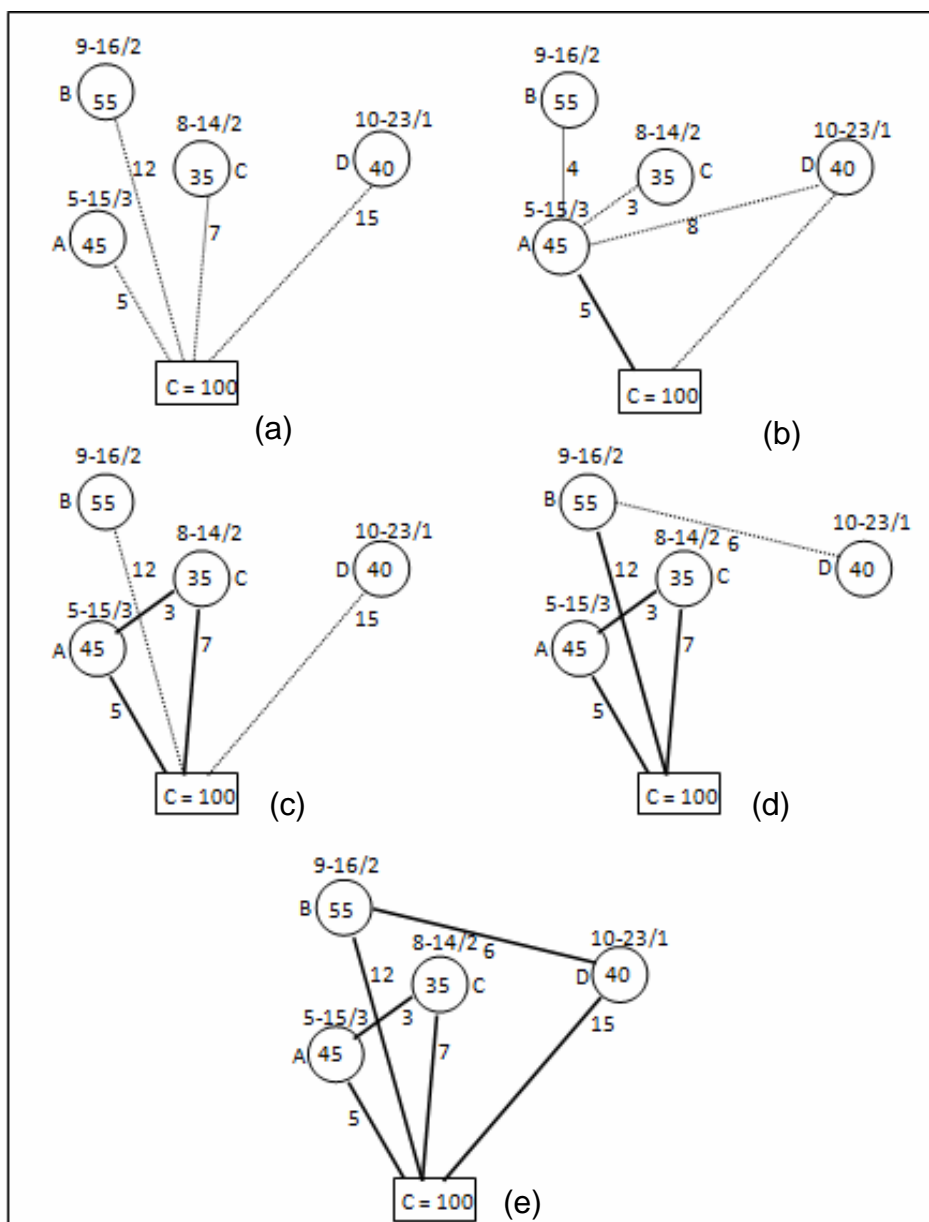


Figura 4.3. Representação da obtenção da solução inicial da Busca Tabu

A Figura 4.3, mostra um exemplo do processo que irá ocorrer para gerar as soluções iniciais para o módulo da Busca Tabu. Existe um depósito com uma capacidade Q igual a 100 e 4 cidades (a, b, c e d). Essas cidades possuem uma demanda associada, $a = 45$, $b = 35$, $c = 55$ e $d = 40$, um tempo de serviço, $a = 3$, $b = 2$, $c = 2$ e $d = 1$, além de uma janela de tempo, $a = 5 - 15$, $b = 9 - 16$, $c = 8 - 14$ e $d = 10 - 23$. O algoritmo do vizinho mais próximo irá selecionar sempre a cidade mais perto da posição atual do veículo. Partindo desse ponto, o veículo atende a cidade A por estar mais perto. Essa cidade está a 5 unidades de distância, sua janela de tempo possibilita que o veículo a atenda entre o tempo 5 e o tempo 15 e que o

veículo levará 3 unidades de tempo para atendê-lo. A partir da cidade A, a cidade C é selecionada por ser a cidade mais próxima e não violar nenhuma restrição. Em seguida, a rota retorna ao depósito, pois todas as cidades seguintes irão violar alguma das restrições.

Desse modo, por ainda existir cidades sem rota, o algoritmo é reiniciado, selecionando, inicialmente, a cidade B, por estar mais próxima ao depósito, em seguida a cidade D e por fim retornar ao depósito.

4.3.2.2 População Inicial para os Módulos do Algoritmo Genético e Algoritmo Híbrido

A técnica de geração da população inicial para os módulos do Algoritmo Genético e do Algoritmo Híbrido será a mesma. Ela consiste na geração de soluções aleatórias, através da geração de números aleatórios, uniformemente distribuídos, associados a cada cliente. A cada escolha de um cliente, a distribuição da associação dos números aleatórios aos clientes será recalculada para evitar que o método tenha de sortear números aleatórios sem necessidade. Em um conjunto de 10 clientes, por exemplo, o primeiro cliente seria escolhido se fosse sorteado um número aleatório entre 0 e 0,1. Na segunda iteração, onde algum cliente já haveria sido escolhido, o primeiro cliente na lista seria escolhido se o número sorteado estiver no intervalo entre 0 e 0,1111111, e assim, sucessivamente.

4.3.3 Módulo da Busca Tabu

Tendo em vista o que foi discutido sobre a Busca Tabu, na seção 3.3.2, o módulo da Busca Tabu, a ser desenvolvido, terá um funcionamento parecido com o fluxograma apresentado na Figura 3.2 da seção 3.3.2. A Figura 4.4 tem o objetivo de apresentar o macro processo que será realizado pelo algoritmo, enquanto que as sessões seguintes irão detalhar e esclarecer melhor cada processo.

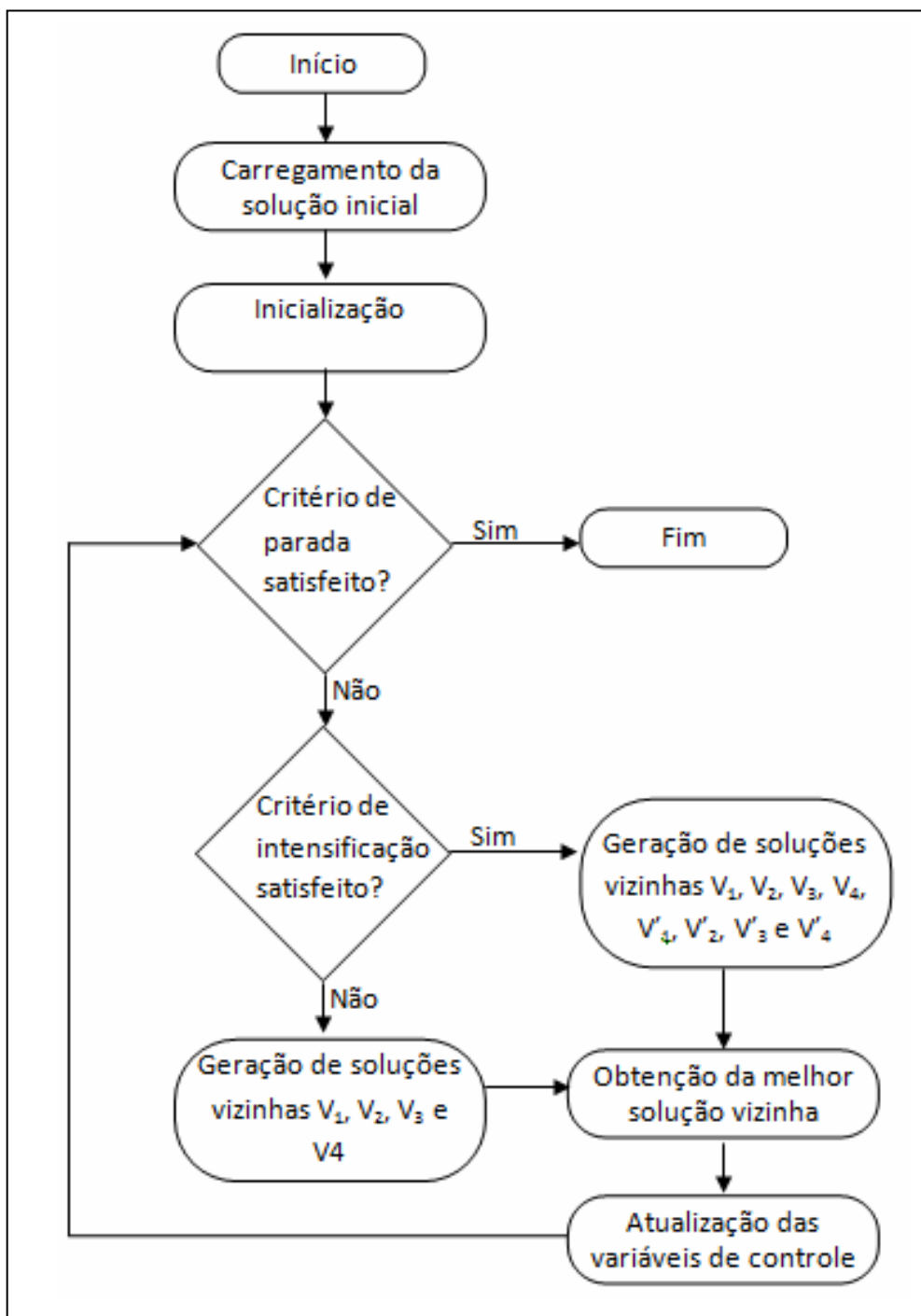


Figura 4.4. Fluxo do processo de Busca Tabu

Na Figura 4.4, pode ser verificada uma mudança em relação ao fluxo básico. O algoritmo iniciará carregando uma solução definida, gerada pelo módulo gerador de soluções iniciais, e as variáveis de controle e funções da Busca Tabu serão inicializadas. As possíveis soluções serão compostas pela sequencia das cidades a serem visitadas, de modo que cada rota será definida pela violação da restrição, ou seja, uma nova rota começa quando uma cidade não consegue visitar outra sem

violam uma das restrições do modelo. O Critério de Parada irá verificar o número máximo de iterações sem melhoras na função objetivo. Em seguida, verifica-se um critério de intensificação, o qual irá verificar o número de iterações sem melhora da função objetivo atende ao critério. Caso não seja atendido, utiliza-se a estrutura de geração de vizinhança padrão para gerar novas soluções, porém, caso o critério seja aceito, utiliza-se uma estrutura de geração de vizinhança intensificada para gerar novas soluções, onde a melhor solução gerada será selecionada desde que atenda às restrições da Lista Tabu e do Critério de Aspiração. Por fim, as variáveis de controle e as funções da Busca Tabu são atualizadas. As sessões a seguir irão detalhar cada um dos elementos expostos na Figura 4.4.

4.3.3.1 Representação da Solução

A representação adotada neste estudo consiste em soluções compostas por N posições. Cada posição representará um cliente do problema e a sequência em que eles se encontram representam a sequência de visitação. Uma possível solução pode ser “3 9 2 6 1 8 7 10 4 5” o qual pode ser particionado em um conjunto de rotas, $r_1 = [v_3, v_9, v_2, v_6]$, $r_2 = [v_1, v_8, v_7, v_{10}]$ e $r_3 = [v_4, v_5]$ assumindo que as rotas serão definidas pela não violação das restrições do modelo.

Opta-se por esse tipo de representação, pois pode ser utilizado exatamente igual para o Algoritmo Genético e também para o Algoritmo Híbrido.

4.3.3.2 Estrutura de Vizinhanças

A solução inicial será gerada no módulo de Geração de Soluções Iniciais e servirá como base para a primeira geração de vizinhanças. A geração de vizinhanças consiste na utilização das heurísticas de melhoramento *Intra-Route* (V_1) e *Inter-Route* (V_2), apresentadas na seção 3.2.5 e 3.2.6, respectivamente, em uma técnica de *swap* (V_3) que consiste na troca de posições de dois clientes pertencentes

a rotas distintas e na heurística do Vizinho Mais Próximo (V_4), apresentada na seção 3.2.7, aplicada dentro de cada rota, individualmente. A heurística do Vizinho Mais Próximo, aplicada na geração da vizinhança tem a função de buscar um arranjo de clientes que busque minimizar o custo na rota que estiver operando.

A Figura 4.5, representa a estrutura de vizinhança V_1 , onde um cliente é removido de sua posição em sua rota atual e é inserido em outra posição da mesma rota. No exemplo da Figura 4.5, as ligações entre o cliente C e B, C e D e E e o depósito são removidas. Em seguida, os clientes B e D são interligados, de modo a deixar o cliente C sem nenhuma ligação. Por fim, o cliente C é inserido entre o cliente E e o depósito.

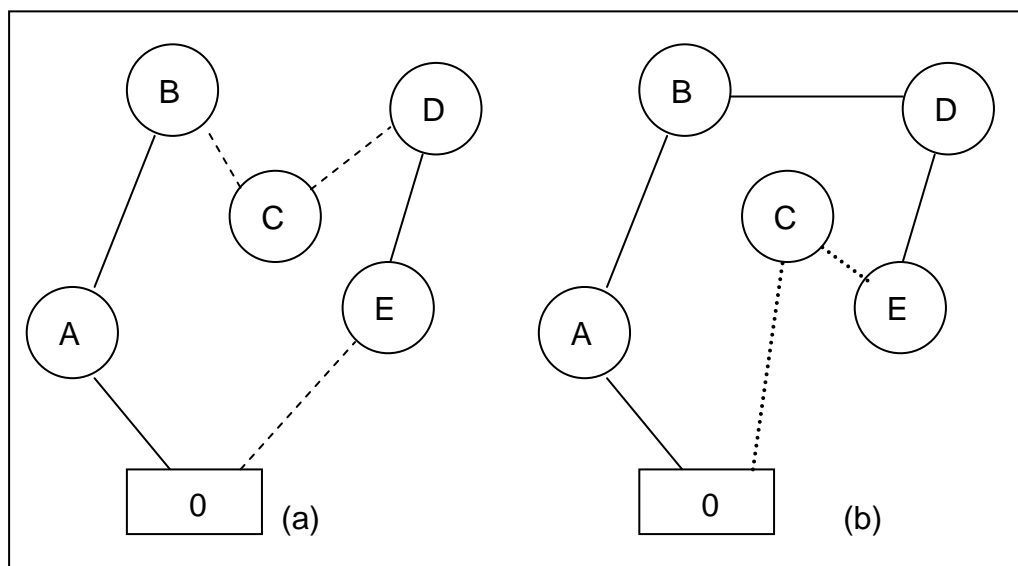


Figura 4.5. Representação da estrutura de vizinhança V_1

A Figura 4.6, representa a estrutura de vizinhança V_2 , onde um cliente é removido de sua rota e inserido em uma nova. Nesse exemplo, as ligações entre B e E e E e D são removidas para que o cliente E possa ser retirado de sua rota e a ligação entre F e G também é removida para que o cliente E possa ser inserido em sua nova rota. Com isso, uma ligação entre B e D é criada para manter a integridade da rota que E pertencia e, também, as ligações entre F e E e E e G são criadas para acomodar o cliente E em sua nova rota.

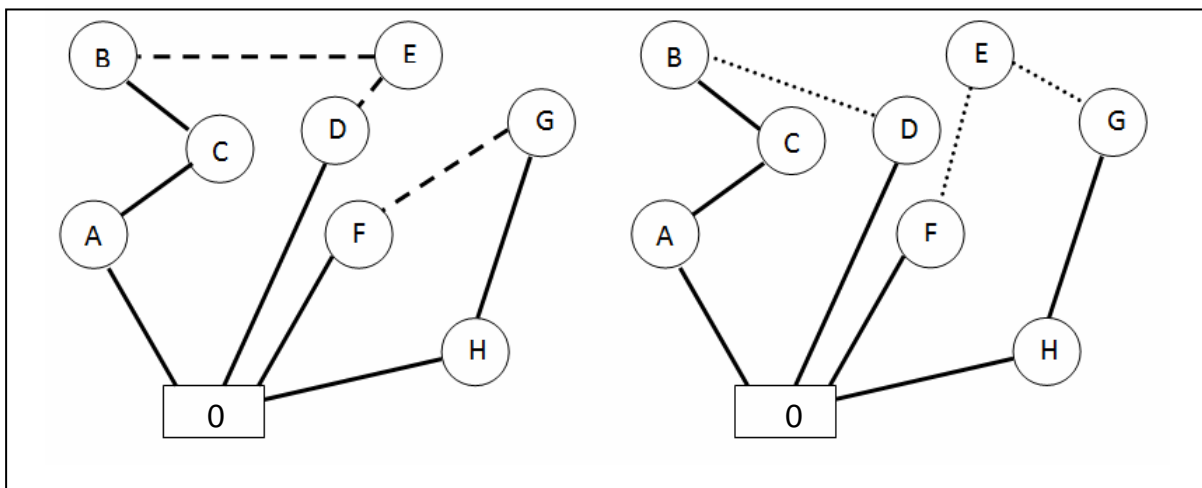


Figura 4.6. Representação da estrutura de vizinhança V_2

A Figura 4.7, representa a estrutura de vizinhança V_3 , onde as posições de dois clientes são trocadas. No exemplo da Figura 4.7, os clientes B e C são trocados. Essa troca consiste na alteração da ordem em que esses clientes são visitados.

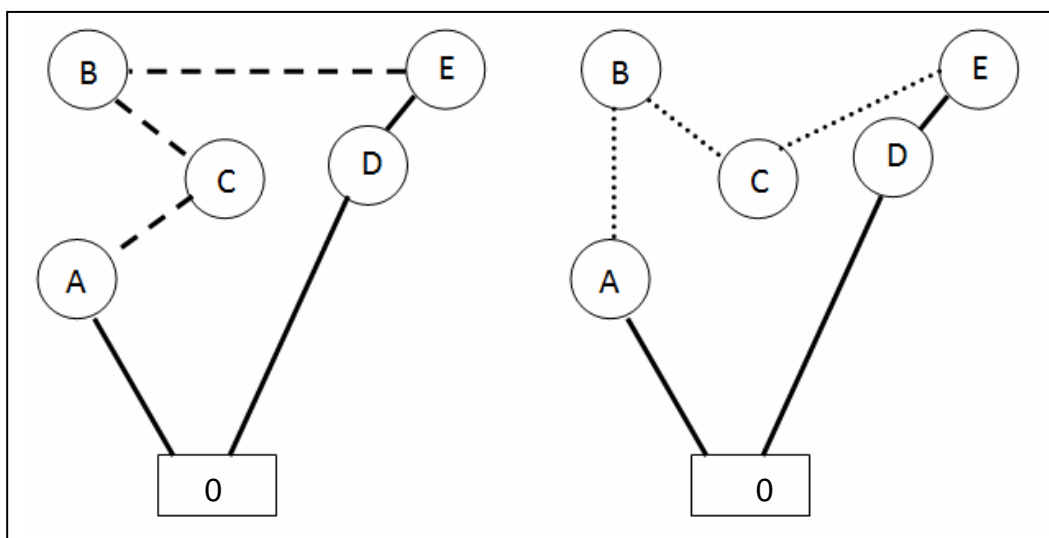


Figura 4.7. Representação da estrutura de vizinhança V_3

A estrutura de vizinhança V_4 está representada na Figura 4.3, da seção 4.3.2.1, e será aplicada em uma rota selecionada *a priori* de modo aleatório.

4.3.3.3 Lista Tabu

A Lista Tabu é uma estrutura definida para guardar os movimentos considerados proibidos. Neste modelo, utiliza-se a representação da Lista Tabu por meio de uma dupla (cliente, rota). Pode-se representar, por exemplo, um cliente v_1 sendo retirado da rota R_1 e sendo alocado na rota R_2 por (v_1, R_1) , impedindo, dessa forma, que o cliente v_1 volte para R_1 enquanto o movimento ficar armazenado na Lista Tabu. O tamanho da Lista Tabu irá variar conforme os testes que forem realizados, capítulo 5. Ainda assim, caso a solução escolhida como a nova solução seja proibida, esta solução ainda poderá ser admitida pelo Critério de Aspiração.

4.3.3.4 Critério de Aspiração

O Critério de Aspiração utilizado consiste em permitir que uma solução Tabu seja aceita apenas se ela melhorar o valor da função objetivo em comparação com a melhor solução encontrada pelo algoritmo até o momento.

4.3.3.5 Intensificação

A estratégia de intensificação tem o objetivo de direcionar a busca em soluções que historicamente produziram bons resultados. Neste estudo, utiliza-se uma Lista de Soluções Elite, armazenando as 10 melhores soluções, que será utilizada após 15 iterações sem melhora da função objetivo. A técnica consiste em intensificar a busca, gerando um número muito maior de vizinhos, a partir de uma determinada solução, armazenada na Lista de Soluções Elite, e tomar a melhor encontrada como solução atual. A estrutura V'_1 baseia-se na estrutura V_1 , apresentada na Figura 4.5, na seção 4.3.3.1, mas será aplicada em todos os clientes da rota selecionada. A estrutura V'_2 baseia-se na estrutura V_2 , apresentada na Figura 4.6, na seção 4.3.3.1, mas será selecionado um cliente de cada rota, esse

cliente será inserido em todas as posições possíveis de todas as rotas. A estrutura V'_3 baseia-se na estrutura de vizinhança V_3 e consiste na estrutura de vizinhança apresentada na Figura 4.7, na seção 4.3.3.1, mas irá realizar todas as possíveis trocas, dois a dois, de uma rota. Por fim, a estrutura V'_4 baseia-se na estrutura de vizinhança V_4 e consiste na aplicação da estrutura de vizinhança apresentada na Figura 4.3, na seção 4.3.2.1, em todas as rotas existentes.

4.3.3.6 Critério de Parada

O Critério de Parada a ser utilizado será um critério clássico, onde será definido um número máximo de iterações sem que ocorra uma melhora da função objetivo (Nbmax). Esse critério irá variar conforme o teste que estiver sendo aplicado (capítulo 5).

4.3.4 Módulo do Algoritmo Genético

No módulo do Algoritmo Genético são aplicadas duas dinâmicas diferentes para a política de elitismo. As duas políticas de elitismo são aplicadas, testadas e comparadas individualmente com o objetivo de verificar qual irá produzir melhores resultados.

4.3.4.1 Codificação

A codificação adotada neste estudo consiste em cromossomos compostos por N genes. Cada gene representará um cliente do problema e a sequência desses genes corresponde à sequência de visitas aos clientes. Um possível indivíduo, por exemplo, pode ser “5 2 1 3 7 9 8 4 6 10”, o qual pode ser particionado em um

conjunto de rotas, $r_1 = [v_5, v_2, v_1]$, $r_2 = [v_3, v_7, v_9, v_8]$, $r_3 = [v_4, v_6, v_{10}]$, assumindo que as rotas serão definidas pela não violação das restrições do modelo.

Opta-se por esse tipo de representação, pois pode ser utilizado exatamente igual para a Busca Tabu e também para o Algoritmo Híbrido.

4.3.4.2 Operadores de Recombinação Genética

Uma vez que a codificação escolhida para representar os cromossomos são permutações, diversos problemas de viabilidade surgem no momento da escolha dos operadores de cruzamento e mutação. Devido a esses problemas, apresentados na seção 3.3.6, são utilizados os três operadores de cruzamento, apresentados na mesma seção, como possíveis soluções esse problema de viabilidade. São eles PMX, OBX e UOBX. Ainda considerando os problemas de viabilidade, utiliza-se um operador de mutação, chamado como *swap mutation*, apresentado na seção 3.3.7.

4.3.4.3 Critério de Parada

Considerando o que é apresentado na seção 3.3.3, sobre os critérios de parada, neste trabalho, será utilizado o critério de número máximo de populações geradas, quando utilizada a política de elitismo DJ, onde o algoritmo irá parar somente quando atingir um determinado número de populações geradas. Quanto utilizada a política de elitismo GE, considera-se um número máximo de indivíduos gerados. Esse valor é obtido através do produto entre o número máximo de indivíduos na população e o número máximo de gerações populacionais.

4.3.4.4 Elitismo

Uma das técnicas de elitismo que será utilizada neste trabalho está baseada no conceito inicial de elitismo proposto por De Jong (1975) (DJ), onde o melhor indivíduo era copiado para a nova população, evitando assim que esse se perdesse. Busca-se utilizar esse conceito com uma variação, onde esse método possa ser ampliado, permitindo a cópia de m melhores indivíduos para a nova população. A Figura 4.8, apresenta uma variação no modelo apresentado na Figura 3.3, na seção 3.3.3, pois, nesse caso, está sendo incorporada a técnica de elitismo. Pode ser analisado que antes do início da seleção de indivíduos para o cruzamento, será realizada uma cópia de um determinado número de indivíduos. Esses são os melhores indivíduos da população atual.

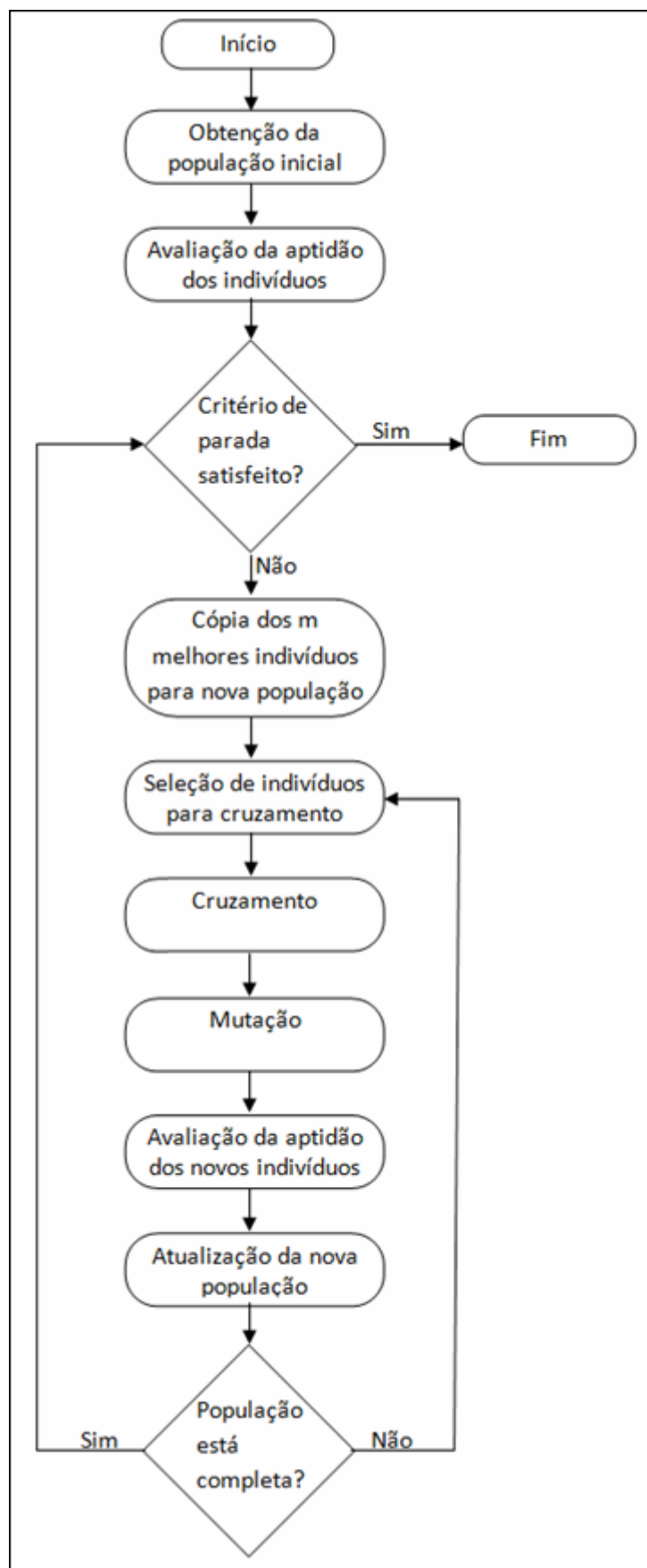


Figura 4.8. Fluxo do processo do Algoritmo Genético com elitismo DJ

Além desse método, outro conceito, estudado em Geiger (2008) (GE), será utilizado. Essa técnica consiste em manter a população intacta até o momento em que um indivíduo mais apto apareça, sendo assim, removendo o indivíduo menos apto. A Figura 4.9, demonstra a dinâmica do método.

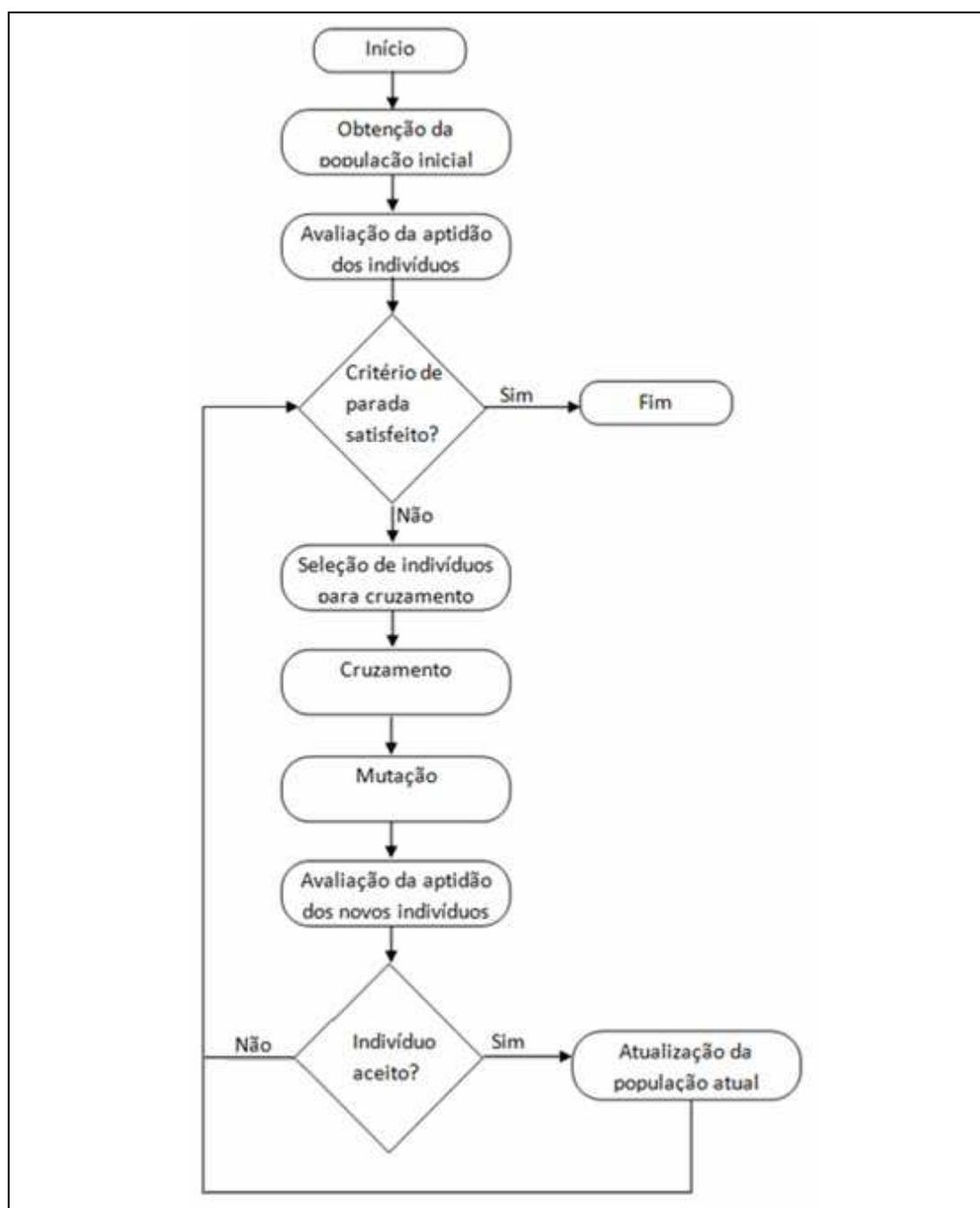


Figura 4.9. Fluxo do processo do Algoritmo Genético com elitismo GE

A Figura 4.9 também apresenta uma variação no modelo apresentado na Figura 3.3, na seção 3.3.3. Nesse caso, está sendo incorporado um teste de aceitação e uma mudança na atualização da população. Depois de os indivíduos

terem sido gerados, é realizado um teste de aptidão. Caso o novo indivíduo for melhor do que algum indivíduo da população atual, ele será inserido na população e o pior indivíduo será descartado. As duas técnicas de elitismo serão testadas e comparadas entre si.

4.3.5 Módulo do Algoritmo Híbrido

O algoritmo híbrido é baseado nas duas metaheurísticas que estão sendo estudadas, Busca Tabu e Algoritmos Genéticos. A Figura 4.10 apresenta o fluxo desse Algoritmo Híbrido. Ele consiste no fluxo do Algoritmo Genético apresentado na Figura 3.3, da seção 3.3.3, com um adicional, a intensificação da busca proporcionada pela Busca Tabu. O Algoritmo Genético irá gerar uma nova população. Dentro dessa nova população, o melhor indivíduo que ainda não tenha sido intensificado passará por um processo de intensificação pela Busca Tabu. Após a intensificação, é verificado se esse indivíduo já se encontra na população, sendo assim descartado. Caso contrário, ele é aceito e será devidamente classificado conforme sua aptidão.

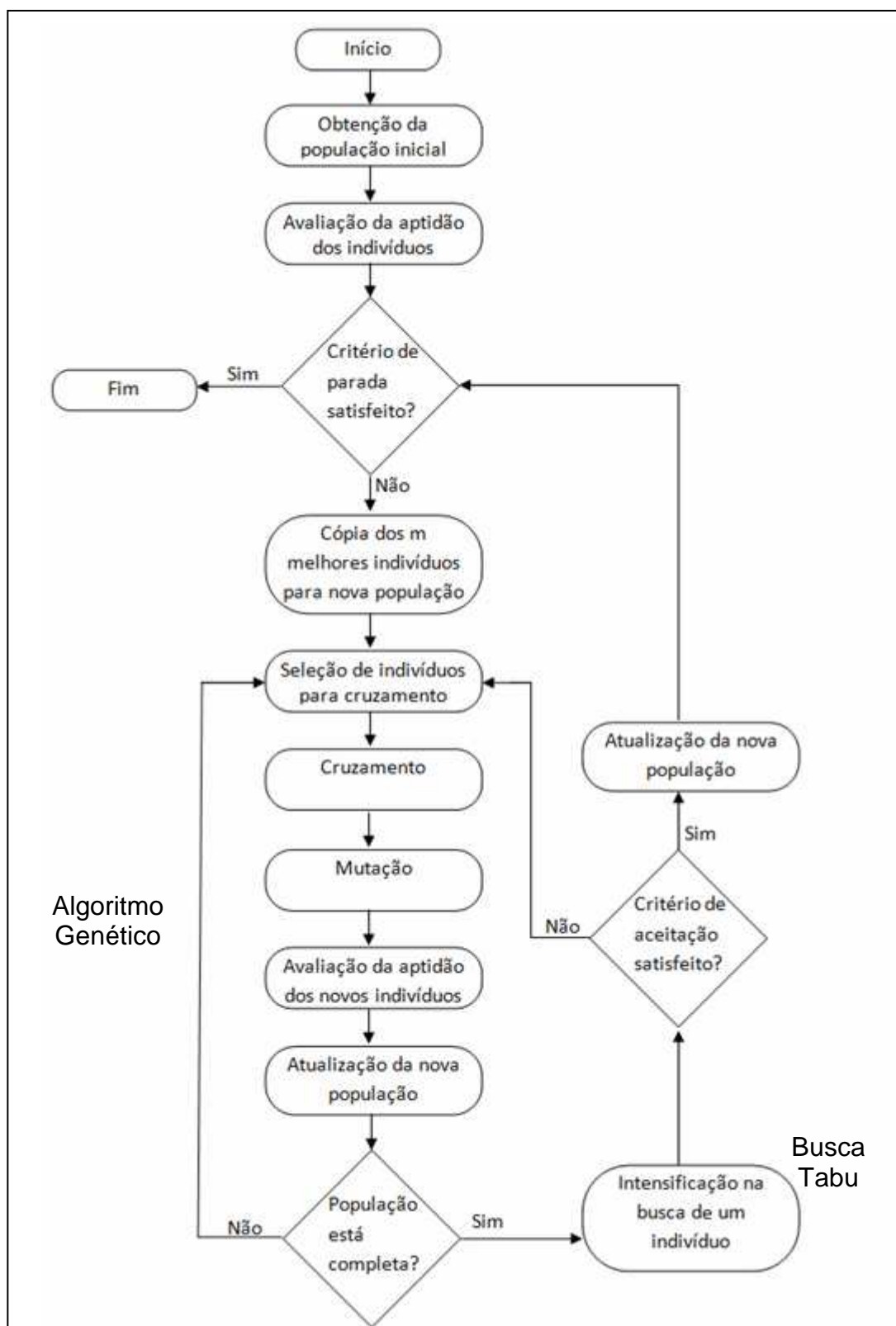


Figura 4.10. Fluxograma do processo do Algoritmo Híbrido

O fluxograma de processo do Algoritmo Híbrido, apresentado na Figura 4.10 apresenta um algoritmo que irá processar em duas etapas. A primeira etapa irá

proporcionar uma diversificação na população, possibilitando que a busca seja direcionada para um espaço de busca ainda não explorado. Essa diversificação será realizada por um Algoritmo Genético, onde seus parâmetros serão aplicados com o objetivo de guiar o algoritmo a espaços, até então, não explorados. Nesse processo utiliza-se a política elitista apresentada por De Jong (1975), onde somente o melhor indivíduo é copiado para a próxima população, de modo que os indivíduos possam se cruzar e abranger uma maior área de busca no espaço de soluções. Os operadores genéticos utilizados são UOBX para realizar o cruzamento e *swap mutation* para realizar a mutação.

A segunda etapa irá proporcionar uma intensificação na busca, forçando a busca a um espaço promissor do espaço de soluções. Essa intensificação será realizada por algoritmo de Busca Tabu, Figura 4.4, seção 4.3.3, onde seus parâmetros serão aplicados com o objetivo de guiar a solução para o ponto ótimo em seu espaço de busca. Nesse processo, utiliza-se a estrutura de vizinhança apresentada na seção 4.3.3.1. Contudo, apesar do método ser semelhante ao apresentado na seção 4.3.3, o processo de intensificação descrito na seção 4.3.3.4, é realizado a cada 10 iterações sem melhor da função objetivo, uma vez que o objetivo é realizar uma exploração mais consistente do espaço de busca.

4.3.5.1 Parâmetros e Definições do Método

Devido à complexidade e o número de parâmetros existentes para este método, todas as definições serão feitas depois dos testes que serão realizados nos outros módulos, para buscar a melhor configuração individual e verificar se, combinadas as técnicas, serão obtidos melhores resultados. Será necessário definir:

- Tamanho da população;
- Método de seleção;
- Taxa de cruzamento;
- Tamanho da população que será clonada;

- Taxa de mutação;
- Tamanho da Lista Tabu;
- Tipos de geração de vizinhança;
- Valor empregado no critério de parada da Busca Tabu;
- Valor empregado no critério de parada do Algoritmo Genético.

Os únicos critérios que são definidos, *a priori*, são o critério de aceitação do novo indivíduo, o tipo de elitismo que será utilizado e o método de seleção. O critério de aceitação consiste em verificar se o novo indivíduo está apto, viável, como uma possível solução e se ele não é igual a nenhum membro da população atual ou da nova população. Com essa validação, não será permitida nenhuma clonagem acidental, somente os melhores indivíduos serão clonados.

O tipo de elitismo definido tem o objetivo de diversificar a solução, permitindo que a Busca Tabu realize uma busca intensificada em espaços ainda não explorados do espaço de solução. Caso fosse utilizado o conceito de elitismo onde somente os indivíduos mais aptos fossem aceitos, toda a população poderia ser intensificada, impedindo o Algoritmo Genético de gerar novos indivíduos, e impossibilitando uma exploração do espaço de busca mais consistente.

O método de seleção utilizado no módulo do Algoritmo Genético e no módulo do Algoritmo Híbrido é conhecido como Roleta, seção 3.3.5. Esse método de seleção faz com que a probabilidade de escolha de um indivíduo esteja diretamente ligada a sua aptidão.

Por fim, poderiam ser utilizadas técnicas, no módulo do Algoritmo Genético e no módulo do Algoritmo Híbrido, para gerar uma nova população quando a mesma perder sua diversidade. Contudo, neste estudo, essa técnica não será desenvolvida. Para minimizar esse problema utiliza-se uma alta taxa de mutação.

5 Validação

A validação do modelo foi realizada com base nos problemas apresentados por Solomon (1987). Vale ressaltar que tratam de problemas de minimização, dessa forma, considera-se os melhores resultados como sendo os de menor valor. Segundo o autor, esses problemas estão divididos em seis conjuntos, aqui separados em três classes: R, C e RC, e duas séries: 1 e 2.

As classes se diferem nos dados geográficos dos clientes. Para os problemas da classe R, os dados geográficos foram gerados aleatoriamente. Na classe C esses dados foram obtidos através de agrupamentos de clientes, enquanto que a classe RC possui, ao mesmo tempo, características comuns às outras duas classes.

As séries definem a flexibilidade das janelas de tempo. A série 1 possui um horizonte de sequenciamento mais restrito, sendo que as janelas de tempo permitem, no máximo, dez clientes por rota. Em contra partida, na série 2 as janelas de tempo possuem grande flexibilidade, muitas vezes não gerando restrições aos problemas e permitem que mais de trinta clientes possam ser atendidos por rota.

Esses problemas se dividem três classes e duas séries, formando um total de seis conjuntos de problemas (R1, R2, C1, C2, RC1, RC2). Em cada conjunto, a localização geográfica é mesma, tendo apenas a variação das janelas de tempos. O conjunto R1 possui doze problemas, variando de R101 à R112, o conjunto R2 possui onze problemas, variando de R201 à R211. Nessa classe, a localização dos clientes foi obtida aleatoriamente. Na classe C, os clientes foram obtidos através da identificação de agrupamentos. Os conjuntos C1 e C2 possuem nove e oito problemas, variando de C101 à C109 e C201 à C208, respectivamente. Por fim, os conjuntos RC1 e RC2 possuem oito problemas cada, variando de RC101 à RC108 e RC201 à RC208, respectivamente, e a classe RC combina distribuições geográficas aleatórias com agrupadas.

Para esses experimentos, a Busca Tabu possui nbmax 1000 e Lista Tabu de tamanho 75. O Algoritmo Genético utiliza o operador de cruzamento UOBX, com a política de elitismo GE e com as taxas de cruzamento e mutação de 90% e 15%,

respectivamente. O Algoritmo Híbrido utiliza as taxa de cruzamento e mutação de 95% e 50%, respectivamente, e Lista Tabu de tamanho 50. Ainda assim, vale lembrar que o valor médio da Função Custo, que representa a Função Objetivo da Busca Tabu e do Algoritmo Híbrido e o *Fitness* do Algoritmo Genético, e o desvio padrão foram obtidos através de cem execuções de cada algoritmo.

Para identificar os autores que foram usados como comparativo foram utilizadas notas de rodapé correspondentes aos autores.

Inicialmente consideraram-se os problemas da classe C1 com 25 clientes. A Tabela 5.1 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.1 – Média dos valores obtidos para a série C100 com 25 clientes

Problemas 25 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média da Função Custo	Desvio Padrão	Média da Função Custo	Desvio Padrão	Média da Função Custo	Desvio Padrão
C101	299,1	48,0	800,5	68,3	209,7	11,7
C102	242,8	26,1	736,1	74,3	207,3	18,7
C103	308,3	27,1	531,9	94,8	206,4	12,4
C104	374,9	28,7	530,4	65,5	231,9	11,1
C105	232,0	29,6	610,0	59,3	262,0	17,0
C106	276,2	41,6	724,3	75,3	237,0	13,3
C107	242,7	30,6	653,8	66,8	255,7	33,6
C108	234,8	36,3	787,7	69,0	211,7	11,0
C109	405,2	47,6	688,5	81,0	192,9	1,4

Observa-se que o Algoritmo Híbrido obteve soluções de melhor qualidade em quase todos os problemas, com exceção dos problemas C105 e C107, onde a Busca Tabu obteve melhores resultados. De modo geral, o Algoritmo Híbrido se mostra mais eficiente, uma vez que a média de seu desvio padrão equivale a apenas 6,2% da média dos valores da Função Custo, diferentemente da Busca Tabu, onde a média de seu desvio padrão é o dobro, 12,6%.

Apesar do Algoritmo Genético não obter soluções de tão boa qualidade, se comparado com a Busca Tabu e o Algoritmo Híbrido, ele produziu um desvio padrão médio equivalente a 11% de sua Função Custo, sendo inferior ao valor produzido pela Busca Tabu.

Os resultados obtidos estão diretamente ligados às características dos algoritmos. A Busca Tabu possui características de busca local, além de ter sido aplicada uma política de intensificação, desse modo, ela tende a realizar uma exploração mais consistente de uma determinada região do espaço de soluções. O Algoritmo Genético inicia sua exploração analisando um amplo espaço de soluções, tendendo a focar sua busca em áreas mais específicas na medida em que novas populações são geradas. Em contra partida, o Algoritmo Híbrido tende a produzir soluções de melhor qualidade, pois promove uma exploração mais ampla do espaço de soluções, característica inicial do Algoritmo Genético, e intensificando a busca em regiões promissoras. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 5 a 20 segundos para cada problema.

Ainda analisando problemas com 25 clientes, na Tabela 5.2 são analisados os resultados obtidos para o conjunto de problemas RC1. Esse conjunto de problemas se difere do conjunto C1 na localização geográfica dos clientes, onde surge uma mistura em clientes agrupados e aleatoriamente distribuídos. A Tabela 5.2 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.2 – Média dos valores obtidos para a série RC100 com 25 clientes

Problemas 25 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
RC101	965,0	82,9	1494,9	170,1	499,7	31,5
RC102	432,4	68,2	1347,1	110,3	429,9	20,2
RC103	400,1	45,5	826,0	153,1	417,3	50,4
RC104	481,2	67,5	666,6	130,8	371,8	58,7
RC105	814,3	78,0	862,2	163,7	454,7	39,2
RC106	441,4	59,6	1007,1	120,0	399,1	20,4
RC107	548,1	89,3	1090,1	111,3	344,0	38,2
RC108	516,8	85,7	693,9	120,0	406,3	43,7

Analisando a Tabela 5.2, pode-se verificar que o Algoritmo Híbrido se mostra superior as demais técnicas na maioria dos problemas. A Busca Tabu obteve melhor média em um dos oito problemas analisados, RC103. Apesar disso, o desvio padrão

médio do Algoritmo Híbrido é equivalente a 9,3% da Função Custo, enquanto que a Busca Tabu obteve 13,2% e o Algoritmo Genético 14,5%.

A variação da localização geográfica dos clientes dificultou, em geral, a obtenção de soluções de boa qualidade, o que pode ser visto pelo aumento da dispersão, se comparado com os dados obtidos na Tabela 1. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 5 a 20 segundos para cada problema.

O conjunto de problemas R1 possui todos os clientes distribuídos aleatoriamente, sendo inicialmente analisados com 25 clientes cada. A Tabela 5.2 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.3 – Média dos valores obtidos para a série R100 com 25 clientes

Problemas 25 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
R101	821,5	161,8	1477,5	217,9	796,5	79,7
R102	885,2	136,2	2092,8	259,4	669,2	38,3
R103	812,5	89,4	1468,4	165,1	528,9	74,1
R104	753,1	84,8	1801,8	157,0	477,2	31,9
R105	1148,4	62,8	1752,8	208,2	727,4	88,2
R106	588,8	90,6	1579,3	185,2	534,5	43,9
R107	860,5	126,7	1316,9	183,2	515,1	68,5
R108	592,0	118,6	1489,4	164,1	481,5	47,3
R109	691,9	117,9	1563,6	193,3	543,4	72,8
R110	585,7	133,7	2041,7	180,0	582,3	48,5
R111	576,5	124,3	996,5	132,9	586,3	49,8
R112	638,8	85,5	950,2	141,5	521,2	72,6

Pode-se observar na Tabela 5.4, que o Algoritmo Híbrido novamente obteve soluções de melhor qualidade que os demais algoritmos. A Busca Tabu obteve melhores resultados somente em um problema, R111, apesar de ter apresentado maior dispersão. O desvio padrão médio gerado pelo Algoritmo Híbrido é equivalente 10,3% da média de sua Função Custo, sendo inferior a 15,6%, obtido pela Busca Tabu, e 12%, obtido pelo Algoritmo Genético. Com base nisso, verifica-se que a distribuição totalmente aleatória dos clientes promove um aumento da dispersão, em comparação aos resultados anteriores. Como informações adicionais,

o tempo computacional necessário para obter essas soluções variou de 5 a 20 segundos para cada problema.

A seguir, analisam-se os resultados obtidos para os conjuntos de problemas da série 2. A diferença entre a série 1 e 2 está nas janelas de tempo. Na série 1 as janelas de tempo provocavam restrições nas rotas, de modo que no máximo 10 clientes são permitidos nas rotas. Na série 2, as janelas de tempo passam a ser flexíveis, de modo que geralmente não causam restrições nas rotas, o que dificulta a obtenção de soluções de boa qualidade.

A Tabela 5.4 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.4 – Média dos valores obtidos para a série C200 com 25 clientes

Problemas 25 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
C201	370,0	37,1	681,1	86,1	251,1	21,4
C202	334,5	42,3	1063,4	103,2	281,0	17,8
C203	416,3	56,8	697,6	65,2	231,0	13,8
C204	378,8	39,6	1016,2	165,7	254,4	41,3
C205	351,4	44,1	778,9	95,4	249,4	16,8
C206	347,3	33,6	900,8	94,1	255,0	30,4
C207	378,7	39,1	900,2	89,7	239,8	21,9
C208	332,6	51,3	945,7	128,4	233,5	15,9

Observa-se que o Algoritmo Híbrido obteve melhores resultados que os demais algoritmos. Além disso, o desvio padrão médio do Algoritmo Híbrido equivale a 8,9% da média da Função Custo, enquanto que para a Busca Tabu o desvio padrão médio equivale a 11,8% e para o Algoritmo Genético equivale a 11,7%. Novamente o Algoritmo Híbrido se mostra mais eficiente, mesma com a maior flexibilidade das janelas de tempo, obtendo soluções de boa qualidade e gerando menor dispersão. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 5 a 20 segundos para cada problema.

O conjunto de problemas RC2 combina as localizações dos clientes, agrupadas e aleatórias, com as janelas de tempo flexíveis. A Tabela 5.5 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio

padrão para os problemas do conjunto RC2 com 25 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.5 – Média dos valores obtidos para a série RC200 com 25 clientes

Problemas 25 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
RC201	691,9	88,1	1702,2	171,6	401,9	34,8
RC202	396,0	47,7	1264,6	118,9	472,1	52,1
RC203	587,6	70,2	1572,3	122,4	386,1	26,1
RC204	402,7	44,1	635,2	144,7	406,2	36,2
RC205	680,9	89,6	1056,1	114,3	402,9	58,8
RC206	555,5	53,7	1216,3	109,6	390,4	58,5
RC207	422,4	74,7	730,9	106,1	341,9	39,3
RC208	558,4	54,1	885,7	128,6	351,7	30,4

Pode-se verificar que, em geral, o Algoritmo Híbrido obteve soluções de melhor qualidade que os demais algoritmos. A Busca Tabu obteve melhores soluções em dois problemas, RC202 e RC204. Apesar da Busca Tabu ter obtido alguns dos melhores resultados, pode-se verificar que o Algoritmo Híbrido obteve um desvio padrão médio equivale a 10,6% da Função Custo enquanto que a Busca Tabu obteve 12,2%. Ainda assim, o Algoritmo Genético não obteve soluções de tão boa qualidade, comparado com os demais algoritmos, gerado um desvio padrão equivalente a 12,3% da média de sua Função Custo. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 5 a 20 segundos para cada problema.

No conjunto de problemas R2, a localização dos clientes foi gerada aleatoriamente e as janelas de tempo são flexíveis. A Tabela 5.6 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto R2 com 25 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.6 – Média dos valores obtidos para a série R200 com 25 clientes

Problemas 25 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
R201	747,2	68,5	1783,6	206,0	542,6	76,3
R202	820,4	48,4	1735,6	150,5	504,4	32,9
R203	846,3	115,5	1837,3	186,1	429,5	36,6
R204	526,2	63,7	1256,8	169,3	405,5	47,6
R205	500,7	57,3	1400,8	123,5	540,1	45,0
R206	503,6	94,2	1872,6	124,7	504,9	53,4
R207	694,7	86,4	1681,9	146,5	459,2	64,1
R208	703,3	82,2	834,4	114,5	389,4	57,0
R209	762,8	98,4	1539,1	172,2	408,8	32,1
R210	532,0	47,6	1132,2	157,4	486,4	29,9
R211	552,4	92,4	1117,1	163,7	442,8	40,6

Observa-se na Tabela 5.6, que o Algoritmo Híbrido obteve, para a maioria dos problemas, os melhores resultados, com exceção dos problemas R205 e R206, onde a Busca Tabu obteve soluções de melhor qualidade. Além disso, analisando o desvio padrão, pode-se verificar que o Algoritmo Híbrido se mostra mais eficiente que os demais algoritmos. O Algoritmo Híbrido obteve um desvio padrão médio equivale a 10,1% da Função Custo enquanto que a Busca Tabu obteve 12,1% e o Algoritmo Genético obteve 11%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 5 a 20 segundos para cada problema.

Analisando as Tabelas 5.1 a 5.6 pode-se verificar que o Algoritmo Híbrido se mostra mais robusto, obtendo melhores resultados na maioria dos problemas e gerando menores desvios padrões. Ainda assim, a Busca Tabu se mostra eficiente em alguns problemas, obtendo resultados de melhor qualidade que as demais técnicas. Em contra partida, o Algoritmo Genético não obteve resultados com qualidade tão boa. Esses resultados estão diretamente ligados às características dos algoritmos. A Busca Tabu tende a intensificar a busca local em uma região promissora do espaço de soluções. Por outro lado, o Algoritmo Genético tem a característica realizar uma busca mais ampla no espaço de soluções. Com base nisso, o Algoritmo Híbrido une as duas características, permitindo que seja realizada uma busca mais abrangente no espaço de soluções ao mesmo tempo que a busca intensifique em áreas promissoras.

Buscando analisar gradualmente a eficiência computacional dos algoritmos, os próximos experimentos são mostrados para problemas com 50 clientes. O aumento do número de clientes promove um aumento da complexidade computacional e as janelas de tempo dos clientes também sofrem alterações. Nos problemas de série 1, as janelas de tempo continuam sendo restritas, contudo, elas são mais flexíveis do que nos problemas analisados com 25 clientes.

Analisam-se, primeiramente, os problemas do conjunto C1 onde as janelas de tempo são restritas e os clientes são agrupados. A Tabela 5.7 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto C1 com 50 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.7 – Média dos valores obtidos para a série C100 com 50 clientes

Problemas 50 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
C101	746,3	62,8	1413,5	168,5	498,8	50,9
C102	574,8	104,8	875,1	165,7	496,2	20,3
C103	368,3	50,2	1776,6	156,8	421,9	54,0
C104	731,0	73,0	761,5	125,0	452,7	40,9
C105	549,4	72,4	1790,2	173,2	383,4	56,9
C106	643,6	54,2	1552	173,2	388,1	31,3
C107	587,4	100,2	1408,8	126,3	439,6	25,7
C108	543,6	73,5	1182,4	132,4	433,2	44,5
C109	771,8	61,6	1651	137,5	505,6	56,8

Observa-se que para os problemas do conjunto C1 com 50 clientes, o Algoritmo Híbrido obteve soluções de boa qualidade, mostrando-se mais eficiente que os demais algoritmos. Pode-se verificar que o Algoritmo Genético gerou um desvio padrão equivalente a 9,6% da média de sua Função Custo, enquanto que a Busca Tabu gerou um desvio padrão médio equivalente a 12,2% e o Algoritmo Genético a 11,7%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 15 a 40 segundos para cada problema.

A seguir são analisados os resultados obtidos para os problemas do conjunto RC1. Vale lembrar que nos problemas da classe RC, os clientes possuem localizações parcialmente agrupadas e parcialmente aleatórias, enquanto que a série 1 designa que as janelas de tempo são restritas. Essa restrição nas janelas de

tempo implica em uma limitação do espaço de soluções. A Tabela 5.8 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto RC1 com 50 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.8 – Média dos valores obtidos para a série RC100 com 50 clientes

Problemas 50 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
RC101	1827,1	242,2	2423,7	418,9	1281,9	92,6
RC102	1375,0	169,1	3345,2	349,0	1015,3	103,6
RC103	1142,8	135,0	3521,3	330,2	769,3	131,0
RC104	863,9	152,5	2374,6	240,0	642,0	50,7
RC105	1711,5	94,4	3417,8	317,6	947,3	62,6
RC106	1262,7	179,4	3103,4	356,0	848,7	81,2
RC107	1135,0	66,9	2223,4	239,6	665,0	39,6
RC108	999,9	66,7	2664,5	285,3	610,7	91,8

Analisando a Tabela 5.8, pode-se verificar claramente que o Algoritmo Híbrido obteve soluções de melhor qualidade que os demais algoritmos, além de gerar um seu desvio padrão médio equivale a 9,9% da média de sua Função Custo. A Busca Tabu obteve desvio padrão médio equivalente a 10,9% da média de sua Função Custo e o Algoritmo Genético obteve desvio padrão médio de 11,8%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 15 a 40 segundos para cada problema.

Novamente a característica de diversificação permite que o Algoritmo Híbrido realize uma exploração mais ampla no espaço de soluções enquanto que uma política de intensificação aprofunde a busca em uma região promissora. Desse modo, soluções de boa qualidade podem ser obtidas.

Ainda considerando a série 1 para problemas com 50 clientes. A Tabela 5.9 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto R1 com 50 clientes. Vale lembrar que para os problemas da classe R, a localização dos clientes é aleatória. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.9 – Média dos valores obtidos para a série R100 com 50 clientes

Problemas 50 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
R101	1325,1	306,9	2950,5	401,3	1378,1	185,7
R102	1448,5	179,9	2359,0	284,6	1228,9	96,6
R103	818,9	184,5	2165,8	232,3	1019,2	130,7
R104	1375,5	108,4	2277,8	288,1	846,5	64,7
R105	1470,5	229,5	2895,2	392,2	912,9	128,4
R106	824,9	120,4	3414,9	275,1	1074,7	157,2
R107	1027,6	119,5	1897,3	298,5	780,4	89,1
R108	1171,6	122,4	1604,0	278,8	618,8	90,6
R109	1366,0	150,9	3162,2	263,4	1096,4	46,3
R110	1017,9	124,4	2145,9	290,5	919,8	128,8
R111	875,3	115,3	2552,9	263,4	863,3	60,1
R112	1168,5	189,9	2573,2	260,3	704,2	53,2

Na Tabela 5.9, diferentemente dos experimentos anteriores, a Busca Tabu obteve melhores resultados em três problemas, R101, R103 e R106. Nos demais problemas o Algoritmo Híbrido obteve melhores resultados. Por outro lado, verifica-se que o Algoritmo Híbrido obteve um desvio padrão médio equivalente a 10,7% de sua média, ligeiramente maior que nos experimentos anterior. Apesar disso, a Busca Tabu não se mostra tão consistente, obtendo um desvio equivalente a 14,2% enquanto que o Algoritmo Genético obteve 12,1%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 15 a 40 segundos para cada problema.

Partindo para a análise de problemas da série 2, analisa-se o conjunto C2. Nesses problemas são considerados 50 clientes com janelas de tempo flexíveis. O aumento da flexibilidade das janelas de tempo permite que um número maior de clientes possa ser visitado em uma mesma rota. Desse modo, aumenta-se o número de possíveis soluções, exigindo maior eficiência dos algoritmos.

A Tabela 5.10 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto C2 com 50 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.10 – Média dos valores obtidos para a série C200 com 50 clientes

Problemas 50 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
C201	711,9	106,1	1413,1	126,4	501,4	43,1
C202	588,8	90,9	1548,5	123,6	413,3	43,7
C203	407,9	57,9	1392,7	172,5	404,6	22,1
C204	373,7	104,2	1164,4	140,8	365,5	47,8
C205	746,0	45,4	1257,9	166,3	416,0	21,9
C206	483,3	43,1	1042,2	132,1	392,2	21,5
C207	568,8	107,6	1113,4	114,6	411,1	67,4
C208	606,0	85,2	1630,1	172,3	432,1	24,8

Pode-se analisar, na Tabela 5.10, que a flexibilização das janelas de tempo exigiu maior eficiência na exploração do espaço de soluções. O Algoritmo Híbrido obteve soluções de melhor qualidade que os demais algoritmos, pois se utiliza de uma política de diversificação para abranger uma grande área de busca no espaço de soluções enquanto que a política de intensificação tende a realizar uma busca local em uma região promissora. Observa-se, também, que o Algoritmo Híbrido obteve um desvio padrão médio equivalente a 8,8% da média de sua Função Custo, enquanto que a Busca Tabu obteve 15% e o Algoritmo Genético 11%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 15 a 40 segundos para cada problema.

Analisando os problemas do conjunto RC2 com 50 clientes, a Tabela 5.11 apresenta a média dos valores da Função Custo obtida por cada algoritmo. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.11 – Média dos valores obtidos para a série RC200 com 50 clientes

Problemas 50 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
RC201	855,7	134,9	2506,5	270,1	853,3	56,1
RC202	645,2	142,9	2290,4	230,3	629,6	123,0
RC203	869,9	89,6	2021,2	378,5	719,8	62,1
RC204	634,1	126,8	1382,8	177,8	542,8	84,8
RC205	970,3	78,6	3102,5	318,0	695,1	74,7
RC206	759,7	172,7	2605,6	302,8	657,5	64,1
RC207	831,4	90,0	2477,3	258,6	568,6	33,5
RC208	936,7	88,5	2278,7	248,2	709,6	88,3

Novamente observa-se que o Algoritmo Híbrido se mostra mais eficiente, obtendo soluções de melhor qualidade em todos os problemas. Esses resultados ficam mais claros à medida que o número de clientes aumenta e as janelas de tempo não restringem a busca. Além da qualidade das soluções, o Algoritmo Híbrido se mostra mais consistente, gerando um desvio padrão médio equivalente a 11,1% da média de sua Função Custo. A Busca Tabu obteve 14,9% e o Algoritmo Genético obteve 11,9%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 15 a 40 segundos para cada problema.

Finalizando a análise de problemas com 50 clientes, são analisados os problemas do conjunto R2. A Tabela 5.12 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto R2 com 50 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.12 – Média dos valores obtidos para a série R200 com 50 clientes

Problemas 50 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
R201	1203,2	190,4	1619,9	315,7	935,3	40,1
R202	1510,8	180,8	2611,3	210,2	778,2	42,6
R203	1179,2	179,8	1924,4	295,5	649,3	56,3
R204	1006,8	108,1	2361,3	242,3	630,2	61,6
R205	1115,6	143,8	1977,4	262,0	965,5	55,0
R206	645,2	186,0	1295,5	197,0	643,2	98,6
R207	1119,2	109,3	2689,8	161,6	795,4	136,4
R208	1265,2	104,2	2325,3	133,3	866,2	35,1
R209	689,5	84,4	1216,9	243,5	686,1	58,9
R210	1290,1	89,6	3007,1	320,3	787,7	128,1
R211	1148,3	148,1	1580,6	218,3	583,6	92,7

Pode-se verificar que o Algoritmo Híbrido obteve soluções de melhor qualidade que os demais algoritmos, como esperado. O aumento das janelas de tempo dificulta a obtenção de soluções de boa qualidade, necessitando de uma técnica explore de modo consistente o espaço de soluções. Ainda assim, o desvio padrão médio obtido pelo Algoritmo Híbrido equivale a 10,1% da média de sua Função Custo, enquanto que a Busca Tabu obteve 13,2% e o Algoritmo Genético obteve 12,5%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 15 a 40 segundos para cada problema.

Analisando as Tabelas 5.7 a 5.12 pode-se verificar que o Algoritmo Híbrido se mostra mais robusto, obtendo melhores resultados na maioria dos problemas e gerando menores desvios padrões. A Busca Tabu se mostrar mais eficiente em três problemas, obtendo resultados de melhor qualidade que o Algoritmo Híbrido. Esses resultados foram obtidos em problemas da série 1 onde as janelas de tempo são mais restritas, diminuindo o número de possíveis soluções. Nos problemas da série 2, as janelas de tempo são mais flexíveis, existindo mais soluções possíveis, de modo que o Algoritmo Híbrido pode usar as características de diversificação e intensificação de modo mais eficiente. Essas características tendem a ficar evidentes à medida que o número de clientes aumentarem. A seguir, as mesmas técnicas serão testadas e analisadas com 100 clientes. Inicialmente serão analisados os problemas da classe C e série 1.

A Tabela 5.13 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto C1 com 100 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.13 – Média dos valores obtidos para a série C100 com 100 clientes

Problemas 100 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
C101	1028,4	67,1	2104,2	194,4	893,8	51,1
C102	1023,8	60,3	2657,9	172,2	870,8	49,9
C103	1035,7	79,7	2976,6	177,7	908,1	57,2
C104	951,9	74,4	2148,0	240,0	870,3	54,8
C105	965,9	75,5	2406,2	166,4	863,1	55,5
C106	1047,5	96,3	2807,8	186,6	851,1	46,4
C107	1104,3	103,2	2937,9	248,9	871,0	58,3
C108	1010,4	109,8	2094,4	217,7	906,5	51,3
C109	956,2	85,2	2147,6	182,3	891,2	60,7

Na Tabela 5.13, pode-se verificar que o Algoritmo Híbrido obteve soluções de melhor qualidade que os demais algoritmos. Além disso, o desvio padrão médio obtido pelo Algoritmo Híbrido equivale a 6,1% da média de sua Função Custo, enquanto que a Busca Tabu obteve 8,2% e o Algoritmo Genético obteve 8,2%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 25 a 65 segundos para cada problema.

Ainda analisando os conjuntos de problemas da série 1 com 100 clientes, a Tabela 5.14 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto RC1 com 100 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.14 – Média dos valores obtidos para a série RC100 com 100 clientes

Problemas 100 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
RC101	1924,0	191,8	1955,5	229,7	1671,0	75,2
RC102	2065,1	107,6	2099,2	204,7	1634,5	88,4
RC103	2156,9	116,5	1970,8	184,6	1703,9	59,2
RC104	1774,3	118,6	1695,0	213,3	1718,5	89,9
RC105	1914,6	132,7	2555,4	232,3	1730,4	57,0
RC106	1752,2	132,1	2174,3	192,5	1699,4	88,8
RC107	1860,3	182,8	3078,8	197,3	1682,5	78,4
RC108	1634,5	98,0	3060,2	199,0	1681,9	82,8

Observa-se, na Tabela 5.14 que na maioria dos problemas, o Algoritmo Híbrido obtém soluções de melhor qualidade. Contudo o Algoritmo Genético se mostra superior no RC104 e a Busca Tabu no problema RC108. Além disso, o Algoritmo Híbrido obteve um desvio padrão médio equivalente a 4,5% da média de sua Função Custo, enquanto que a Busca Tabu obteve 7,1% e o Algoritmo Genético 9,2%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 25 a 65 segundos para cada problema.

A Busca Tabu e o Algoritmo Genético conseguem obter soluções de boa qualidade, geralmente, nos problemas de série 1, onde as janelas de tempo são mais restritas, o que diminui o número de soluções possíveis. Em contra partida, quando as janelas de tempo passam a não restringir a rota do veículo, surgem grandes possibilidades de combinações, dificultando uma exploração consistente do espaço de soluções.

Finalizando a análise dos problemas da série 1 com 100 clientes, a Tabela 5.15 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto R1 com 100 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.15 – Média dos valores obtidos para a série R100 com 100 clientes

Problemas 100 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
R101	1935,7	118,0	2899,6	238,7	1716,1	87,1
R102	1615,2	103,0	3227,3	232,8	1540,3	90,8
R103	1363,0	89,7	3010,9	240,4	1300,6	91,0
R104	1206,8	110,9	2438,7	215,0	1002,5	53,7
R105	1609,0	94,7	3361,9	227,1	1394,9	89,7
R106	1343,2	80,4	2836,4	206,7	1298,4	90,7
R107	1300,3	94,0	3075,0	186,7	1075,5	71,6
R108	1213,9	112,4	2569,5	189,7	975,9	85,5
R109	1432,1	113,0	3212,6	208,4	1217,0	57,4
R110	1574,3	122,3	2450,3	211,3	1128,5	61,5
R111	1291,8	104,1	2429,8	239,9	1053,0	52,6
R112	1082,5	113,5	1950,4	188,7	1029,9	70,3

Observa-se que mesmo se tratando de problemas da série 1, a Busca Tabu e o Algoritmo Genético não conseguiram obter soluções de melhor qualidade que o Algoritmo Híbrido. Além disso, o Algoritmo Híbrido se mostra mais consistente que os demais algoritmos, obtendo um desvio padrão médio equivalente a 6,1% da média de sua Função Custo, enquanto que a Busca Tabu obteve 7,5% e o Algoritmo Genético 7,8%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 25 a 65 segundos para cada problema.

Iniciando a análise nos problemas de série 2, onde as janelas de tempo são mais flexíveis, espera-se que o Algoritmo Híbrido se mostre mais consistente e obtenha soluções de melhor qualidade que os demais algoritmos.

A Tabela 5.16 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto C2 com 100 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.16 – Média dos valores obtidos para a série C200 com 100 clientes

Problemas 100 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
C201	656,6	63,5	3206,7	85,6	602,6	34,4
C202	744,0	79,9	3826,9	86,1	629,2	39,2
C203	732,9	80,2	3612,0	110,9	666,0	24,7
C204	739,7	76,2	2936,6	95,6	646,3	22,9
C205	735,9	67,7	2719,2	81,8	664,6	27,6
C206	720,7	62,5	3920,4	104,1	638,4	20,3
C207	711,4	55,2	2912,2	86,3	600,1	20,3
C208	792,3	75,8	3282,4	86,7	645,0	34,2

Como esperado, observa-se que as melhores soluções foram obtidas pelo Algoritmo Híbrido. Essa característica tende a ser mais evidente nos próximos experimentos devido à complexidade dos problemas analisados. Além disso, o desvio padrão médio obtido pelo Algoritmo Híbrido equivale a 4,4% da média de sua Função Custo. A Busca Tabu obteve um desvio de 9,6% e o Algoritmo Genético de 2,8%. Pode-se verificar que o Algoritmo Genético se mostra mais consistente que o Algoritmo Híbrido, porém não mais eficiente. Apesar da média do desvio padrão ter sido menor, o Algoritmo Genético não conseguiu obter soluções de tão boa qualidade, como o Algoritmo Híbrido. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 25 a 65 segundos para cada problema.

A Tabela 5.17 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto RC2 com 100 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.17 – Média dos valores obtidos para a série RC200 com 100 clientes

Problemas 100 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
RC201	1831,6	84,0	2397,5	154,8	1372,6	77,4
RC202	1318,4	91,4	2322,6	144,1	1266,0	52,0
RC203	1618,0	82,0	2787,0	159,8	1319,7	54,5
RC204	1748,6	87,4	1372,1	132,0	1361,4	71,3
RC205	1548,0	109,6	3160,7	161,9	1276,3	58,8
RC206	1326,2	89,2	3245,5	161,3	1265,7	80,8
RC207	1820,3	88,5	1434,5	133,1	1274,9	65,7
RC208	1636,7	88,5	2278,7	148,2	1459,6	66,5

Novamente observa-se que o Algoritmo Híbrido obteve soluções de melhor qualidade em todos os problemas. Além disso, o Algoritmo Híbrido obteve um desvio padrão médio equivalente a 4,9% da média de sua Função Custo enquanto que a Busca Tabu obteve 5,7% e o Algoritmo Genético 6,7%. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 25 a 65 segundos para cada problema.

Por fim, a Tabela 5.18 apresenta a média dos valores da Função Custo obtida por cada algoritmo e o respectivo desvio padrão para os problemas do conjunto R2 com 100 clientes. Os valores em negrito mostram as melhores médias obtidas e seu desvio padrão.

Tabela 5.18 – Média dos valores obtidos para a série R200 com 100 clientes

Problemas 100 Clientes	Busca Tabu		Algoritmo Genético		Algoritmo Híbrido	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
R201	1474,0	100,2	2862,3	148,3	1175,1	80,5
R202	1523,1	104,2	2346,0	143,9	1041,5	73,0
R203	1317,7	90,3	2392,5	156,5	992,7	63,5
R204	1312,6	97,8	2422,3	167,9	772,9	24,4
R205	1247,5	80,4	2755,5	157,6	952,1	84,6
R206	1036,8	105,6	2992,8	149,5	961,5	30,5
R207	1319,2	109,3	2689,8	161,6	795,4	35,8
R208	1265,2	104,2	2325,3	133,3	866,2	79,5
R209	1106,0	82,5	2676,7	147,0	1042,7	82,9
R210	1289,5	105,6	2967,9	153,5	955,8	52,4
R211	1169,5	110,4	2414,6	130,2	937,4	74,0

Pode-se verificar que, como esperado, o Algoritmo Híbrido obteve soluções de melhor qualidade em comparação com os demais algoritmos. Seu desvio padrão médio equivale a 6,4% da média de sua Função Custo enquanto que a Busca Tabu obteve um desvio equivalente a 7,8% e o Algoritmo Genético equivalente a 5,5%. Apesar do desvio gerado pelo Algoritmo Genético ter sido menor, a qualidade das soluções obtidas pelo Algoritmo Híbrido é muito superior. Como informações adicionais, o tempo computacional necessário para obter essas soluções variou de 25 a 65 segundos para cada problema.

Analisando as Tabelas 5.13 a 5.18, pode-se constatar que o aumento do número de clientes aumenta a complexidade computacional. Ao analisar os resultados obtidos pela Busca Tabu e pelo Algoritmo Genético, verifica-se que fazem

parte dos problemas menos complexo, classe 100, enquanto que para os problemas de maior complexidade, classe 200, faz-se necessário uma técnica mais habilidosa, que consiga diversificar e intensificar de modo eficiente, como no caso do Algoritmo Híbrido.

Na parte final da validação analisam-se as melhores soluções obtidas por cada algoritmo e comparam-se com as melhores soluções conhecidas. No caso dos problemas com 25 e 50 clientes, a maioria das soluções ótimas é conhecida. Para os problemas com 100 clientes, essas informações não são tão consistentes. Desse modo, além de compara as soluções obtidas com as soluções ótimas, for realizada uma comparação com as melhores soluções obtidas por métodos heurísticos.

Inicialmente são comparados os melhores resultados obtidos por cada algoritmo com as soluções ótimas encontradas na literatura. A Tabela 5.19 apresenta os melhores valores obtidos para a Função Custo por cada algoritmo e as soluções ótimas encontradas na literatura, juntamente com seus autores, para os 56 problemas de Solomon (1987) com 25 clientes. Os valores em negrito representam o melhor valor obtido para a Função Custo, considerando os algoritmos estudados. O * define que determinada soluções atingiu o valor ótimo.

Tabela 5.19 – Melhores resultados para os problemas da série RC200 com 25 clientes

Problema 25 Clientes	Busca Tabu	Algoritmo Genético	Algoritmo Híbrido	Soluções Ótimas	Autor
C101	251,1	732,2	191,3*	191,30	KDMSS ¹
C102	190,3*	661,8	190,3*	190,30	KDMSS ¹
C103	281,2	437,1	190,3*	190,30	KDMSS ¹
C104	346,2	464,9	186,9*	186,90	KDMSS ¹
C105	191,3*	550,7	191,3*	191,30	KDMSS ¹
C106	234,6	649,0	191,3*	191,30	KDMSS ¹
C107	212,1	587,0	191,3*	191,30	KDMSS ¹
C108	191,3*	718,7	191,3*	191,30	KDMSS ¹
C109	357,6	607,5	191,3*	191,30	KDMSS ¹
RC101	882,1	1324,8	468,2	461,10	KDMSS ¹
RC102	364,2	1236,8	409,7	351,80	KDMSS ¹
RC103	354,6	672,9	366,9	332,80	KDMSS ¹
RC104	413,7	535,8	313,1	306,60	KDMSS ¹
RC105	736,3	698,5	411,3*	411,30	KDMSS ¹
RC106	381,8	887,1	378,7	345,50	KDMSS ¹
RC107	458,8	978,8	305,8	298,30	KDMSS ¹
RC108	431,1	573,9	362,6	294,50	KDMSS ¹
R101	659,7	1259,6	716,8	617,10	KDMSS ¹

¹ Kohl et al (1999)

Problema 25 Clientes	Busca Tabu	Algoritmo Genético	Algoritmo Híbrido	Soluções Ótimas	Autor
R102	749,0	1833,4	630,9	547,10	KDMSS ¹
R103	723,1	1303,3	454,6*	454,60	KDMSS ¹
R104	668,3	1644,8	445,3	416,90	KDMSS ¹
R105	1085,6	1544,6	639,2	530,50	KDMSS ¹
R106	509,2	1394,1	490,6	465,40	KDMSS ¹
R107	733,8	1133,7	446,6	424,30	KDMSS ¹
R108	473,4	1325,3	434,2	397,30	KDMSS ¹
R109	574,0	1370,3	470,6	441,30	KDMSS ¹
R110	552,0	1861,7	533,8	444,10	KDMSS ¹
R111	552,2	863,6	536,5	428,80	KDMSS ¹
R112	553,3	808,7	448,6	393,00	KDMSS ¹
C201	332,9	595,0	214,7*	214,70	CR ² +L ³
C202	292,2	960,2	214,7*	214,70	CR ² +L ³
C203	359,5	632,4	214,7*	214,70	CR ² +L ³
C204	339,2	950,5	213,1*	213,10	CR ² +KLM ⁴
C205	307,3	683,5	214,7*	214,70	CR ² +KLM ⁴
C206	313,7	806,7	214,7*	214,70	CR ² +L ³
C207	339,6	810,5	214,5*	214,50	CR ² +L ³
C208	281,3	867,3	214,5*	214,50	CR ² +L ³
RC201	653,8	1530,6	367,1	360,20	CR ² +L ³
RC202	348,3	1145,7	420,0	338,00	CR ² +KLM ⁴
RC203	517,4	1449,9	360,0	326,90	IV ⁵ +C ⁶
RC204	358,6	490,5	370,0	299,70	IV ⁵ +C ⁶
RC205	591,3	941,8	338,0*	338,00	L ³ +KLM ⁴
RC206	501,8	1106,7	331,9	324,00	KLM ⁴
RC207	377,7	624,8	302,6	298,30	KLM ⁴
RC208	504,3	757,1	321,3	269,10	C ⁶
R201	678,7	1577,6	466,3	463,30	CR ² +KLM ⁴
R202	772,0	1585,1	471,5	410,50	CR ² +KLM ⁴
R203	730,8	1651,2	392,9	391,40	CR ² +KLM ⁴
R204	462,5	1087,5	357,9	355,00	IV ⁵ +C ⁶
R205	443,4	1277,3	495,1	393,00	CR ² +KLM ⁴
R206	409,4	1747,9	431,5	374,40	CR ² +KLM ⁴
R207	608,3	1535,4	395,1	361,60	KLM ⁴
R208	621,1	719,9	332,4	328,20	IV ⁵ +C ⁶
R209	664,4	1366,9	376,7	370,70	KLM ⁴
R210	484,4	974,8	456,5	404,60	CR ² +KLM ⁴
R211	460,0	953,4	402,2	350,90	KLM ⁴

Na Tabela 5.19, pode-se verificar que Algoritmo Híbrido obteve a solução ótima em 20 dos 56 problemas estudados. A Busca Tabu se saiu melhor em 6 problemas, sendo 2 da série 1, RC103 e R101, e 3 da série 4, RC202, RC204, R205

² Cook e Rich (1999)

³ Larsen (1999)

⁴ Kallehauge et al (2000)

⁵ Irnich e Villeneuve (2005)

⁶ Chabrier (2005)

e R206. Em contra partida, analisando os outros 52 problemas, fica claro que o poder de diversificação pode guiar a busca habilmente para regiões ainda não exploradas, assim como o poder de intensificação consegue extrair soluções de boa qualidade dessas regiões.

A seguir, a Tabela 5.20 apresenta os melhores valores obtidos para a Função Custo por cada algoritmo e as soluções ótimas encontradas na literatura, juntamente com seus autores, para os 56 problemas de Solomon (1987) com 50 clientes. Os valores em negrito representam o melhor valor obtido para a Função Custo, considerando os algoritmos estudados. O * define que determinada soluções atingiu o valor ótimo. As células da coluna Autor não possuem valores denotam que ainda não se sabe qual é a solução ótima para o problema.

Tabela 5.20 – Melhores resultados para os problemas da série RC200 com 50 clientes

Problema 50 Clientes	Busca Tabu	Algoritmo Genético	Algoritmo Híbrido	Soluções Ótimas	Autor
C101	367,6	1420,9	362,4*	362,4	KDMSS ¹
C102	696,2	702,4	398,5	361,4	KDMSS ¹
C103	382,2	687,8	375,5	361,4	KDMSS ¹
C104	360,3	913,9	408,1	358,0	KDMSS ¹
C105	368,3	1675,9	432,7	362,4	KDMSS ¹
C106	441,8	951,8	443,8	362,4	KDMSS ¹
C107	649,8	1228,0	369,6	362,4	KDMSS ¹
C108	672,0	1484,9	402,1	362,4	KDMSS ¹
C109	604,5	1273,1	362,4*	362,4	KDMSS ¹
RC101	1682,7	4194,2	1032,3	944,0	KDMSS ¹
RC102	993,3	1609,1	822,5*	822,5	KDMSS ¹
RC103	1067,9	3013,5	780,3	710,9	KDMSS ¹
RC104	728,3	2308,2	545,8*	545,8	KDMSS ¹
RC105	1540,8	2706,9	1020,4	855,3	KDMSS ¹
RC106	899,4	2702,5	777,6	723,2	KDMSS ¹
RC107	1138,0	1082,4	642,7*	642,7	KDMSS ¹
RC108	1045,4	2126,3	641,2	598,1	KDMSS ¹
R101	1693,1	3549,6	1044,0*	1044,0	KDMSS ¹
R102	1468,0	1953,0	1020,4	909,0	KDMSS ¹
R103	950,6	1480,3	954,4	772,9	KDMSS ¹
R104	1035,3	2499,9	625,4*	625,4	KDMSS ¹
R105	1043,4	3745,5	899,3*	899,3	KDMSS ¹
R106	1017,9	2319,1	793,0*	793,0	KDMSS ¹
R107	711,1*	2366,9	711,1*	711,1	KDMSS ¹
R108	1074,6	2328,6	750,7	617,7	CR+KLM
R109	807,5	3546,0	931,8	786,8	KDMSS ¹
R110	879,8	2169,0	729,7	697,0	KDMSS ¹
R111	707,2*	2722,6	707,2*	707,2	CR ¹ +KLM ⁴

¹ Kohl et al (1999)

Problema 50 Clientes	Busca Tabu	Algoritmo Genético	Algoritmo Híbrido	Soluções Ótimas	Autor
R112	1091,8	2274,4	704,9	630,2	CR ² +KLM ⁴
C201	433,2	804,3	399,1	360,2	CR ² +L ³
C202	642,0	1627,7	360,2*	360,2	CR ² +KLM ⁴
C203	359,8*	630,8	429,2	359,8	CR ² +KLM ⁴
C204	644,2	582,0	451,5	350,1	KML ⁴
C205	545,8	1293,3	426,6	359,8	CR ² +KLM ⁴
C206	372,4	1362,4	359,8*	359,8	CR ² +KLM ⁴
C207	361,4	1464,1	359,6*	359,6	CR ² +KLM ⁴
C208	384,0	601,5	350,5*	350,5	CR ² +KLM ⁴
RC201	1284,3	1328,7	713,3	684,8	L ³ +KLM ⁴
RC202	860,4	1588,9	613,6*	613,6	IV ⁵ +C ⁶
RC203	764,8	1221,5	555,3*	555,3	IV ⁵ +C ⁶
RC204	579,9	1732,5	503,1	444,2	DLP ⁷
RC205	751,3	1399,6	630,2	630,2	IV ⁵ +C ⁶
RC206	673,6	1378,8	672,4	610,0	IV ⁵ +C ⁶
RC207	815,8	2240,2	558,6*	558,6	C ⁶
RC208	1176,4	1665,8	620,9		
R201	952,1	2894,2	797,5	791,9	CR ² +KLM ⁴
R202	940,1	2351,6	698,5*	698,5	CR ² +KLM ⁴
R203	695,3	1761,0	605,3*	605,3	CR ² +KLM ⁴
R204	601,2	1075,2	506,4*	506,4	IV ⁵ +C ⁶
R205	730,7	1564,1	699,9	690,1	IV ⁵
R206	1058,7	2030,5	632,4*	632,4	IV ⁵ +C ⁶
R207	852,6	2983,2	576,1		
R208	1088,4	1272,1	663,6		
R209	630,1	2248,2	600,6*	600,6	IV ⁵ +C ⁶
R210	1164,3	2109,5	645,6*	645,6	IV ⁵ +C ⁶
R211	1097,4	1036,4	569,2	535,5	IV ⁵ +DLP ⁷

Observa-se, na Tabela 5.20, que o Algoritmo Híbrido obteve soluções ótimas em 23 dos 56 problemas analisados enquanto que a Busca Tabu obteve soluções ótimas em 3 problemas, R107, R111 e C203. Ainda assim, os melhores resultados obtidos pela Busca Tabu foram em problemas de menor complexidade, série 1, enquanto que no Algoritmo Híbrido, 13 das 23 soluções ótimas obtidas foram em problemas de maior complexidade.

¹ Cook e Rich (1999)

⁴ Kallehauge et al (2000)

² Cook e Rich (1999)

³ Larsen (1999)

⁴ Kallehauge et al (2000)

⁵ Irnich e Villeneuve (2005)

⁶ Chabrier (2005)

⁷ Danna e Le Pape (2005)

Por fim, comparam-se os melhores resultados obtidos pelos algoritmos analisados com as soluções ótimas e com as melhores soluções heurísticas obtidas para os problemas de Solomon (1987) com 100 clientes.

A Tabela 5.21 apresenta os melhores valores obtidos para a Função Custo por cada algoritmo, as soluções ótimas e as melhores soluções obtidas por métodos heurísticos encontrados na literatura, juntamente com seus respectivos autores. Essa análise é realizada para os 56 problemas de Solomon (1987) com 100 clientes. Os valores em negrito representam o melhor valor obtido para a Função Custo, considerando os algoritmos estudados. O * define que determinada soluções atingiu o valor ótimo e o # define que a solução obtida é melhor que a melhor solução heurística conhecida até o momento. As células da coluna Autor não possuírem valores denotam que ainda não se sabe qual é a solução ótima para o problema.

Tabela 5.21 – Melhores resultados para os problemas da série RC200 com 100 clientes

Problema 100 Clientes	Busca Tabu	Algoritmo Genético	Algoritmo Híbrido	Soluções Ótimas	Autor	Melhores Soluções Heurísticas	Autor
C101	857,0	1046,7	827,3*#	827,3	KDMSS	828.9	RT ⁸
C102	1187,1	1637,9	827,3*#	827,3	KDMSS	828.9	RT ⁸
C103	1267,9	1795,0	826,3*#	826,3	KDMSS	828.0	RT ⁸
C104	1143,5	1759,5	822,9*#	822,9	KDMSS	824.7	RT ⁸
C105	837,6	1097,6	827,3*#	827,3	KDMSS	828.9	RT ⁸
C106	1168,3	884,1	827,3*#	827,3	KDMSS	828.9	RT ⁸
C107	875,3	1026,7	827,3*#	827,3	KDMSS	828.9	RT ⁸
C108	844,6	1629,9	827,3*#	827,3	KDMSS	828.9	RT ⁸
C109	1041,1	2322,9	827,3*#	827,3	KDMSS	828.9	RT ⁸
RC101	1951,5	2247,3	1636,1 #	1619,8	KDMSS	1696.9	TBGGP ⁹
RC102	1464,4	2092,6	1461,6 #	1457,4	CR+KLM	1554.7	TBGGP ⁹
RC103	1639,3	1930,5	1267,3	1258,0	CR+KLM	1261.6	S98 ¹⁰
RC104	1507,2	2609,9	1139,5			1135.4	CLM ¹¹
RC105	1950,8	2202,9	1520,3 #	1513,7	KDMSS	1629.4	BBB ¹²
RC106	1811,1	2256,1	1424,7			1424.7	BBB ¹²
RC107	1605,6	1925,8	1221,6 #	1207,8	IV	1230.4	S97 ¹³
RC108	1274,3	2248,6	1142,4	1114,2	IV	1139.8	TBGGP ⁹
R101	2040,7	3120,5	1657,7	1637,7	KDMSS	1645.7	H ¹⁴
R102	1949,2	2544,0	1480,5 #	1466,6	KDMSS	1486.1	RT ⁸

⁸ Rochat e Taillard (1995)

⁹ Taillard et al (1997)

¹⁰ Shaw (1998)

¹¹ Cordeau et al (2000)

¹² Berger et al (2001)

¹³ Shaw (1997)

¹⁴ Homberger (2000)

¹⁵ Li et al (2001)

¹⁶ Mester (2002)

Problema 100 Clientes	Busca Tabu	Algoritmo Genético	Algoritmo Híbrido	Soluções Ótimas	Autor	Melhores Soluções Heurísticas	Autor
R103	1597,6	1907,2	1213,9 #	1208,7	CR+L	1292.6	LLH ¹⁵
R104	1115,9	1036,1	980,8 #	971,5	IV	1007.2	M ¹⁶
R105	1384,6	2770,4	1367,5 #	1355,3	KDMSS	1377.1	RT ⁸
R106	1610,2	1719,0	1235,4 #	1234,6	CR+KLM	1251.9	M ¹⁶
R107	1550,1	2409,4	1082,5 #	1064,6	CR+KLM	1104.6	S97 ¹³
R108	995,7	1933,5	983,3			960.8	BBB ¹²
R109	1627,4	1532,3	1162,9 #	1146,9	CR+KLM	1194.7	HG ¹⁷
R110	1391,0	2008,4	1084,5 #	1068,0	CR+KLM	1118.5	M ¹⁶
R111	1233,7	2480,6	1064,6 #	1048,7	CR+KLM	1096.7	RGP ¹⁸
R112	1291,5	1027,5	1018,2			982.1	GTA ¹⁹
C201	993,8	1768,7	589,1*#	589,1	CR+KLM	591.5	RT ⁸
C202	774,9	1602,0	589,1*#	589,1	CR+KLM	591.5	RT ⁸
C203	772,5	1267,5	588,7*#	588,7	KLM	591.1	RT ⁸
C204	659,0	888,2	588,1*#	588,1	IV	590.6	RT ⁸
C205	763,1	1918,0	604,7	586,4	CR+KLM	588.8*	RT ⁸
C206	964,7	841,1	594,9 #	586,0	CR+KLM	588.4*	RT ⁸
C207	931,1	773,3	585,8*#	585,8	CR+KLM	588.2	RT ⁸
C208	953,7	1040,6	585,8*#	585,8	KLM	588.3	RT ⁸
RC201	1483,0	1424,8	1267,7 #	1261,8	KLM	1406.9	M ¹⁶
RC202	1433,0	1421,0	1096,6 #	1092,3	IV+C	1367.0	CC ²⁰
RC203	1245,7	1318,4	1070,3			1049.6	CC ²⁰
RC204	930,5	1440,1	798.4			798.4	M ¹⁶
RC205	1588,2	1865,2	1161,4 #	1154,0	IV+C	1297.1	M ¹⁶
RC206	1635,0	1964,7	1265,7	1261,8	KLM	1146.3	H ¹⁴
RC207	1346,3	2107,7	962,9*#	962,9	DLH	1061.1	BVH ²¹
RC208	1291,3	930,0	828.1			828.1	IKMUY ²²
R201	1382,3	1900,9	1159,1 #	1143,2	KLM	1252.3	HG ¹⁷
R202	1512,1	1292,8	1209,0			1191.7	RGP ¹⁸
R203	1109,6	1612,3	939.5			939.5	M ¹⁶
R204	1075,1	1640,9	825.5			825.5	BVH ²¹
R205	1338,1	1489,1	949.8*#	949,8	DLH	994.4	RGP ¹⁸
R206	1182,3	2241,8	1011,8	875,9	DLH	906.1	SSSD ²³
R207	1070,2	938,1	794,0*#	794,0	DLH	893.3	BVH ²¹
R208	755,3	903,3	726.7			726.7	M ¹⁶
R209	1375,2	1699,0	909.1			909.1	H ¹⁴
R210	1036,6	1478,2	900,5*#	900,5	DLH	939.3	M ¹⁶
R211	1070,8	2087,2	892.7			892.7	BVH ²¹

Na Tabela 5.21, pode-se verificar que o Algoritmo Híbrido obteve, em todos os problemas, soluções de melhor qualidade, se comparado a Busca Tabu e ao

¹⁷ Homberger e Gehring (1999)

¹⁸ Rousseau et al (2002)

¹⁹ Gambardella (1999)

²⁰ Czech e Czarnas (2002)

²¹ Bent e Van Hentenryck (2001)

²² Ibaraki et al (2001)

²³ Schrimpf et al (2000)

Algoritmo Genético. Além disso, obteve a solução ótima em 19 dos 56 problemas analisados, sendo 9 em problemas de menor complexidade, série 1, e 10 em problemas de maior complexidade, série 2.

Ao comparar com os melhores métodos heurísticos, o Algoritmo Híbrido obtém soluções de melhor qualidade em 37 dos 56 problemas analisados, mostrando-se mais eficiente que outros métodos heurísticos aplicados ao PRVCJT.

O aumento do número de clientes proporciona um aumento na complexidade computacional, o que evidencia as características das três técnicas. A Busca Tabu intensifica a busca em regiões promissoras, contudo não consegue diversificar de forma eficaz. O Algoritmo Genético se mostra ineficaz em intensificar a busca em regiões promissoras, se comparado a Busca Tabu, contudo, promove uma busca mais ampla no espaço de soluções, chegando a áreas que a Busca Tabu não explorou. Combinando o poder de diversificação do Algoritmo Genético e o poder de intensificação da Busca Tabu, o Algoritmo Híbrido promove uma busca mais abrangente do espaço de soluções sem perder o poder de intensificação, o que pode ser visto nas soluções obtidas.

6 Resultados Obtidos e Análise dos Dados

Neste capítulo são realizados experimentos na Busca Tabu, no Algoritmo Genético e no Algoritmo Híbrido. Busca-se analisar o processo de exploração do espaço de soluções realizado por cada algoritmo, além de situações específicas. Na Busca Tabu, são avaliadas as vantagens da aplicação da política de intensificação. No Algoritmo Genético, considera-se a aplicação de duas políticas de elitismos distintas e três operadores de cruzamento. Além disso, as taxas de mutação e cruzamento são variadas, de modo a analisar características da busca. No Algoritmo Híbrido são avaliados os parâmetros: tamanho da Lista Tabu, taxa de cruzamento e taxa de mutação. O tamanho da Lista Tabu tem impacto no processo de intensificação do algoritmo, enquanto que as taxas de cruzamento e mutação influenciam o processo de diversificação.

Os experimentos serão realizados em três etapas distintas, considerando três problemas da série 2, R205, R206 e RC207. O problema RC207 foi escolhido, pois combina duas características distintas, no que tange a localização geográfica dos clientes além de, ao menos um algoritmo ter obtido a solução ótima. Vale lembrar que nos problemas da classe RC os clientes estão, parte agrupados e parte aleatoriamente distribuídos, diferentemente da classe R, onde os clientes estão aleatoriamente distribuídos. Os problemas R205 e R206 foram escolhidos devido à complexidade e a localização geográfica dos clientes. Apesar de possuírem características semelhantes, o problema R206 também foi escolhido, pois nenhum dos três algoritmos produziu soluções de melhor qualidade que as atualmente conhecidas. Esses problemas foram escolhidos por possuírem maior flexibilidade nas janelas de tempo. A Figura 6.1 mostra a separação das etapas.

Etapa 1	<p>Processo de Busca Tabu</p> <p>Análise da variação do tamanho da Lista Tabu e do nbmax aplicados a uma política de intensificação</p>
Etapa 2	<p>Processo do Algoritmo Genético</p> <p>Análise da variação das taxas de cruzamento e mutação e dos operadores de cruzamentos aplicados a duas políticas de elitismo.</p>
Etapa 3	<p>Processo do Algoritmo Híbrido</p> <p>Análise da variação das taxas de cruzamento e mutação e do tamanho da Lista Tabu.</p>

Figura 6.1 – Etapas do processo de experimentação

Os experimentos analisam o comportamento da busca no espaço de soluções e a qualidade das soluções obtidas. Conforme pode ser observado na Figura 4.1, a primeira etapa dos experimentos diz respeito à aplicação da Busca Tabu nos três experimentos. Nessa etapa analisa-se a influencia do tamanho da Lista Tabu e do nbmax aplicados a uma política de intensificação. A segunda etapa considera o Algoritmo Genético, analisando as taxas de cruzamento e mutação, bem como três operadores genéticos e duas políticas de elitismo. Por fim, na terceira etapa analisa-se o Algoritmo Híbrido estudando a variação do tamanho da Lista Tabu e das taxas de cruzamento e mutação.

6.1 Busca Tabu

A primeira etapa de experimentos tem o objetivo de estudar a influencia do nbmax, da Lista Tabu e da política de intensificação proposta na obtenção de soluções. Analisa-se a combinação de valores distintos para o nbmax e Lista Tabu com e sem a aplicação da política de intensificação. A Tabela 6.1 mostra os valores

propostos para o tamanho do nbmax e a Tabela 6.2 mostra os valores propostos para o tamanho da Lista Tabu.

Tabela 6.1 – Relação de valores para o nbmax

Tamanho do nbmax
50
100
200
750
1000
2000

Tabela 6.2 – Relação de valores para o tamanho da Lista Tabu

Tamanho da Lista Tabu
15
25
50
75
100
200

Os valores apresentados nas Tabelas 6.1 e 6.2 serão combinados entre si, gerando um total de 72 conjuntos de parametrizações para cada problema estudado, sendo 36 deles com e 36 sem a aplicação da política de intensificação.

A Tabela apresentada com os resultados segue o padrão da Tabela 4.4, onde a primeira linha irá designar o problema que está sendo utilizado nos experimentos e se houve a aplicação da política de intensificação, as colunas mostram a variação do tamanho da Lista Tabu e as linhas mostram a variação do nbmax. Os resultados exibidos são o valor médio obtido para a Função Custo (FC) e o desvio padrão (σ) em torno dessa média, com base nos parâmetros mostrados. Além disso, os resultados são analisando com base em quadrantes, conforme mostrado na Tabela 6.3.

Tabela 6.3 - Modelo de tabela de resultados do módulo da Busca Tabu

Problema:								
nbmax		Lista Tabu						
		50	100	200	500	750	1000	2000
15	FC σ	1º Quadrante			2º Quadrante			
25	FC σ							
50	FC σ							
75	FC σ	3º Quadrante			4º Quadrante			
100	FC σ							
200	FC σ							

Os melhores e os piores resultados são destacados, em negrito, para facilitar a análise. Além da análise através dessas tabelas, são traçados comparativos entre as soluções obtidas com e sem a política de intensificação, avaliando os benefícios e peculiaridades, através de gráficos, relacionando percentuais de melhora com tempos computacionais.

Os experimentos são realizados, inicialmente, para o problema RC207. A Tabela 6.4 mostra o cruzamento dos parâmetros nbmax, nas colunas, e o tamanho da Lista Tabu, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, sem a aplicação da política de intensificação. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.4 – BT aplicada ao problema RC207 sem política de intensificação

Problema: RC207 sem intensificação								
Lista Tabu		nbmax						
		50	100	200	500	750	1000	2000
15	FC	2994,8	2859,2	2527,8	2196,6	2681,4	1948,8	1697,4
	σ	835,0	833,8	702,6	610,1	638,7	562,8	531,3
25	FC	2532,6	2075,7	2395,9	2235,9	2583,6	1796,4	2262,5
	σ	806,4	799,1	674,3	558,8	601,4	538,5	503,7
50	FC	2343	1958,2	2043,3	1871,4	1604,0	1625,2	1737,7
	σ	863,1	816,5	798,5	602,4	542,7	578,2	527,4
75	FC	2021,5	1723,1	1492,9	1435,5	1611,9	1433,2	1586,9
	σ	850,7	706,6	568,7	420,2	597,1	392,9	509,2
100	FC	2300,3	2228,4	1752,3	1553,5	1619,5	2002,3	1349,0
	σ	793,4	774,3	801,5	521,2	615,4	551,7	324,8
200	FC	2105,4	1657,5	2173,4	2031,5	1673,3	2088,9	1687,8
	σ	815,2	807,3	659,9	639,1	518,6	526,5	468,2

Observa-se que as melhores soluções foram obtidas com o tamanho da Lista Tabu entre 75 e 100. No caso da Lista Tabu com tamanho 200, não foram obtidas soluções de tão boa qualidade. A Lista Tabu tem a característica de diversificar a busca, guiando a exploração para regiões ainda não exploradas no espaço de soluções. Desse modo, com o aumento desse valor, a Busca Tabu acaba diversificando a busca, impedindo que regiões promissoras sejam melhores exploradas. Percebe-se, também, que as melhores soluções ficam dispersas em mais de um quadrante, 3 e 4, denotando a dificuldade de realizar uma exploração habilidosa no espaço de soluções.

Considerando a qualidade das soluções, pode-se verificar que mesmo com um nbmax de tamanho 200, soluções de boa qualidade foram obtidas. O aumento do nbmax permite que a busca seja realizada por um período maior de tempo, permitindo que a busca realize uma exploração mais abrangente do espaço de soluções. Em contra partida, o aumento do nbmax produz um aumento considerável no tempo computacional.

Analisando os experimentos com a Lista Tabu de tamanho 15, observa-se que o aumento do nbmax permite que a busca explore regiões mais promissoras, em geral, promovendo melhorias na média da Função Custo. Apesar disso, a melhor

média obtida com Lista Tabu 15 foi conseguida com nbmax 2000, e demandou um tempo computacional de 20 segundos para execução. Utilizando Lista Tabu 75, soluções de boa qualidade foram obtidas com nbmax 200, demandando apenas 9 segundos de processamento. Se considerado a melhor média obtida com Lista Tabu 75, onde o nbmax tem tamanho 1000, o tempo computacional necessário foi de 15 segundos, ainda menor que utilizando Lista Tabu 15 e nbmax 2000.

Com base nos dados apresentados na Tabela 6.1, a Figura 6.1, apresenta o comportamento das médias obtidas para o problema RC207, sem a política de intensificação. No eixo x mostra-se a variação do nbmax enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para o tamanho da Lista Tabu.

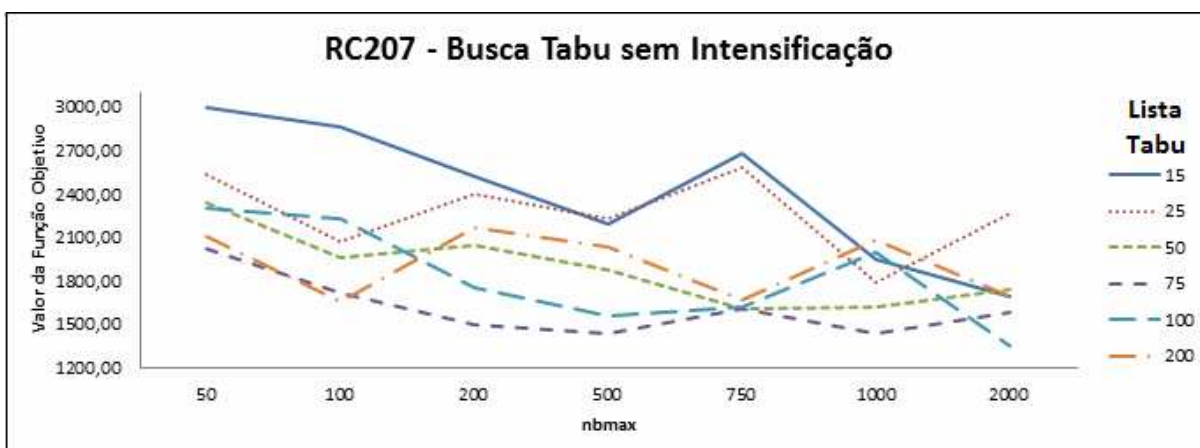


Figura 6.2 – BT aplicada ao problema RC207 sem política de intensificação

A Figura 6.1 torna mais clara a variação das médias da Função Custo, onde se percebe claramente que o aumento exagerado da Lista Tabu, 200, não obteve soluções de boa qualidade.

Ainda para o problema RC207, aplica-se a Busca Tabu com a política de intensificação. Desse modo, espera-se realizar uma exploração mais habilidosa em regiões promissoras do espaço de soluções, obtendo soluções de melhor qualidade que no experimento anterior.

Na Tabela 6.2, é mostrado o cruzamento dos parâmetros nbmax, nas colunas, e o tamanho da Lista Tabu, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com a

aplicação da política de intensificação. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.5 – BT aplicada ao problema RC207 com política de intensificação

Problema: RC207 com intensificação								
Lista Tabu		nbmax						
		50	100	200	500	750	1000	2000
15	FC	2452	2387,3	2109,2	2004,7	1867,4	1842,5	1665,3
	σ	341,9	308,2	326,4	312,5	315,4	308,7	305,1
25	FC	2242,8	1765,4	1596,7	1583,2	1584,5	1602,4	1592,7
	σ	330,9	299,3	317,5	318,4	306,5	299,8	296,2
50	FC	2034	1891,5	1713	1567,1	1502,7	1490,5	1468,3
	σ	321,7	308,5	297,5	281,8	290,8	287,2	270,5
75	FC	1976,8	1657,2	1345,2	1263,9	1255,4	1223,1	1220,9
	σ	319	286,4	270,1	254,2	257,1	277,8	258
100	FC	1988,3	1752,3	1250,7	1249,8	1237,5	1241,5	1230,4
	σ	316,5	277,3	268,4	260,3	255,5	268,9	245,4
200	FC	1986,7	1628,1	1300,5	1283,7	1284,6	1279,2	1265,3
	σ	290,5	259,6	277,6	275,1	266,7	260,1	256,6

Analisando a Tabela 6.2, pode ser percebido que as melhores soluções foram obtidas no quadrante 4, com nbmax entre 750 e 2000 e com Lista Tabu entre 75 e 100. Diferentemente do experimento anterior, com a aplicação da política de intensificação, a Busca Tabu precisa que a busca seja constantemente guiada para novas regiões do espaço, de modo que política possa agir.

Para que isso ocorra, o nbmax deve permitir que a busca se prolongue, um dos motivos pelos quais as melhores soluções foram obtidas com nbmax 1000 e 2000. A Lista Tabu permite que a busca escape de ótimos locais e seja guiada para outras áreas do espaço de soluções. Contudo, um valor muito elevado impede que as regiões promissoras do espaço de soluções sejam devidamente exploradas.

Vale ressaltar que o tempo computacional, quando aplicada uma política de intensificação, normalmente eleva-se consideravelmente. Nesse caso, em termos de valor absoluto, o tempo computacional necessário para executar uma vez cada problema variou entre 7 e 35 segundos.

Para facilitar a visualização da dinâmica das médias obtidas, a Figura 6.2, apresenta o comportamento das médias obtidas para o problema RC207, com a política de intensificação. No eixo x mostra-se a variação do nbmax enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para o tamanho da Lista Tabu.

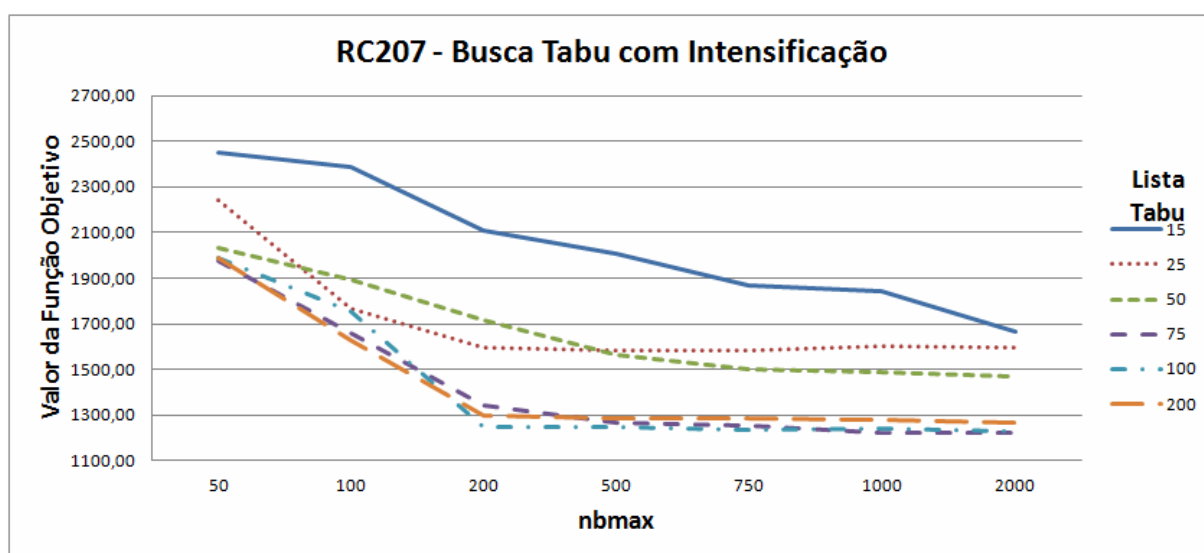


Figura 6.3 – BT aplicada ao problema RC207 com política de intensificação

Pode-se verificar que o aumento do nbmax promove melhorias nos resultados obtidos. Conforme visto na Tabela 6.2, a Lista Tabu de tamanho 75 e 100 obtiveram melhores resultados. Ainda assim, ao comparar as Figuras 6.1 e 6.2, verifica-se que quando aplicada a política de intensificação, de modo geral, o aumento do nbmax promove melhorias nas soluções obtidas. Por outro lado, quando não se aplica a política de intensificação, o aumento do nbmax não garante a melhoria da qualidade da solução.

A seguir, a mesma dinâmica de experimentos é mostrada para o problema R205. Nesse problema, a localização dos clientes é aleatória e as janelas de tempo são flexíveis, exigindo maior destreza para obtenção de soluções de boa qualidade.

Inicialmente aplica-se a Busca Tabu sem a incorporação da política de intensificação para o problema R205 com 100 clientes.

A Tabela 6.3, é mostra o cruzamento dos parâmetros nbmax, nas colunas, e o tamanho da Lista Tabu, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, sem a aplicação da política de

intensificação. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.6 – BT aplicada ao problema R205 sem política de intensificação

Problema: R205 sem intensificação								
Lista Tabu		Nbmax						
		50	100	200	500	750	1000	2000
15	FC	3452,2	3212,7	3037,6	2849,3	2858,7	2801,6	2411,6
	σ	868,2	912,8	751,3	634,8	668	604,2	572,2
25	FC	3352,2	3105,3	3162,5	2587,4	2743,2	2608,5	2348,5
	σ	877,5	836,1	732	585	638,2	584,5	533,5
50	FC	3294,3	3004,5	2245,3	2091	2070,7	2044,9	1946,1
	σ	877,9	853	834,2	666,5	599	585,4	577,5
75	FC	3217	2878,8	2375	2274,7	2092,8	2111,7	1711,2
	σ	930,8	822,3	749,9	645	645,3	556	513,3
100	FC	2658,9	2847,3	2271,6	2049,4	2033	1919,8	2099,1
	σ	848,5	823,1	830	706,9	618,9	576,4	572,8
200	FC	3077,3	2601,8	2128,6	1813,8	2088,7	2075,9	1984,5
	σ	900,1	817,5	678,1	551,6	555,4	531,7	519,1

Na Tabela 6.3, pode-se verificar que soluções de boa qualidade foram obtidas com Lista Tabu de tamanho 50 a 200 e nbmax de 500 a 2000. Essa variação pode ser vista nos quadrantes 2, 3 e 4, onde foram obtidas soluções de boa qualidade.

Comparando as soluções obtidas nas Tabelas 6.1 e 6.3, onde não há a aplicação da política de intensificação, percebe-se que para o problema RC207 o aumento excessivo do tamanho da Lista Tabu não produz soluções de boa qualidade, diferentemente dos dados obtidos para o problema R205. As diferenças nas localizações geográficas dos clientes implicam em maiores dificuldades na obtenção de soluções de boa qualidade. Desse modo, a busca precisa ser guiada para novas regiões do espaço de busca e para isso necessita do aumento da Lista Tabu. Como informações adicionais, o tempo computacional necessário para executar os problemas variou entre 10 e 35 segundos.

Para melhor visualizar a evolução das médias das soluções, a Figura 6.3, apresenta o comportamento das médias obtidas para o problema R205, sem a política de intensificação. No eixo x mostra-se a variação do nbmax enquanto que no

eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para o tamanho da Lista Tabu.

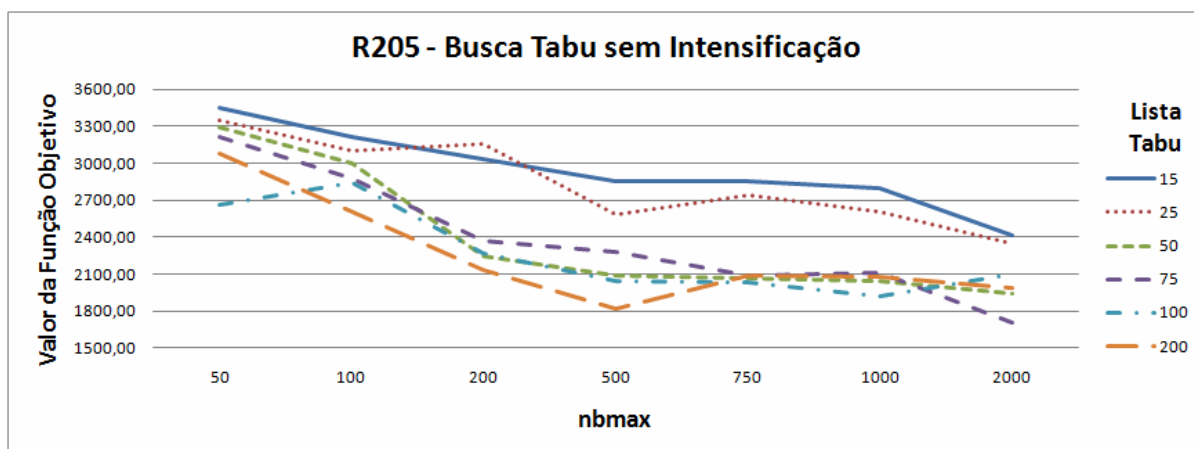


Figura 6.4 – BT aplicada ao problema R205 sem política de intensificação

A Figura 6.3 facilita a visualização da variação das médias da Função Custo, onde se percebe claramente que com nbmax 2000, a Lista Tabu de tamanho 75 obteve melhores resultados. Isso se deve, pois o aumento do tamanho da Lista Tabu força busca para outra região do espaço de soluções, impedindo que a Busca Tabu explore com maior destreza uma região promissora.

Analisando a aplicação da política de intensificação, a Tabela 6.4, é mostra o cruzamento dos parâmetros nbmax, nas colunas, e o tamanho da Lista Tabu, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com a aplicação da política de intensificação. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.7 – BT aplicada ao problema R205 com política de intensificação

Problema: R205 com intensificação								
Lista Tabu		nbmax						
		50	100	200	500	750	1000	2000
15	FC	2712,9	2512,7	2471,3	2452,3	2250,5	2171,4	2024,9
	σ	353,5	358,3	392,1	325	362,6	314,3	368,2
25	FC	2438,8	2346,6	2240,9	2127,5	2038,5	1976,5	1853,3
	σ	347,5	339,4	378,1	362,7	319,5	340,3	315,1
50	FC	2524	2267,6	1998,8	1932,2	1845,5	1770,7	1669
	σ	362,3	351,1	329,5	342,8	341,4	308,3	277,3
75	FC	2416,9	2250,8	1991,4	1920,8	1842,4	1694,9	1668,5
	σ	325,4	348,5	277,4	290	276,9	329,1	275,6
100	FC	2366,4	2226,5	1937,8	1810,2	1788,1	1671	1665,8
	σ	322,4	278,2	314,3	330,7	335,4	319	268,3
200	FC	2346,6	2211,4	1843,2	1775	1719,6	1659,4	1633,7
	σ	348	302,1	301,2	293,8	308,8	283,6	263,3

Observa-se, na Tabela 6.4, que os melhores resultados estão presentes no quarto quadrante, onde estão presentes valores elevados para o nbmax e para a Lista Tabu. Isso ocorre devido às características da política de intensificação. No processo de intensificação, uma determinada região de espaço de soluções é fortemente explorada, desse modo, necessita-se de um valor grande para o nbmax, fazendo com que a busca seja realizada por mais tempo. Por outro lado, uma vez que apenas algumas regiões do espaço de soluções são avaliadas, faz-se necessário um valor grande, também, para a Lista Tabu, guiando a busca para regiões ainda não exploradas. Como informações adicionais, o tempo computacional necessário para executar o problema variou entre 20 e 50 segundos.

A Figura 6.4, apresenta o comportamento das médias obtidas para o problema R205, com a política de intensificação. No eixo x mostra-se a variação do nbmax enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para o tamanho da Lista Tabu.

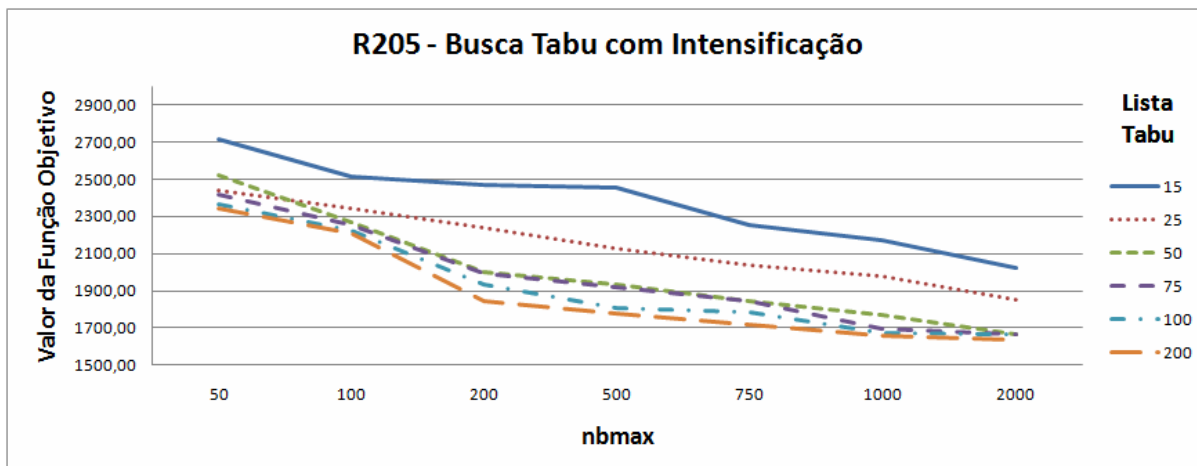


Figura 6.5 – BT aplicada ao problema R205 com política de intensificação

Na Figura 6.4, torna-se mais fácil visualizar a dinâmica apresentada pelas médias da Função Custo. Verifica-se que, com nbmax 2000, os tamanhos de Lista Tabu acima de 50, tendem a obter soluções de boa qualidade. Verifica-se que as Listas Tabus de tamanho 15 e 25 não conseguem direcionar a busca tão habilmente para regiões promissoras quanto os valores mais elevados.

A seguir, analisa-se o problema R206. Nesse problema, as janelas de tempo são menos restritas, aumentando o número possível de soluções, e a localização geográfica dos clientes é aleatória.

A Tabela 6.5, é mostra o cruzamento dos parâmetros nbmax, nas colunas, e o tamanho da Lista Tabu, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, sem a aplicação da política de intensificação. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros. Vale lembrar que a média da Função Custo foi obtida através de cem execuções para cada conjunto de parâmetros.

Tabela 6.8 – BT aplicada ao problema R206 sem política de intensificação

Problema: R206 sem intensificação								
Lista Tabu		Lista Tabu						
			100	200	500	750	1000	2000
15	FC	7078,8	6133,5	6061,7	5124,2	4981,3	5089,3	4533,2
	σ	991,5	950,5	784	717,3	704,7	670,7	608,8
25	FC	5882,8	5213,5	4830,3	4806	4657	3950	3621,3
	σ	919,4	930,8	803,9	657	707,7	625,7	589,5
50	FC	5601	4529,3	5000,4	4142,4	4046,3	3932,1	3875,1
	σ	965,1	916,5	907,1	722,7	623,1	668,4	611,3
75	FC	6087,4	4367	3422,9	3378,1	3476,7	3742,7	3579
	σ	977,9	962,2	812,8	694,9	676,3	605	588,6
100	FC	5133,2	3931	3986,1	3931,4	3803,5	3362	3135,8
	σ	950,2	924,1	910,5	839,7	729,9	626,8	580,9
200	FC	5098,6	4964	4372,9	3697,9	3649,6	3230,9	3125,4
	σ	883,8	902,6	760,6	718,4	584,5	600,7	557,5

Na Tabela 6.5, pode-se verificar que as melhores soluções foram obtidas com Lista Tabu de tamanho 75 a 200 e nbmax de 500 a 2000. Essa variação pode ser vista nos quadrantes 2 e 4.

Observa-se que, assim como para o R205, a Busca Tabu necessita de valores elevados para a Lista Tabu. A elevação do valor desse parâmetro força a busca para novas regiões do espaço de soluções, permitindo que novas áreas possam ser exploradas. Contudo para que isso ocorra de forma eficiente, os valores do nbmax também devem ser elevados. Como informações adicionais, o tempo computacional necessário para executar os problemas variou entre 10 e 40 segundos.

Para melhor visualizar a evolução das médias das soluções, a Figura 6.5, apresenta o comportamento das médias obtidas para o problema R206, sem a política de intensificação. No eixo x mostra-se a variação do nbmax enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para o tamanho da Lista Tabu.

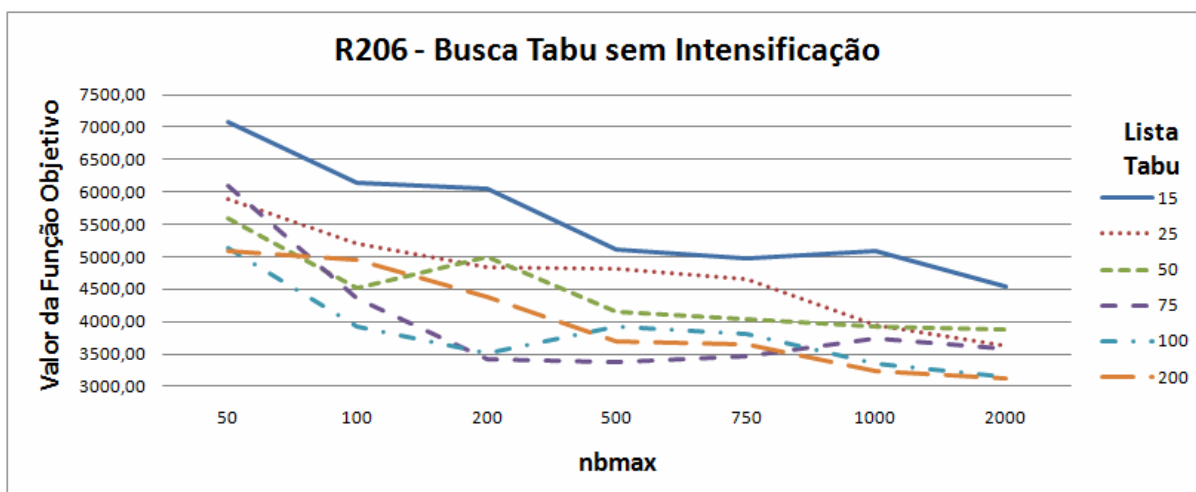


Figura 6.6 – BT aplicada ao problema R206 sem política de intensificação

Na Figura 6.5, pode-se analisar a evolução dos valores das soluções para os determinados conjuntos de parâmetros. Percebe-se que, inicialmente, o aumento do nbmax produz uma melhora considerável no valor da Função Custo, porém, após o valor de 500, pouca variação ocorre se comparada com a variação inicial.

O próximo experimento busca analisar a política de intensificação aplicada ao problema R206. A Tabela 6.6, mostra o cruzamento dos parâmetros nbmax, nas colunas, e o tamanho da Lista Tabu, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com a aplicação da política de intensificação. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.9 – BT aplicada ao problema R206 com política de intensificação

Problema: R206 com intensificação								
Lista Tabu		nbmax						
		50	100	200	500	750	1000	2000
15	FC	3297,9	2918,5	2953,6	2768,2	2593,3	2601,7	2599,8
	σ	421,1	408,6	459,8	358,1	418,9	363,5	409,1
25	FC	2423,5	2297	2296,3	2069,9	2069,4	2084,6	2150,3
	σ	409,1	393,8	448,3	402,7	373,1	385,7	365,9
50	FC	2198,9	1892,9	1564,4	1545,5	1657,1	1515,7	1578,4
	σ	423,1	390,2	367	406,9	380,7	352,3	328,6
75	FC	2319,1	1532,2	1342,3	1349,2	1444,3	1491,6	1326,9
	σ	383,6	404,6	317,4	319,7	214,1	379,6	112,2
100	FC	2323,3	1804,8	1337,7	1323,7	1267,2	1283,8	1257,4
	σ	383,6	323,7	359,4	379,7	195,3	171,7	118,3
200	FC	2481,1	1745,4	1397,5	1483,7	1326,5	1262,4	1263,6
	σ	391,8	333,2	356,1	349,5	148,6	132,9	104,4

Observa-se, na Tabela 6.6, que as melhores soluções obtidas encontram-se no quarto quadrante, com Lista Tabu entre 100 e 200 e nbmax entre 750 e 2000. A política de intensificação exige que a solução seja constantemente guiada para regiões ainda não exploradas. Para que isso ocorra, faz-se necessário que os valores da Lista Tabu sejam elevados, forçando a busca para outras regiões do espaço de soluções. Desse modo, o aumento do nbmax permite que a busca seja executada por mais tempo, permitindo uma busca mais consistente do espaço de soluções. Como informações adicionais, o tempo computacional necessário para executar cem vezes cada configuração de parâmetros variou entre 20 e 60 segundos.

Para facilitar a visualização da dinâmica das médias obtidas, a Figura 6.6, apresenta o comportamento das médias obtidas para o problema R206, com a política de intensificação. No eixo x mostra-se a variação do nbmax enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para o tamanho da Lista Tabu.

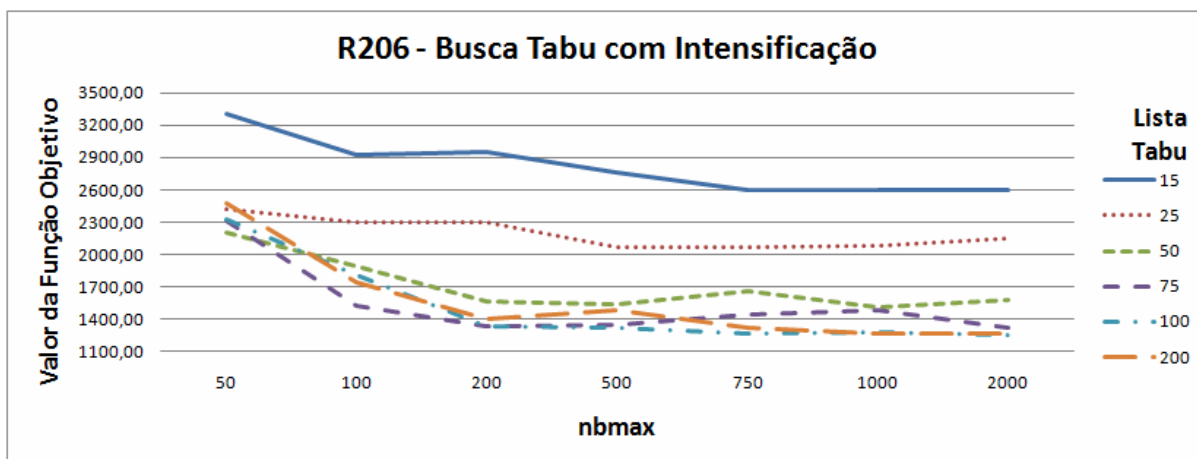


Figura 6.7 – BT aplicada ao problema R206 com política de intensificação

Percebe-se, analisando a Figura 6.6, que com um nbmax muito pequeno, 50, a busca não consegue explorar o espaço de soluções com grande destreza. Pode-se ver que o aumento desse parâmetro permite que a busca obtenha melhores soluções. Nota-se uma sensível melhora entre os nbmax 50 e 200, porém nos demais, essa melhora não é sentida com tanta intensidade.

Visando analisar o processo de busca no espaço, realizado pela Busca Tabu, não aplicando a política de intensificação, a Tabela 6.10 mostra o número de incidências das melhores soluções obtidas com cada conjunto de parâmetros, analisando as Tabelas 6.4, 6.6 e 6.8.

Tabela 6.10 – Análise de busca sem política de intensificação

Sem intensificação							
Lista Tabu	nbmax						
	50	100	200	500	750	1000	2000
15							
25							
50							1
75						1	2
100			1	2		2	2
200				2		1	2

Observa-se que quando não se aplica a política de intensificação, a busca obteve as melhores soluções com diferentes conjuntos de parâmetros. Nesse caso, soluções de boa qualidade, considerando apenas as tabelas analisadas, foram obtidas utilizando diversos conjuntos de parâmetros.

A Tabela 6.11 mostra o número de incidências das melhores soluções obtidas com cada conjunto de parâmetros, analisando as Tabelas 6.5, 6.7 e 6.9, onde se aplica a política de intensificação.

Tabela 6.11 – Análise de busca com política de intensificação

Com intensificação							
Lista Tabu	nbmax						
	50	100	200	500	750	1000	2000
15							
25							
50							
75						1	2
100					2	3	3
200						2	2

Na Tabela 6.11, diferentemente da Tabela 6.10, com a aplicação da política de intensificação a busca se torna mais consistente obtendo as melhores soluções com nbmax entre 750 e 2000 e Lista Tabu entre 75 e 200.

Buscando compara a qualidade das soluções geradas com e sem a política de intensificação, a Figura 6.7 mostra o percentual de melhora obtido comparando a melhor solução obtida, sem aplicação da política de intensificação local, com a melhor solução obtida, com aplicação da política de intensificação local. Esse comparativo é realizado para cada uma dos três problemas, sendo que a solução inicial para o problema RC207 é de 5925,9, para o problema R205 é de 8241,5 e para o problema R206 é de 9413,8. As melhores soluções obtidas sem a política de intensificação foram 1954,2 para o problema RC207, 2014,7 para o problema R205

e 2830,1 para o problema R206. Com a aplicação da política de intensificação, as melhores soluções obtidas foram 1346,3 para o problema RC207, 1338,1 para o problema R205 e 1182,3 para o problema R206.

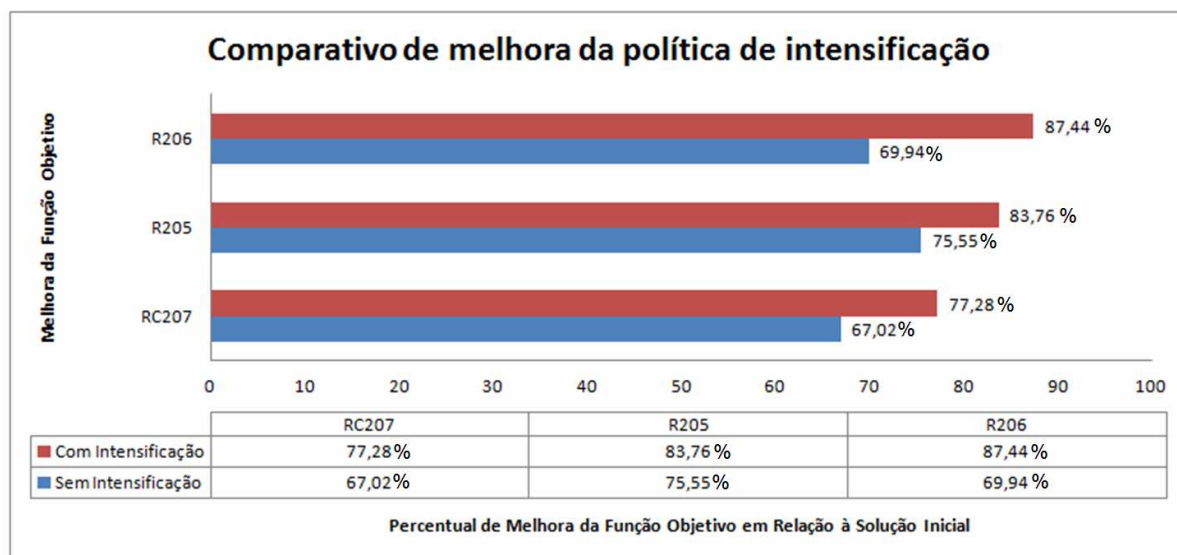


Figura 6.8 – Comparativo de melhora da Função Objetivo considerando a política de intensificação

Analisando a Figura 6.7, pode-se verificar claramente que a aplicação da política de intensificação gera soluções de melhor qualidade. Essa diferença se faz mais sensível para o problema R206, onde a diferença é de 17,5 pontos percentuais.

A aplicação de políticas de intensificação, normalmente causa um aumento considerável no tempo computacional. Considerando o universo dos experimentos, pode-se constatar que o tempo computacional sofre uma variação de apenas poucos segundos, não sendo comprometido.

A seguir, é analisado o comportamento de convergência da busca tabu com e sem a aplicação da política de intensificação local. A Figura 6.8 mostra o valor da função objetivo em relação ao número de iterações para o problema RC207. Nesse experimento, o critério de parada adotado foi um total de 500 iterações e a Lista Tabu possui tamanho 50.

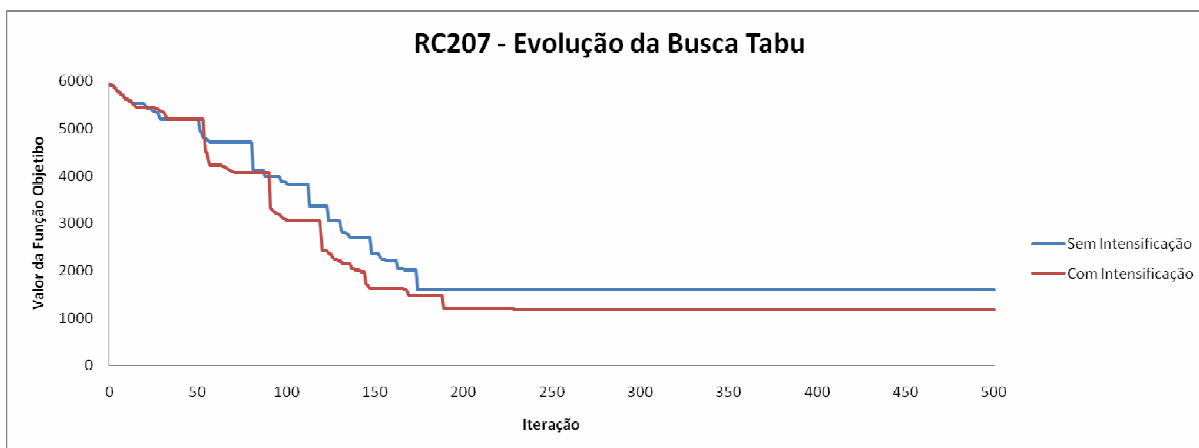


Figura 6.9 – Evolução da Busca Tabu com e sem intensificação para o problema RC207

Na Figura 6.8, pode-se verificar que a cada processo de intensificação, ocorre uma queda considerável no valor da função objetivo. Isso se deve ao processo de exploração ser realizado em uma área considerada como próspera pela política de intensificação. Ainda assim, verifica-se que ao aplicar a política de intensificação, obtém-se uma convergência mais acentuada, fazendo com que soluções de melhor qualidade sejam obtidas com menos iterações.

Avaliando o problema R205, a Figura 6.9 mostra o valor da função objetivo em relação ao número de iterações. Nesse experimento, o critério de parada adotado foi um total de 500 iterações e a Lista Tabu possui tamanho 50.

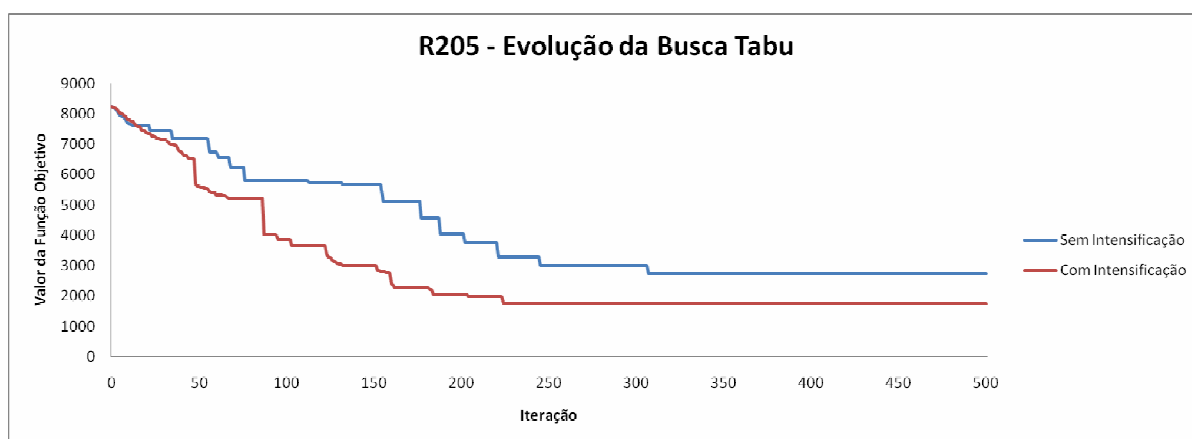


Figura 6.10 – Evolução da Busca Tabu com e sem intensificação para o problema R205

Observa-se, novamente, uma convergência mais acentuada com a aplicação da política de intensificação. Isso se deve a intensificação direcionar a busca para regiões historicamente conhecidas como de boa qualidade.

Por fim, a Figura 6.10 mostra o valor da função objetivo em relação ao número de iterações para o problema R206. Nesse experimento, o critério de parada adotado foi um total de 500 iterações e a Lista Tabu possui tamanho 50.

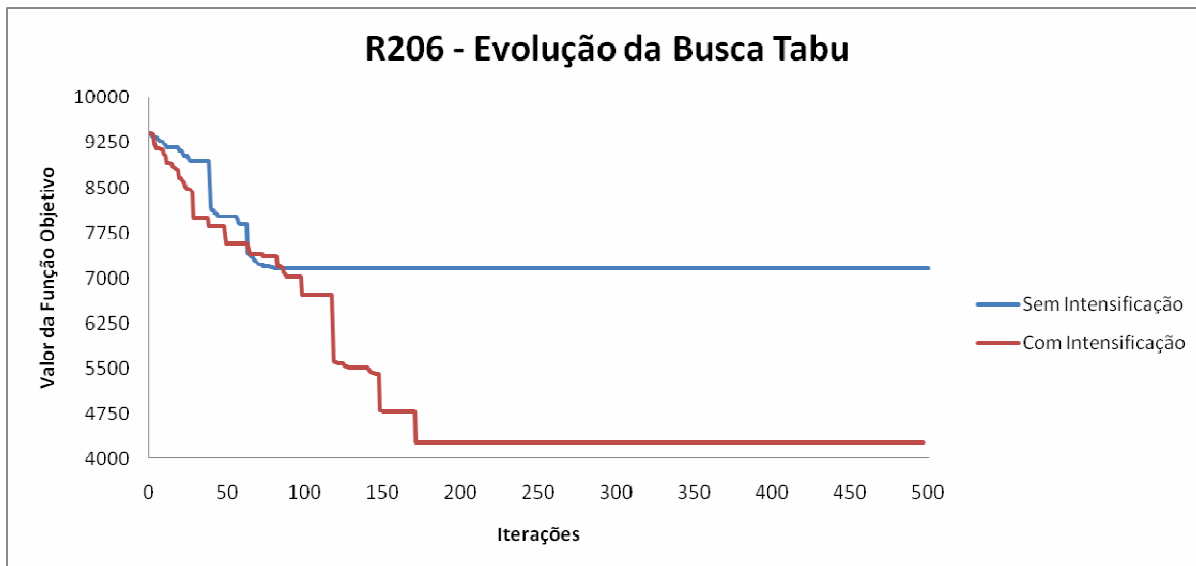


Figura 6.11 – Evolução da Busca Tabu com e sem intensificação para o problema R206

Na Figura 6.10, torna-se evidente que a Busca Tabu não conseguiu explorar, de forma consistente, o espaço de soluções. Observa-se que a aplicação da política de intensificação promove melhorias graduais no valor da função objetivo. Isso se deve pela dinâmica da política, onde a busca é constantemente direcionada para regiões promissora no espaço de soluções.

6.2 Algoritmo Genético

A segunda etapa de experimentos tem o objetivo de estudar a influencia da taxa de cruzamento em três tipos de operadores de cruzamento, seção 4.3.4.2, da taxa de mutação e a aplicação de duas políticas de elitismos, aqui definidas como DJ, Figura 4.8, e GE, Figura 4.9, seção 4.3.4.4.

O tamanho da população será de 1000 indivíduos. A Tabela 6.12 mostra os valores propostos para a taxa de cruzamento e a Tabela 6.13 mostra os valores propostos para a taxa de mutação.

Tabela 6.12 – Relação de valores para a taxa de cruzamento

Taxa de cruzamento
5
10
25
50
75
90

Tabela 6.13 – Relação de valores para taxa de mutação

Taxa de mutação
0
2
5
10
15
25

Os resultados obtidos serão apresentados em tabelas seguindo o padrão da Tabela 6.14 onde a primeira linha irá designar o problema que está sendo utilizado nos experimentos e a política de elitismo utilizada e a segunda linha irá designar o operador de cruzamento utilizado. As colunas mostram a variação da taxa de cruzamento e as linhas mostram a variação da taxa de mutação. Os resultados exibidos são o valor médio obtido para a Função Custo (FC), *fitness*, e o desvio padrão (σ) em torno dessa média, com base nos parâmetros mostrados. Além disso, os resultados são analisando com base em quadrantes, conforme mostrado na Tabela 6.14.

Tabela 6.14 - Modelo de tabela de resultados

Problema:							
Operador de Cruzamento:							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC σ	1º Quadrante			2º Quadrante		
2	FC σ						
5	FC σ						
10	FC σ	3º Quadrante			4º Quadrante		
15	FC σ						
25	FC σ						

Os melhores e os piores resultados são destacados, em negrito, para facilitar a análise. A análise dos dados é realizada utilizando, individualmente, as políticas de elitismo. Além da análise através dessas tabelas, são traçados comparativos entre as soluções obtidas com cada uma das políticas de elitismo, avaliando os benefícios e peculiaridades, através de gráficos, relacionando percentuais de melhora com tempos computacionais.

Inicialmente analisa-se o problema RC207. Nesse problema as localizações dos clientes são parte agrupada e parte aleatória. Além disso, esse problema pertence à série 2, onde as janelas de tempo são mais flexíveis, aumento o número de possíveis soluções.

A Tabela 6.15, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador OBX e política DJ. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.15 – AG aplicada ao problema RC207 com operador OBX e com a política DJ

Problema: RC207 – DJ							
Operador de Cruzamento: OBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5698,9	5201,4	5137,9	5182,9	5363,6	5132,8
	σ	1361,5	1315,8	1494,6	1082,4	1112,6	1191,1
2	FC	5977,6	5865,7	5733,9	5252,5	5029,5	5017,1
	σ	947,6	1062,2	1350,7	1349,5	1328,8	1347
5	FC	5956,4	5454,9	5763,8	5385,4	5093	5449,4
	σ	1461	1451,7	1389,4	939,2	1342,1	1014,7
10	FC	5905	5685,5	5770,9	5048	5037,6	5388,6
	σ	1399,8	1044,9	1382,7	1406,8	1248	1476,2
15	FC	5750,2	5581,7	5268,6	5407,2	5587,3	5560,5
	σ	1102,2	1312,3	905,4	1030	1430,3	1004,9
25	FC	5937	5947,1	5576	5905,8	5936,6	5661,4
	σ	957,6	1201,9	1258,8	1081,6	1104,6	1225,3

Observa-se que as solu es de melhor qualidade foram obtidas nos quadrantes 2 e 4, onde a taxa de muta o varia de 2 a 10, e a taxa de cruzamento varia de 50 a 90. Pode-se perceber que a eleva o das taxas de muta o n o gerou solu es de boa qualidade. A muta o tem a fun o de promover a diversidade da popula o, permitindo que novas caracter sticas sejam inseridas nos indiv duos. O aumento dessa ocorr ncia dificulta a explora es de regi es promissoras do espa o de solu es, dificultando a obten o de solu es de boa qualidade.

Ainda assim, pode-se verificar que, de modo geral, o processo de cruzamento provoca a converg ncia da popula o, reduzindo a diversidade gen tica e guiando a busca para um  timo local. Por outro lado, o processo de muta o, promove uma diversifica o na popula o, guiando a busca para outras  reas do espa o de solu es. O operador OBX realiza diversas combina es gen ticas, dificultando o processo de converg ncia da popula o, se comparado aos demais operadores. Como informa es adicionais, o tempo computacional necess rio para executar cada configura o de par metros variou entre 30 e 50 segundos.

Para facilitar a an lise dos dados da Tabela 6.15, a Figura 6.12, apresenta o comportamento das m dias obtidas para o problema RC207, com o operador OBX e

a política de elitismo DJ. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

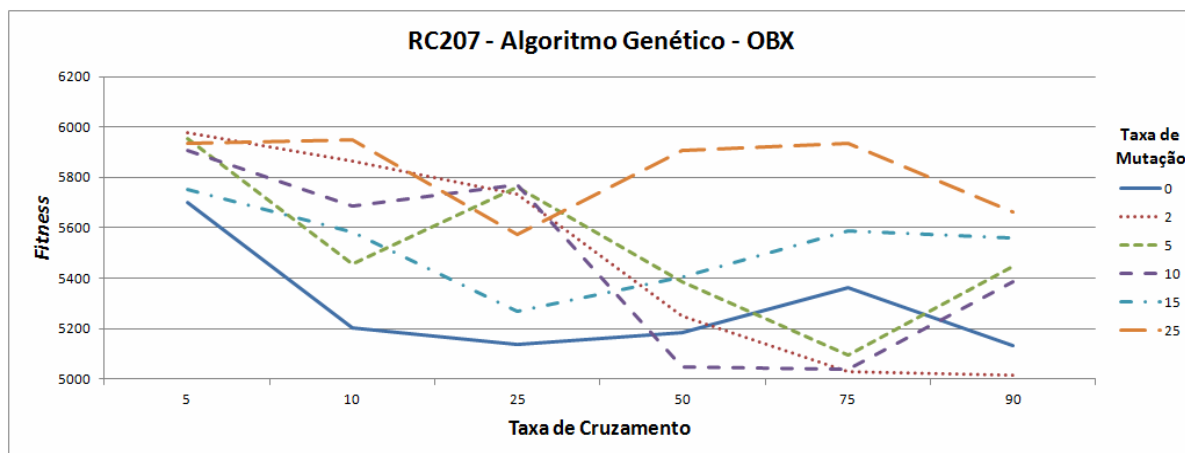


Figura 6.12 – Evolução do Algoritmo Genético com operador OBX e com a política DJ para o problema RC207

Analisando a Figura 6.12, pode-se verificar grande oscilação nos resultados obtidos. Observa-se que as elevadas taxas de mutação não obtiveram resultados de boa qualidade.

Ainda analisando o problema RC207, a Tabela 6.16, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador PMX e política DJ. Os valores destacados na Tabela mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.16 – AG aplicada ao problema RC207 com operador PMX e com a política DJ

Problema: RC207 – DJ							
Operador de Cruzamento: PMX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5477,5	5424,3	5300,6	5080,4	5263,2	5155,9
	σ	1065,4	1117,9	1076,1	1136,1	1076,1	1244,5
2	FC	5496,2	5416,4	5339,7	5250,7	5158,2	5216,3
	σ	1136,8	1192,2	1193,2	1103,3	1066,1	1186
5	FC	5471,1	5315,5	5288,7	5310,6	5117,6	5111,5
	σ	964,1	991,4	1197,5	1173	1103,2	1072,4
10	FC	5410,8	5328,1	5264,8	5239,9	5228	5124,1
	σ	1192,1	1252,2	1236,2	1191,3	997,2	1099,6
15	FC	5495,5	5371	5124,2	5111,1	5183,1	5110,2
	σ	1050,3	1013	1099,7	1147,4	1038,1	1203,5
25	FC	5445,3	5236,4	5284,4	5186	5292,4	5304,8
	σ	1024,5	1052,1	1258	1260,8	1300,5	953,6

Pode-se observar na Tabela 6.16, que as soluções de melhor qualidade foram obtidas nos quadrantes 2, 3 e 4. Verifica-se que, assim como para o operador OBX, a elevação excessiva da taxa de mutação dificulta a obtenção de soluções de boa qualidade, enquanto que o aumento da taxa de cruzamento produz melhores resultados. Diferentemente do operador de cruzamento OBX, o operador PMX, realiza menos recombinações genéticas, mantendo muitas características comuns aos pais, facilitando o processo de convergência. Contudo, ao aumentar excessivamente a taxa de mutação, a busca acaba sendo guiada para novas regiões do espaço de busca, sem conseguir, efetivamente, intensificar a busca em um ponto ótimo. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

Para melhor analisar os dados da Tabela 6.16, a Figura 6.13, apresenta o comportamento das médias obtidas para o problema RC207, com o operador PMX e a política de elitismo DJ. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

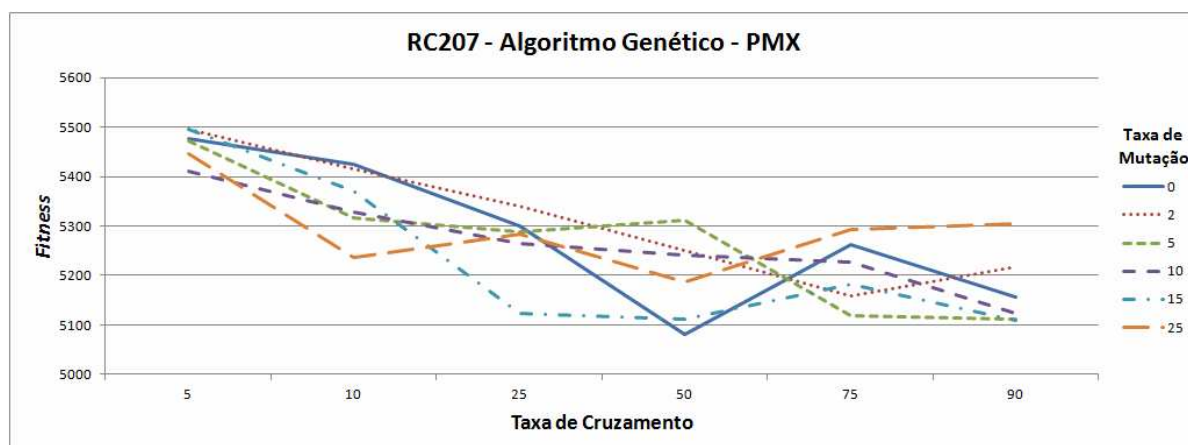


Figura 6.13 – Evolução do Algoritmo Genético com operador PMX e com a política DJ para o problema RC207

Observa-se, na Figura 6.13, que o aumento da taxa de cruzamento tende a produzir soluções de boa qualidade. Pode ser verificado que, com exceção das taxas de cruzamento 25 e 2, as demais combinações de parâmetros tendem a obter soluções de melhor qualidade.

Analisando o operador UOBX para o problema RC207, a Tabela 6.17, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador UOBX e política DJ. Os valores destacados na Tabela 6.15 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.17 – AG aplicada ao problema RC207 com operador UOBX e com a política DJ

Problema: RC207 – DJ							
Operador de Cruzamento: UOBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4893	4861,3	4851,4	4850	4810	4727,7
	σ	1144,8	1170,8	908,7	904	936,7	912,5
2	FC	4837	4715,9	4834,4	4708,8	4828,9	4848,1
	σ	990,3	980,7	973,5	864,2	938	895,6
5	FC	4857,3	4866,9	4719,1	4741,8	4873	4722,4
	σ	1023,8	1187,2	961,2	1044,8	1144,1	1007,4
10	FC	4809,6	4888,3	4752,7	4802,1	4732,9	4731,4
	σ	1014,8	1031,6	871,1	1066,6	1085,1	1099,1
15	FC	4805,3	4766,4	4711	4872,9	4858,7	4810,3
	σ	982,5	1146,6	1164,5	973,6	1045,7	1141,1
25	FC	4863,4	4851,1	4886,3	4893,8	4834,4	4724,4
	σ	1050,7	964,5	1153,1	938,1	1009,7	1168,2

Pode-se observar na Tabela 6.17, que foram obtidas solu es de boa qualidade em todos os quadrantes. Isso se deve pelas caracter sticas do operador de cruzamento UOBX. Ainda assim, verifica-se que, apesar de solu es de boa qualidade terem sido obtidas com altas taxas de muta o, o maior n mero de solu es de melhor qualidade foram obtidos com taxas de muta o entre 2 e 5, onde ocorre uma converg ncia da popula o mais acentuada. Como informa es adicionais, o tempo computacional necess rio para executar cada configura o de par metros variou entre 30 e 50 segundos.

Para melhor analisar os dados da Tabela 6.17, a Figura 6.14, apresenta o comportamento das m dias obtidas para o problema RC207, com o operador UOBX e a pol tica de elitismo DJ. No eixo x mostra-se a varia o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da fun o objetivo. As linhas mostram os diferentes valores analisados para a taxa de muta o.

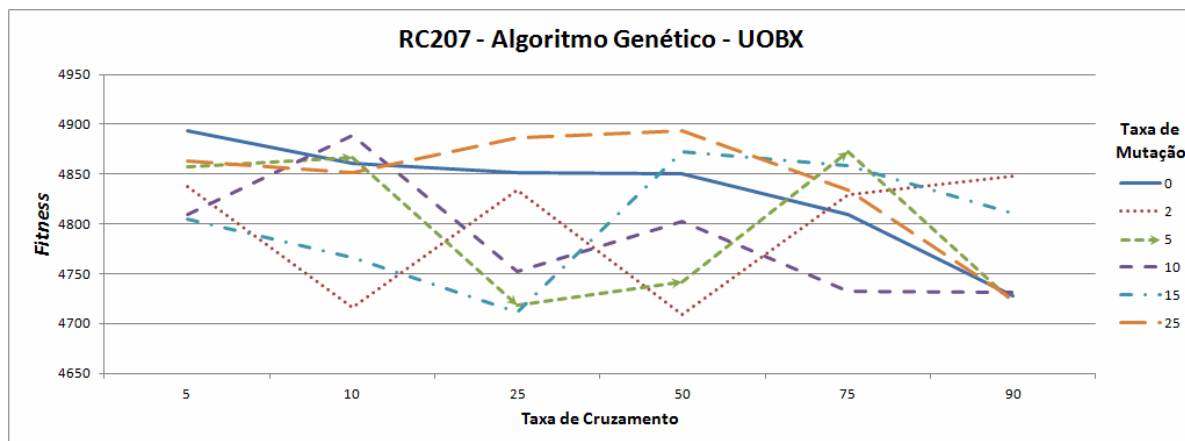


Figura 6.14 – Evolução do Algoritmo Genético com operador UOBX e com a política DJ para o problema RC207

Pode-se observar na Figura 6.14, que apesar da grande variação da qualidade das soluções obtidas, quando a taxa de cruzamento eleva-se para 90, a maior parte dos experimentos obtém soluções de melhor qualidade.

Analisando, em conjunto, as Tabelas 6.15, 6.16 e 6.17, pode-se verificar o aumento da taxa de mutação promove, de modo geral, uma piora nas soluções obtidas. Isso se deve ao processo de busca ser guiado para novas regiões do espaço de busca, mantendo a diversidade da população. Ainda assim, na política de elitismo DJ, quando se cria uma nova população, mantém-se apenas o indivíduo mais apto, facilitando o processo de diversificação da população, se comparado a política de elitismo GE.

A seguir, são analisados os resultados obtidos pela política de elitismo GE aplicada ao problema RC207. A Tabela 6.18, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador OBX e política GE. Os valores destacados na Tabela 6.18 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.18 – AG aplicada ao problema RC207 com operador OBX e com a política GE

Problema: RC207 – GE							
Operador de Cruzamento: OBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5125	5104,4	4363,7	4306	3963,5	3772,3
	σ	356,8	331	347,7	464,3	475,3	373,8
2	FC	5447,6	4903,7	4805,3	4459,1	4307,6	3819,7
	σ	472,9	536,5	365,8	580,7	518,8	581,5
5	FC	5318,3	5167,2	4817,1	4712,5	4665,7	3988,7
	σ	372	433,9	505,1	578,8	520,1	329,3
10	FC	5301,5	5184,8	4818,1	4080,6	3985,4	3775,5
	σ	301,1	445	474,2	473,3	401,5	314,4
15	FC	5442,2	5090,2	4133,7	3911,1	3876,2	3728,4
	σ	364,9	364,3	466,2	542	367,1	539,9
25	FC	5186	4846,4	4604,4	4063,2	3873,5	3711,4
	σ	508,2	481,2	595,3	306,5	424,7	556,2

Na Tabela 6.18, observa-se que todas as solu es de boa qualidade foram obtidas com altas taxas de cruzamento, 75 e 90. Pode-se verificar que o aumento da taxa de cruzamento produz, claramente, uma melhora nas solu es obtidas. Ainda assim, na pol tica de elitismo GE n o existe o conceito de nova popula o, uma vez que um indiv duo mais apto   inserido na popula o somente se este se mostrar mais apto que os demais indiv duos da popula o atual. Nessas condi es, a popula o tende a convergir rapidamente para uma regi o de  timo local, restringindo o espa o de busca. Devido a isso, ao aumentar a taxa de muta o, novas caracter sticas podem ser inseridas na popula o, promovendo melhorias e guiando a busca para novas  reas do espa o de solu es. Como informa es adicionais, o tempo computacional necess rio para executar cada configura o de par metros variou entre 30 e 50 segundos.

A Figura 6.15, apresenta o comportamento das m dias obtidas para o problema RC207, com o operador OBX e a pol tica de elitismo GE. No eixo x mostra-se a varia o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da fun o objetivo. As linhas mostram os diferentes valores analisados para a taxa de muta o.

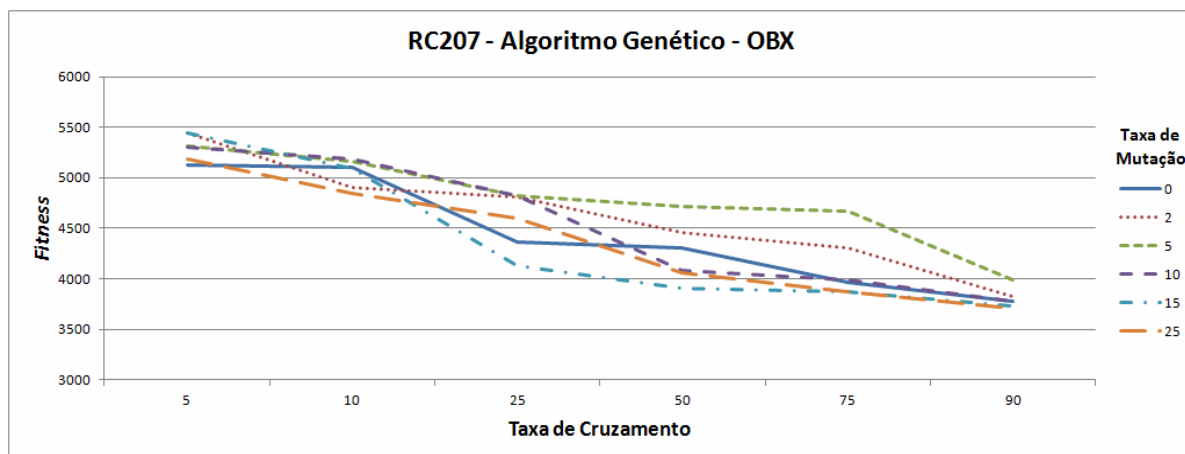


Figura 6.15 – Evolução do Algoritmo Genético com operador OBX e com a política GE para o problema RC207

Na Figura 6.15, pode-se verificar que o aumento da taxa de cruzamento está diretamente ligado a qualidade da solução, característica derivada da política de elitismo GE. Ainda assim, pode-se verificar que as melhores soluções foram obtidas com altas taxas de mutação.

Para analisar o operador PMX juntamente com a política elitista GE, a Tabela 6.19, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador PMX e política GE. Os valores destacados na Tabela 6.17 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.19 – AG aplicada ao problema RC207 com operador PMX e com a política GE

Problema: RC207 – GE							
Operador de Cruzamento: PMX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5075,2	4697,3	4282,9	4087,3	3964,3	3857,5
	σ	489,7	335,8	378	461,7	464,9	418,4
2	FC	4912,5	4831,8	4534	3938,3	3845,8	3674,2
	σ	537,6	370,3	392	543,8	439,2	353,4
5	FC	5065,3	5028,8	4695,5	3926,2	4335	3565,9
	σ	428,5	536,1	385	550,7	381,5	363,7
10	FC	5236,5	5039,2	4898,8	4430,6	4060	3850,4
	σ	405,2	484,4	479,8	473,7	364,9	524,4
15	FC	5218,2	5196,6	4974,2	4782,6	4164,9	3693,1
	σ	351,5	439,2	376,9	444,5	376	410,4
25	FC	5236,6	5171	5092,3	4824,5	4624,8	3570,8
	σ	347,2	356,3	382	406,4	465,3	480,8

Observa-se que o aumento, em conjunto, das taxas de cruzamento e mutação, promovem melhoras nas soluções obtidas. Ao aumentar a taxa de mutação, permite-se que a busca explore novas áreas do espaço de soluções, enquanto que a política de elitismo permite que a busca seja guiada para pontos de ótimos locais. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

A Figura 6.16, apresenta o comportamento das médias obtidas para o problema RC207, com o operador PMX e a política de elitismo GE. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

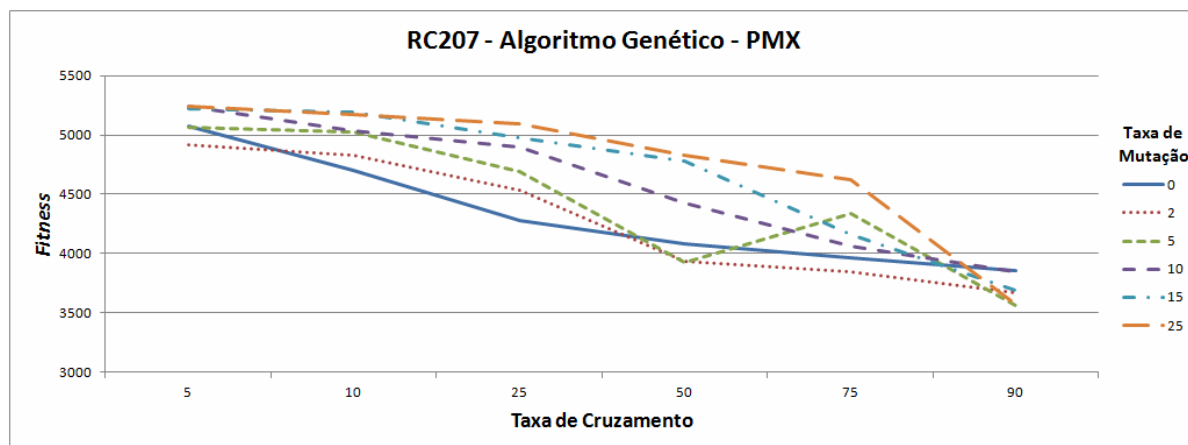


Figura 6.16 – Evolução do Algoritmo Genético com operador PMX e com a política GE para o problema RC207

Observa-se, na Figura 6.16, que o aumento da taxa de cruzamento tende a produzir soluções de boa qualidade, principalmente quando os cruzamentos são realizados em 90% dos casos.

Por fim, para finalizar a análise ao problema RC207, a Tabela 6.20, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador UOBX e política GE. Os valores destacados na Tabela 6.20 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.20 – AG aplicada ao problema RC207 com operador UOBX e com a política GE

Problema: RC207 – GE							
Operador de Cruzamento: UOBX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4506,3	4410,9	4345	4029,4	3950,2	3572,8
	σ	587	487,4	301,7	595,1	304,6	458,4
2	FC	4454,3	4357,2	3955	3890,1	3603,7	3622,7
	σ	505	343,9	541,7	564,2	407,3	463,8
5	FC	4789,6	4478,8	4266,6	4060,6	3834,4	3678,8
	σ	589,4	577,5	550,6	531,9	379,1	584,9
10	FC	4959,8	4728,6	4496,9	4463,6	3837,6	3744,8
	σ	494,6	599,9	369,8	566,9	416	321,8
15	FC	4981,9	4229	4047,9	4025,9	3870,3	3717,6
	σ	525,8	586,3	495,8	305,7	322,4	355,2
25	FC	4978,7	4472,5	4369,4	4358,7	3810	3525,7
	σ	558,6	586,5	435,9	558,7	350,9	379

Pode-se verificar que o aumento das taxas de mutação e cruzamento promovem melhoras nas soluções obtidas. Nesse caso, os melhores resultados foram obtidos com a maior taxa de mutação analisada, uma vez que o operador de mutação permite que a busca seja direcionada a áreas ainda não exploradas do espaço de soluções. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

A Figura 6.17, apresenta o comportamento das médias obtidas para o problema RC207, com o operador UOBX e a política de elitismo GE. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

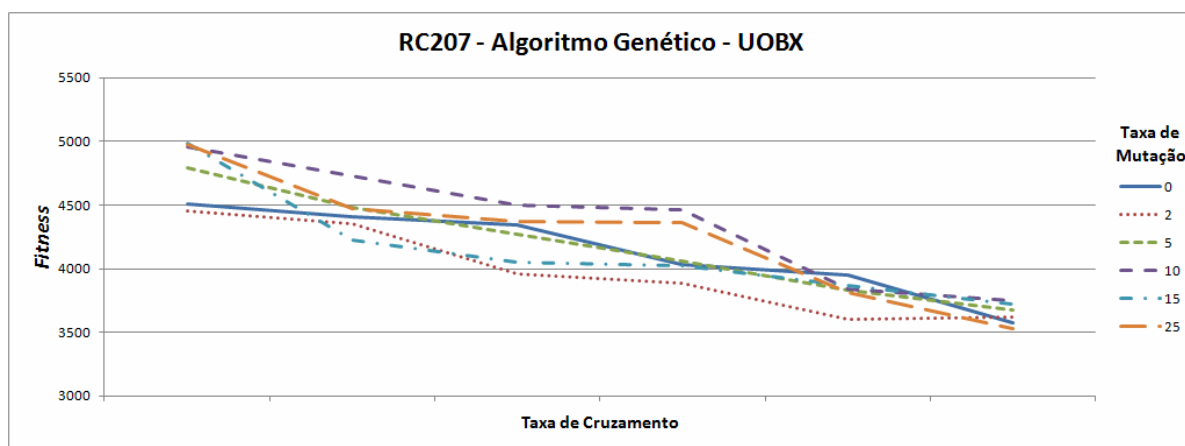


Figura 6.17 – Evolução do Algoritmo Genético com operador UOBX e com a política GE para o problema RC207

Analisando a Figura 6.17, pode-se verificar que o aumento da taxa de cruzamento produz soluções de melhor qualidade. Ainda assim, a elevação da taxa de mutação produz soluções de melhor qualidade somente com altas taxas de cruzamento.

Analisando as Tabelas 6.18, 6.19 e 6.20 pode-se constatar que os tipos de operadores de cruzamento não são relevantes, enquanto aplicados à política de elitismo GE. Ainda assim, pode-se verificar que as melhores soluções são obtidas com altas taxas de cruzamento. Aplicando a política GE, percebe-se que a população tende a convergir rapidamente e que o aumento da taxa de mutação promove uma busca ampla do espaço de busca enquanto as altas taxas de

cruzamento promovem uma intensificação mais habilidosa, nas áreas mais promissoras.

A Figuras 6.18 mostra o percentual de melhora obtido comparando a melhor solução obtida com cada um dos operadores em relação ao indivíduo mais apto da população inicial, *fitness* de 5972,4, para o problema RC207.

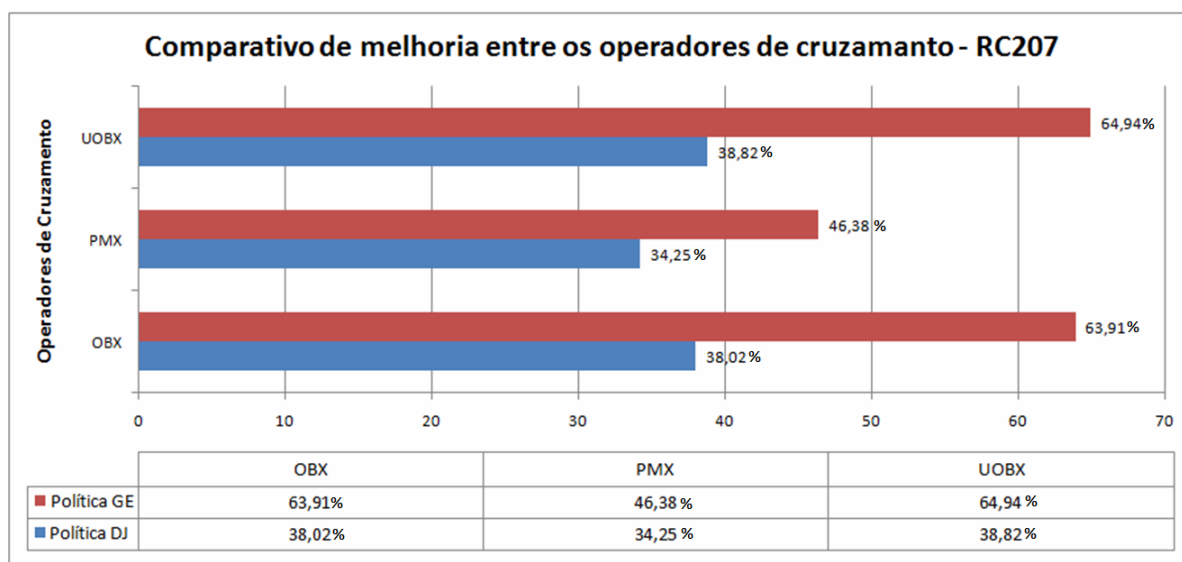


Figura 6.18 – Comparativo de melhora do para o problema RC207

Na Figura 6.18, pode-se verificar que a política elitista GE obteve resultados de boa qualidade, se comparados com a política elitista DJ. Além disso, o operador UOBX obteve resultados de melhor qualidade comparados com os demais operadores. Ainda assim, o operador OBX obteve resultados de melhor qualidade, comparado com o operador PMX.

O problema R205 difere do problema RC207 nas localizações geográficas, onde os clientes são distribuídos aleatoriamente. A Tabela 6.21, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador OBX e política DJ. Os valores destacados na Tabela 6.19 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.21 – AG aplicada ao problema R205 com operador OBX e com a política DJ

Problema: R205 – DJ							
Operador de Cruzamento: OBX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4506,3	4410,9	4345	4029,4	3950,2	3572,8
	σ	587	487,4	301,7	595,1	304,6	458,4
2	FC	4454,3	4357,2	3955	3890,1	3603,7	3622,7
	σ	505	343,9	541,7	564,2	407,3	463,8
5	FC	4789,6	4478,8	4266,6	4060,6	3834,4	3678,8
	σ	589,4	577,5	550,6	531,9	379,1	584,9
10	FC	4959,8	4728,6	4496,9	4463,6	3525,7	3744,8
	σ	494,6	599,9	369,8	566,9	379	321,8
15	FC	4981,9	4229	4047,9	4025,9	3870,3	3717,6
	σ	525,8	586,3	495,8	305,7	322,4	355,2
25	FC	4978,7	4472,5	4369,4	4358,7	3810	3837,6
	σ	558,6	586,5	435,9	558,7	350,9	416

Observa-se na Tabela 6.21, que os melhores resultados foram obtidos com altas taxas de cruzamento e que o aumento excessivo da taxa de mutação impede que a busca intensifique em região promissora do espaço de soluções, forçando a busca para novas áreas de busca. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

A Figura 6.19, apresenta o comportamento das médias obtidas para o problema R205, com o operador OBX e a política de elitismo DJ. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

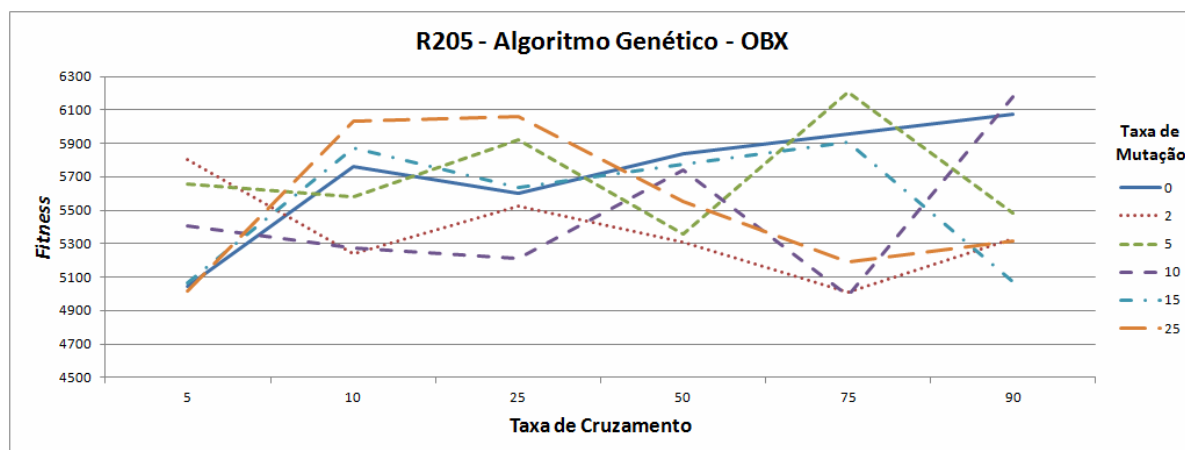


Figura 6.19 – Evolução do Algoritmo Genético com operador OBX e com a política DJ para o problema R206

Observando a Figura 6.19, constata-se uma grande variação nos valores do *Fitness* à medida que se varia a taxa de cruzamento. Não pode ser identificado o mesmo padrão encontrado na Figura 6.16, onde o aumento da taxa de cruzamento estava diretamente ligado a melhora da Função Custo.

Analisando o operador de cruzamento PMX, a Tabela 6.22, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador PMX e política DJ. Os valores destacados na Tabela 6.22 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.22 – AG aplicada ao problema R205 com operador PMX e com a política DJ

Problema: R205 – DJ							
Operador de Cruzamento: PMX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4957,1	5068,9	5419	5617,1	5527	5688
	σ	1049,2	1175,1	1026	1025,3	919,6	997,2
2	FC	4994	5475,1	5292,2	5096,1	5225,1	5424,5
	σ	1237,7	1053,3	984,9	1077,2	988,2	909,4
5	FC	5316,5	4907,6	5214,5	5316,5	4946,8	4956,7
	σ	1150	1056,8	1125,3	1067,5	1032,7	1149,2
10	FC	5618,5	5021,4	5015,2	5685,8	5119,6	5604,8
	σ	1232	1159,5	1003	1012,7	1072,2	990,9
15	FC	5623,7	5297,5	5644,8	5508,7	4912,5	5610,8
	σ	1021,5	1056,3	1037,2	1127,9	1179,5	1099,5
25	FC	5130,7	5314,4	5269	5645	5478,2	4985
	σ	900,9	1223,7	1173	1040,2	1004,5	987,5

Pode-se analisar, na Tabela 6.22, que resultados de boa qualidade foram obtidos utilizando baixas taxas de mutação, desse modo, a busca não foi guiada para diversas regiões do espaço de busca, permitindo que uma região promissora seja explorada. Ainda assim, pode-se verificar que mesmo utilizando baixas taxas de cruzamento, bons resultados foram obtidos. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

Para facilitar a visualização dos resultados mostrados na Tabela 6.22, a Figura 6.20, apresenta o comportamento das médias obtidas para o problema R205, com o operador PMX e a política de elitismo DJ. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

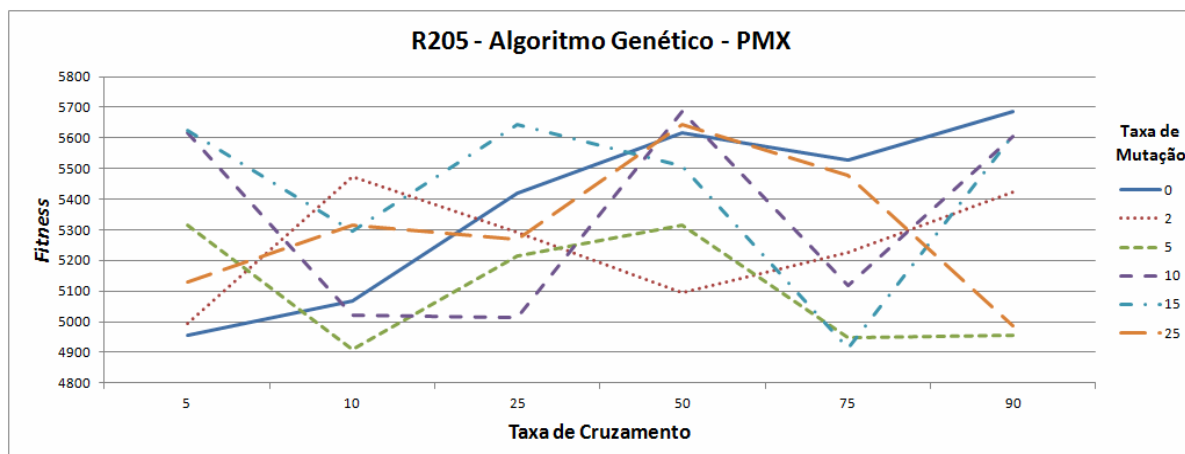


Figura 6.20 – Evolução do Algoritmo Genético com operador PMX e com a política DJ para o problema R205

Observa-se, na Figura 6.20, que soluções de boa qualidade foram obtidas com diversos conjuntos de parâmetros. Essa característica está ligada a política de elitismo DJ, a qual recria todos os indivíduos a medida que uma nova população é gerada.

Para analisar o operador UOBX, a Tabela 6.23, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador UOBX e política DJ. Os valores destacados na Tabela 6.23 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.23 – AG aplicada ao problema R205 com operador UOBX e com a política DJ

Problema: R205 – DJ		Operador de Cruzamento: UOBX					
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5081,7	4901,3	5041,6	4731,6	4954,4	4704,9
	σ	989,8	943,2	806,1	1024,8	752	986,6
2	FC	4941,5	4964,4	5015,3	5020	4824,2	4779,9
	σ	797,9	900,8	1045,8	850,5	862,6	778,9
5	FC	4866,8	4823,5	4864,3	5029,7	5010,1	4713,7
	σ	983,8	762,8	927	870,6	843,6	925
10	FC	4829,3	4834,7	4757,4	4944,9	4750,3	5005,5
	σ	840,2	939,7	786,2	826,8	1021,5	1034,7
15	FC	4984,6	4994,2	4875	4720,7	4705,4	4767,9
	σ	799,6	752,2	1006,3	775,1	788,4	910,6
25	FC	4775,2	4973,1	4775,3	4914,6	5085,1	4988,8
	σ	774,3	775,7	962,2	918,4	834,8	872

Pode-se verificar na Tabela 6.23, que a elevação da taxa de mutação não produziu soluções de boa qualidade. Além disso, as melhores soluções foram obtidas com altas taxas de cruzamento. Essa elevação da taxa de cruzamento permite que a busca seja intensificada em uma área promissora. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

A Figura 6.21, apresenta o comportamento das médias obtidas para o problema R205, com o operador OUBX e a política de elitismo DJ. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

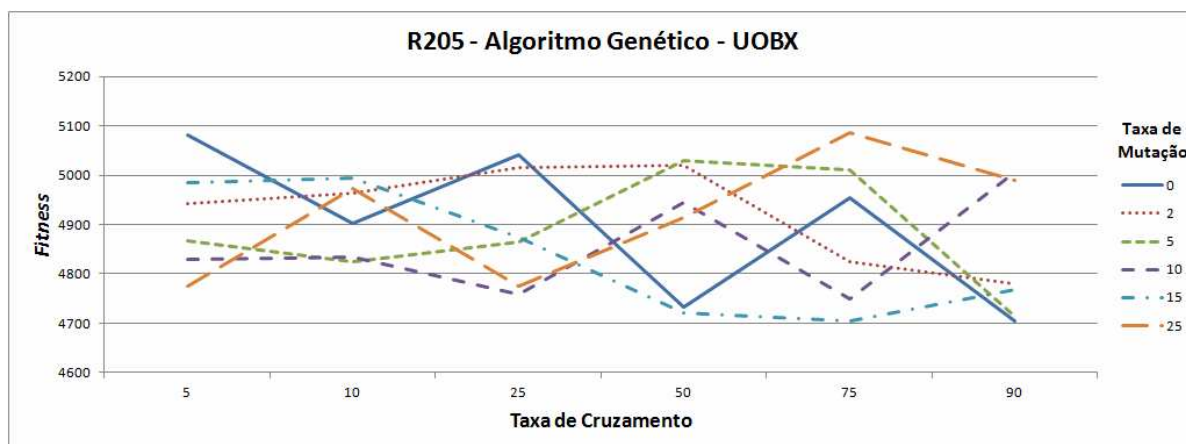


Figura 6.21 – Evolução do Algoritmo Genético com operador UOBX e com a política DJ para o problema R205

Pode-se observar que a Figura 6.21 não apresenta um padrão de melhoria no valor do *Fitness*. Comparando as Figuras 6.20 e 6.21, pode-se verificar que as características do problema R205 dificultaram a exploração do espaço de soluções.

Analisando as Tabelas 6.21, 6.22 e 6.23, pode-se verificar que a política de elitismo DJ não promove uma busca consistente no espaço de soluções, se comparado com a política GE.

Analisando a aplicação da política GE ao problema R205, a Tabela 6.24, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador OBX e política GE. Os valores destacados na Tabela

6.24 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.24 – AG aplicada ao problema R205 com operador OBX e com a política GE

Problema: R205 – GE							
Operador de Cruzamento: OBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4939,8	4004,9	3873,6	3853,7	3726,1	3876,5
	σ	562,8	503,4	501,8	495,9	573,9	561,9
2	FC	4857,2	4833,6	4785,9	4359,9	3857,1	3454,9
	σ	546	486,4	557,8	453,6	425,6	453,3
5	FC	4918,9	4797,8	4592,5	4333	3979,3	3926,2
	σ	468,4	550,4	538,1	483,4	556,6	567,8
10	FC	5055,1	4886	4357,3	4333,2	4105	3713,8
	σ	499,7	599,4	551	492,2	563,3	589,9
15	FC	5267,6	4844,9	4813,3	4282,9	3614,5	3284,4
	σ	527,7	590,8	456,8	489	532,5	281,7
25	FC	5153,1	4606,7	4366	4333,3	3899,4	3212,5
	σ	442,8	587,3	520,5	403	430,5	369,7

Pode-se verificar que as melhores solu es foram obtidas com altas taxas de cruzamento e muta o. A pol tica elitista GE direciona a busca para  reas espec ficas do espa o de solu es, permitindo que o operador de cruzamento possa intensificar a busca enquanto que a taxa de muta o permite que a busca seja guiada para outras  reas do espa o de solu es. Como informa es adicionais, o tempo computacional necess rio para executar cada configura o de par metros variou entre 30 e 50 segundos.

A Figura 6.22, apresenta o comportamento das m dias obtidas para o problema R205, com o operador OBX e a pol tica de elitismo GE. No eixo x mostra-se a varia o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da fun o objetivo. As linhas mostram os diferentes valores analisados para a taxa de muta o.

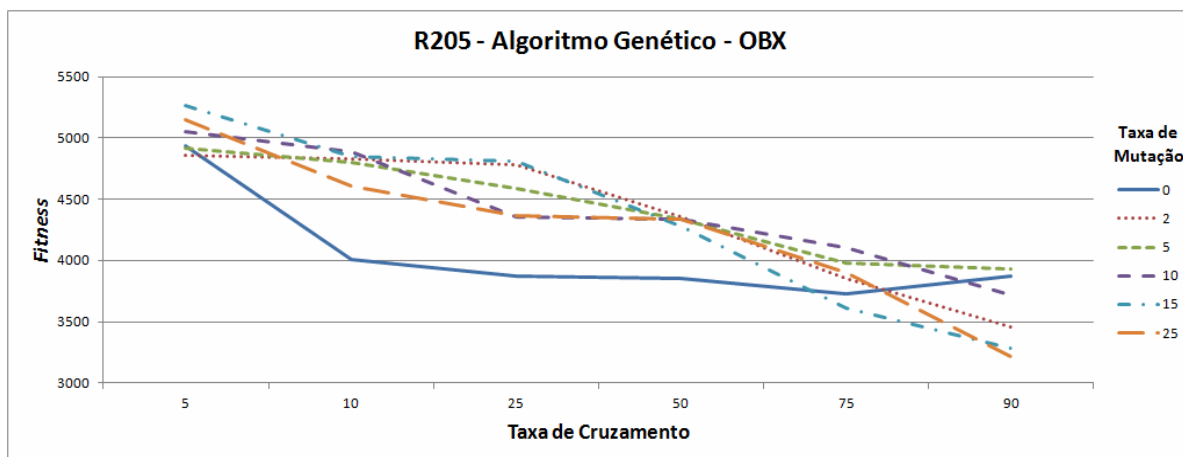


Figura 6.22 – Evolução do Algoritmo Genético com operador OBX e com a política GE para o problema R205

Pode-se verificar claramente que na aplicação da política de elitismo GE, o aumento da taxa de cruzamento está diretamente ligado a qualidade das soluções. Ainda assim, faz-se necessárias elevadas taxas de mutação para poder guiar a busca para novas áreas do espaço de soluções.

Para analisar o operador PMX, aplicado a política elitista GE, a Tabela 6.25, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador PMX e política GE. Os valores destacados na Tabela 6.25 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.25 – AG aplicada ao problema R205 com operador PMX e com a política GE

Problema: R205 – GE							
Operador de Cruzamento: PMX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5129,7	4016,2	3884,5	3600,5	3488,7	3476
	σ	430,7	390,1	435	440,2	409,3	363,6
2	FC	4726,2	4630,4	4349,7	3581	3346	3249
	σ	352,2	367	434	383,4	462,1	452
5	FC	5073,9	4890,8	4542,4	4202,4	3743,7	3620,5
	σ	464,8	473,4	450,7	479,6	402,5	381,2
10	FC	4909,1	4273,3	4082,5	4079,1	3649,7	3196,9
	σ	482,9	399,2	413	374,5	450,4	403,2
15	FC	5068,5	4717,8	4170,8	3961,8	3900	3876,3
	σ	352,8	430	361,4	375,5	460	372,3
25	FC	5034,6	4189,4	4105,3	4021,2	3766,6	3116,1
	σ	465,6	400,5	389,9	379,5	495,7	396,5

Na Tabela 6.25, observa-se que as solu es de melhor qualidade foram obtidas com altas taxas de cruzamento. Ao aumentar a taxa de muta o percebe-se uma melhora nos resultados obtidos, contudo, resultados de boa qualidade tamb m foram obtidos com taxas menores. Isso se deve ao operador de cruzamento, que auxilia na manuten o da diversidade da popula o de forma mais eficiente que os outros operadores analisados. Como informa es adicionais, o tempo computacional necess rio para executar cada configura o de par metros variou entre 30 e 50 segundos.

Para melhor analisar os resultados obtidos na Tabela 6.25, a Figura 6.23, apresenta o comportamento das m dias obtidas para o problema R205, com o operador PMX e a pol tica de elitismo GE. No eixo x mostra-se a varia o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da fun o objetivo. As linhas mostram os diferentes valores analisados para a taxa de muta o.

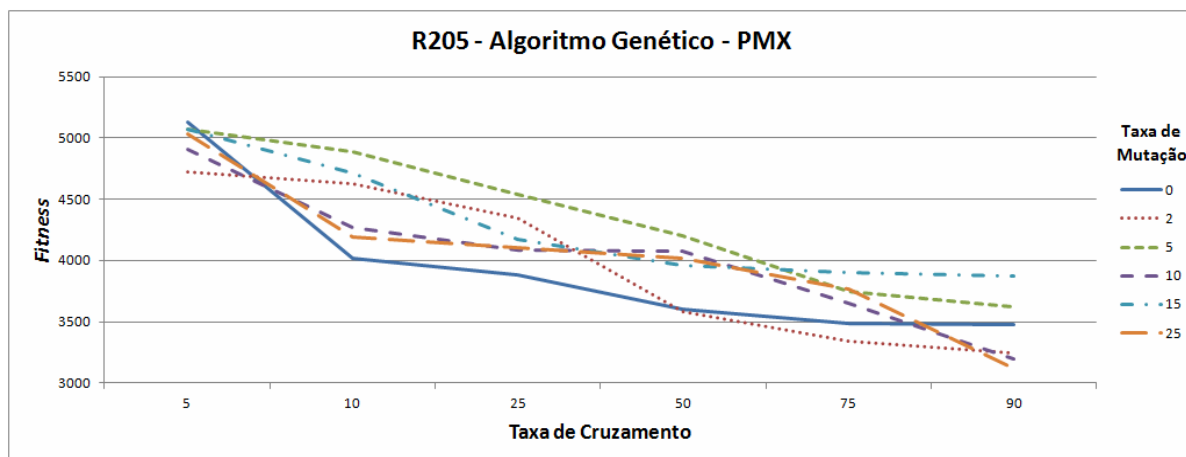


Figura 6.23 – Evolução do Algoritmo Genético com operador PMX e com a política GE para o problema R205

Pode-se verificar, novamente, que o aumento da taxa de cruzamento promove, de modo geral, uma melhora nos resultados obtidos. Além disso, observa-se que baixas taxas de mutação obtiveram soluções de boa qualidade, como é o caso de 2, 10 e 25.

Por fim, analisando o operador de cruzamento UOBX, a Tabela 6.26, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador UOBX e política GE. Os valores destacados na Tabela 6.26 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.26 – AG aplicada ao problema R205 com operador UOBX e com a política GE

Problema: R205 – GE							
Operador de Cruzamento: UOBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4585,5	4400,1	3794,9	3467,4	3269,3	3166,8
	σ	350,3	410,7	440,8	406,6	346	335,3
2	FC	4260,9	4094,2	4053,2	4051,1	3688,8	3549
	σ	365,7	394,9	317,5	333,1	380,9	404,7
5	FC	4780,8	4568,5	4567,9	4057,5	3590,5	2972,9
	σ	313,2	446,9	407,7	390,5	408,9	342,7
10	FC	4608,6	4397,2	4310,4	4154,6	3289,9	3080,2
	σ	436,2	398,4	380,7	422,1	439	413,6
15	FC	4467,4	4450	3983,8	3611,2	3334,8	3234,5
	σ	421,3	335	400,3	351,9	388,8	389
25	FC	4413,9	3401,1	3194,4	3118,4	3089,6	2974,8
	σ	433,1	396,1	399,9	338,3	354,2	328,2

Na Tabela 6.26, pode-se observar que soluç es de boa qualidade foram obtidas com altas taxa de cruzamento e mutaç o. Isso se deve a caracter stica do operador de cruzamento UOBX, o qual direciona a busca para  reas espec ficas do espaço de soluç es, necessitando que o operador de mutaç o diversifique as soluç es, guiando a busca para novas  reas. Como informaç es adicionais, o tempo computacional necess rio para executar cada configuraç o de par metros variou entre 30 e 50 segundos.

Para melhor analisar os resultados obtidos na Tabela 6.26, a Figura 6.24, apresenta o comportamento das m dias obtidas para o problema R205, com o operador UOBX e a pol tica de elitismo GE. No eixo x mostra-se a variaç o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da funç o objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutaç o.

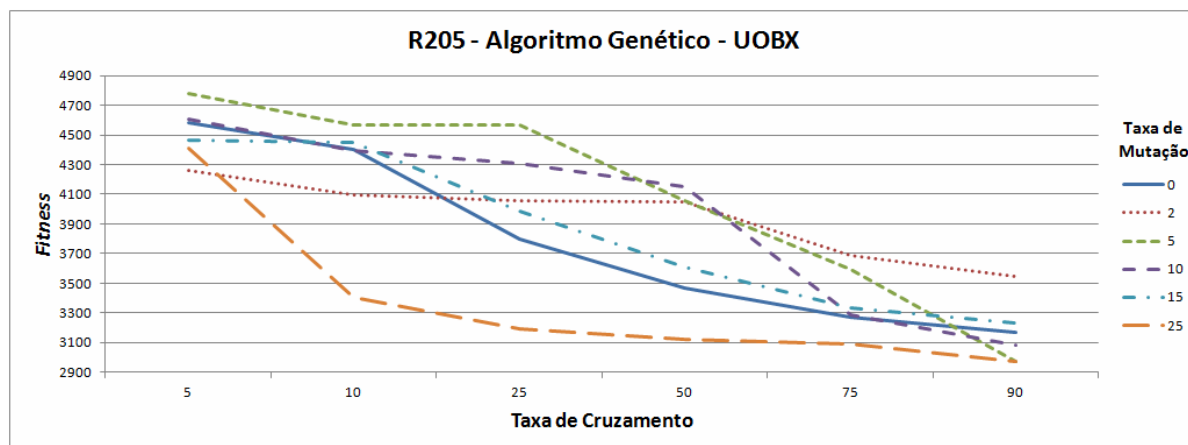


Figura 6.24 – Evolução do Algoritmo Genético com operador UOBX e com a política GE para o problema R205

Na Figura 6.24, observa-se claramente que se faz necessário uma taxa de mutação elevada para que sejam obtidas soluções de boa qualidade. Ainda assim, percebe-se que o aumento da taxa de cruzamento está diretamente ligado a melhora das soluções.

Analisando as Tabelas 6.24, 6.25 e 6.26, pode-se verificar que, apesar do operador UOBX ter a característica de diminuir a diversidade da população rapidamente, quando aliado a uma política que o permita explorar com maior habilidade regiões promissoras do espaço de busca, soluções de melhor qualidade são geradas.

A Figura 6.25 mostra o percentual de melhora obtido comparando a melhor solução obtida com cada um dos operadores em relação ao indivíduo mais apto da população inicial, *fitness* de 6103,7, para o problema R205. As melhores soluções obtidas pelo operador OBX foram de 3756,82 e 2843,10 para as políticas DJ e GE, respectivamente. O operador PMX obteve as soluções 3842,89 e 2719,80 para as políticas DJ e GE, respectivamente, e o operador UOBX obteve as soluções 3870,35 e 2630,08 para as políticas DJ e GE, respectivamente.

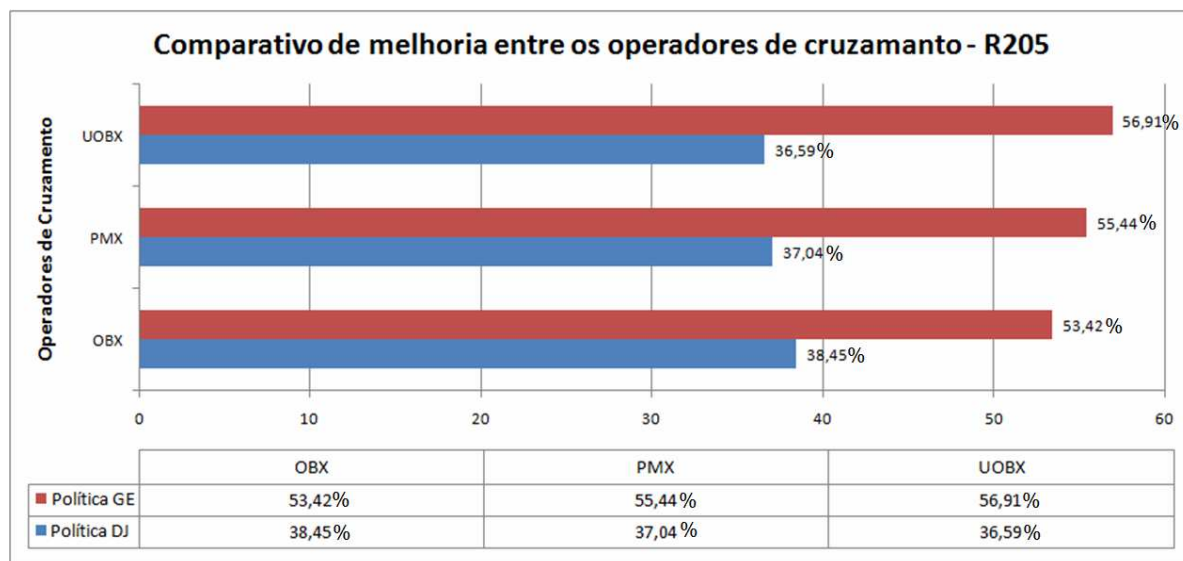


Figura 6.25 – Comparativo de melhora do para o problema R205

Na Figura 6.25 pode-se verificar que a política elitista GE obteve resultados de melhor qualidade, se comparados com a política elitista DJ. Além disso, o operador UOBX obteve resultados de melhor qualidade comparados com os demais operadores, quando aplicado a política de elitismo GE. Isso ocorre devido à política de elitismo GE ter a característica de intensificar a busca em uma região promissora. Desse modo, o operador UOBX pode intensificar a busca nessas áreas, enquanto que o operador de mutação realiza a diversificação, sempre buscando áreas de busca promissoras.

A seguir são apresentados os experimentos para o problema R206. Nesse problema as janelas de tempo são flexíveis, gerando um grande número de soluções possíveis e, por conseguinte, exigindo maior eficiência dos algoritmos.

A Tabela 6.27, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador OBX e política DJ. Os valores destacados na Tabela 6.27 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.27 – AG aplicada ao problema R206 com operador OBX e com a política DJ

Problema: R206 – DJ							
Operador de Cruzamento: OBX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5243	4560,5	4811,5	5560	4796,8	4475,6
	σ	1023,1	999,2	987,6	908,7	960,3	1071,7
2	FC	4758,8	4748,3	5184,1	5038,8	4364,7	4470,5
	σ	954,7	907	947,4	1004,3	1041,4	1068,7
5	FC	4426,1	4733,6	4903,7	4716,8	5009,8	5144,2
	σ	1039,9	1013,2	1015,8	905,3	1063	904,3
10	FC	5063,1	4652,2	4919,4	5078,5	4433,3	4973,8
	σ	932,1	972,2	1052,2	1065,3	928,1	909,4
15	FC	4672,9	5153,4	5502,2	5150,1	4836,5	4622,6
	σ	1008,1	932,7	1006,7	1031,5	977,4	974,7
25	FC	4797,2	4922,3	5397,6	5007,7	5051,3	4504,3
	σ	942,2	1000,4	1094,1	988,1	911,7	932,6

Observa-se que os melhores resultados foram obtidos utilizando altas taxas de cruzamento com baixas taxas de mutação. Pode-se observar que a mutação promove uma diversificação nas soluções obtidas, levando a busca para novas regiões do espaço, porém, o operador de cruzamento aliado a política elitista DJ não realiza o processo de intensificação de forma eficiente, não obtendo resultados de tão boa qualidade. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

Para facilitar a análise dos resultados obtidos na Tabela 6.27, a Figura 6.26, apresenta o comportamento das médias obtidas para o problema R206, com o operador OBX e a política de elitismo DJ. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

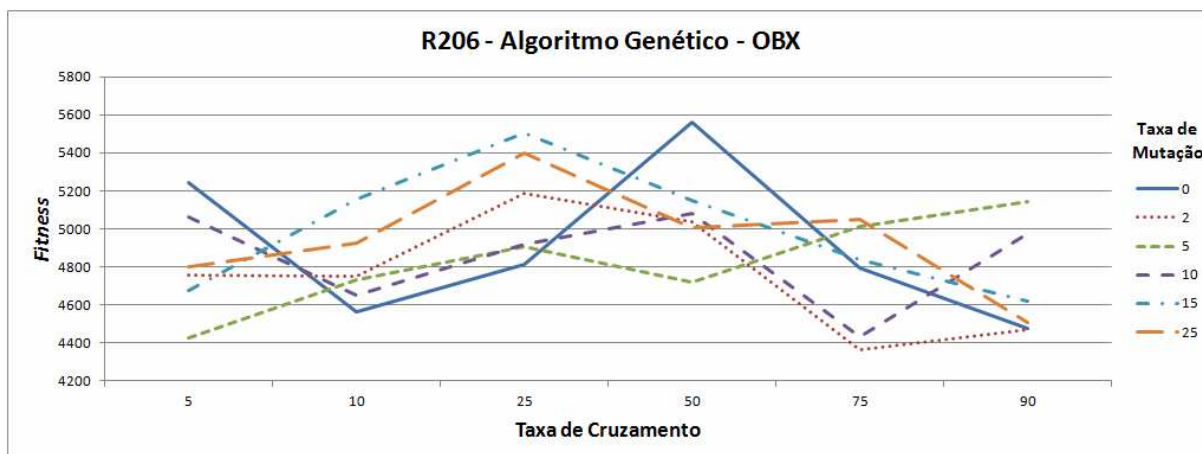


Figura 6.26 – Evolução do Algoritmo Genético com operador OBX e com a política DJ para o problema R206

Pode-se analisar que o aumento da taxa de cruzamento promove melhorias nas soluções obtidas, contudo observa-se que as características do problema dificultam a exploração do espaço de soluções.

Analisando o operador PMX para o problema, A Tabela 6.28, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador PMX e política DJ. Os valores destacados na Tabela 6.28 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.28 – AG aplicada ao problema R206 com operador PMX e com a política DJ

Problema: R206 – DJ							
Operador de Cruzamento: PMX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4996,7	4480,2	5041,6	5097,5	5081,3	4837,6
	σ	814,2	714,2	792,2	748,7	768,9	775,2
2	FC	4643,9	4641,2	4546	4499,9	4794,5	5113,5
	σ	906,5	778,9	817,4	700,4	763,2	708,9
5	FC	4754	4356,5	5164,7	4590,1	5116,8	4806,6
	σ	967,5	721,7	889,1	821,4	769,9	750,7
10	FC	4924,8	4981,1	5078,6	4323,4	5174,2	4499,1
	σ	928,5	832,7	747,2	773,9	731,9	756,5
15	FC	4622,1	4916,6	5096,8	4912,9	4497	4473,1
	σ	708,6	891,5	790,7	912,3	981,6	740
25	FC	5138,6	4802,1	4899,4	5147,8	4756,7	4785
	σ	712,8	805,1	749,2	719,6	862,9	817,2

Pode-se observar que foram obtidos resultados de boa qualidade utilizando baixas taxas de mutação, desse modo, a busca não foi guiada para diversas regiões do espaço de busca, permitindo que uma região promissora seja explorada. Ainda assim, pode-se verificar que, de modo geral, mesmo utilizando baixas taxas de cruzamento, bons resultados foram obtidos. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

A Figura 6.27, apresenta o comportamento das médias obtidas para o problema R206, com o operador PMX e a política de elitismo DJ. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

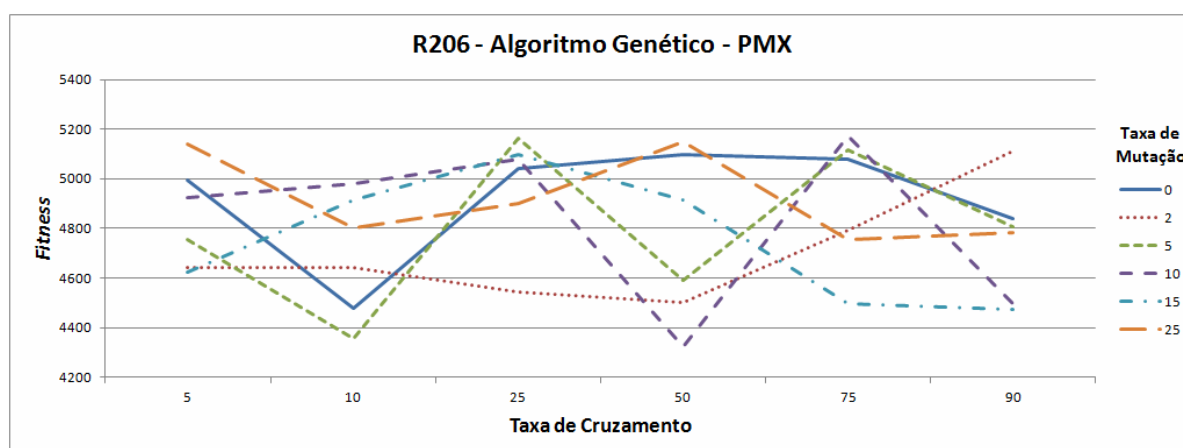


Figura 6.27 – Evolução do Algoritmo Genético com operador PMX e com a política DJ para o problema R206

Pode-se verificar uma grande variação do valor Função Custo. O simples aumento da taxa de cruzamento não prove melhorias. Esse comportamento se deve ao operador de cruzamento que não se mostrar tão eficiente frente à complexidade do problema.

A Tabela 6.29, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador UOBX e política DJ. Os valores destacados na Tabela 6.29 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.29 – AG aplicada ao problema R206 com operador UOBX e com a política DJ

Problema: R206 – DJ							
Operador de Cruzamento: UOBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4570,1	4357,2	4529,1	4559,1	4358,1	4300,8
	σ	658,9	702,2	893,8	719,8	696,8	786,5
2	FC	4542,5	4328	4395,7	4392,4	4532,4	4532,8
	σ	835,8	720,8	789,9	754,6	916,2	771,1
5	FC	4499,5	4439,5	4499,4	4395	4327,2	4369,9
	σ	779,7	705,6	861	945,5	696,4	859,2
10	FC	4498	4402,2	4389,2	4449,3	4348,8	4540,8
	σ	655,4	813,9	925,1	922	829,8	750,8
15	FC	4455,7	4346,6	4461,8	4310,3	4417,8	4311,5
	σ	694,7	662,2	731,3	759,5	709,8	839,9
25	FC	4538,2	4333,1	4506,7	4391,7	4453,1	4303,4
	σ	768,9	937,6	732,3	858,1	891	862,8

Na Tabela 6.29, pode-se verificar que resultados de boa qualidade foram obtidos, em grande parte, com altas taxas de cruzamento e com baixas taxas de muta o. Ao aumentar muito a taxa de muta o, o operador de cruzamento UOBX, aliado a pol tica elitista DJ, n o explora, de forma eficiente,  reas de busca promissoras. Como informa es adicionais, o tempo computacional necess rio para executar cada configura o de par metros variou entre 30 e 50 segundos.

A Figura 6.28, apresenta o comportamento das m dias obtidas para o problema R206, com o operador UOBX e a pol tica de elitismo DJ. No eixo x mostra-se a varia o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da fun o objetivo. As linhas mostram os diferentes valores analisados para a taxa de muta o.

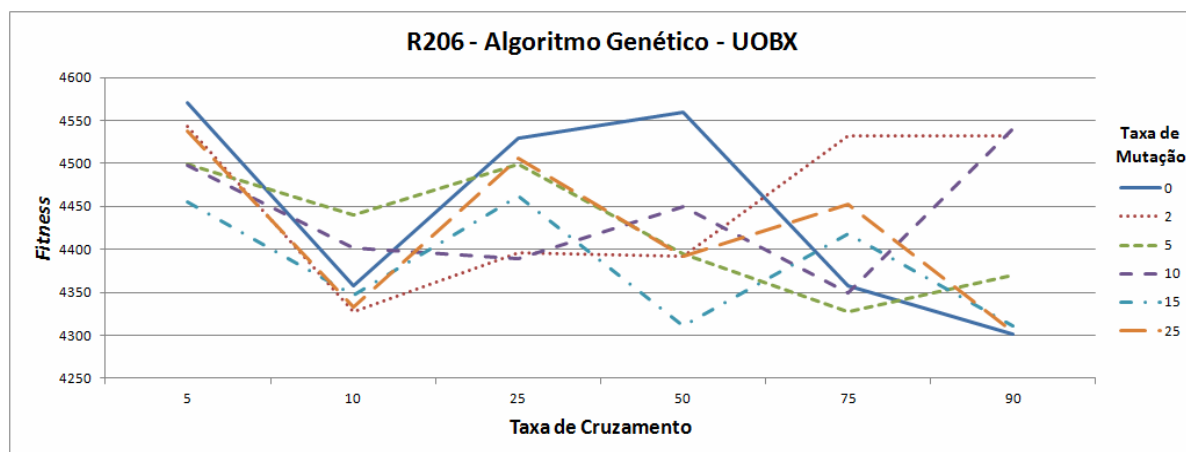


Figura 6.28 – Evolução do Algoritmo Genético com operador UOBX e com a política DJ para o problema R206

Pode-se verificar que aplicando o operador de cruzamento UOBX à política DJ, para o problema R206, não são obtidas soluções de tão boa qualidade. Observa-se a busca não explora de forma eficiente o espaço de soluções.

A política de elitismo DJ promove uma ampla busca no espaço de soluções, permitindo que uma área maior de soluções seja considerada. Apesar disso, ela não direciona, habilmente, a busca para regiões promissoras, dificultando na obtenção de soluções de boa qualidade.

Em alternativa a política de elitismo DJ, aplica-se a política de elitismo GE no problema R206. A Tabela 6.30, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador OBX e política GE. Os valores destacados na Tabela 6.30 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.30 – AG aplicada ao problema R206 com operador OBX e com a política GE

Problema: R206 – GE							
Operador de Cruzamento: OBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	5386,4	4974,9	4256,6	3963,1	3465,7	3228,3
	σ	505,9	504,1	546,8	450,4	455,5	417
2	FC	4748,8	4581,9	3834,2	3729,1	3716,8	3475,4
	σ	419	579,6	596,3	470,2	516,8	553
5	FC	5440,6	4670,7	4448,6	3681,3	3630,4	3490,7
	σ	500,6	596,6	508,6	454,4	482,1	567,8
10	FC	4979,8	4909,8	4736,9	4534,8	4393,5	3886,2
	σ	565,3	537,1	438,5	451,6	491,8	443,5
15	FC	5122	5110,2	4818,8	3899,5	3397,3	3273,2
	σ	446	430,7	595,7	595,5	518,1	567,3
25	FC	5111,5	4995,7	4517,7	3724,3	3615,1	3210
	σ	537,8	435,5	489,1	484,4	538,7	409

Na Tabela 6.30, pode-se verificar que as melhores solu es foram obtidas com altas taxas de cruzamento e muta o. A pol tica elitista GE direciona a busca para  reas espec ficas do espa o de solu es, permitindo que o operador de cruzamento possa intensificar a busca enquanto que a taxa de muta o permite que a busca seja guiada para outras  reas do espa o de solu es. Como informa es adicionais, o tempo computacional necess rio para executar cada configura o de par metros variou entre 30 e 50 segundos.

Para facilitar a an lise dos dados obtidos para a Tabela 6.30, a Figura 6.29, apresenta o comportamento das m dias obtidas para o problema R206, com o operador OBX e a pol tica de elitismo GE. No eixo x mostra-se a varia o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da fun o objetivo. As linhas mostram os diferentes valores analisados para a taxa de muta o.

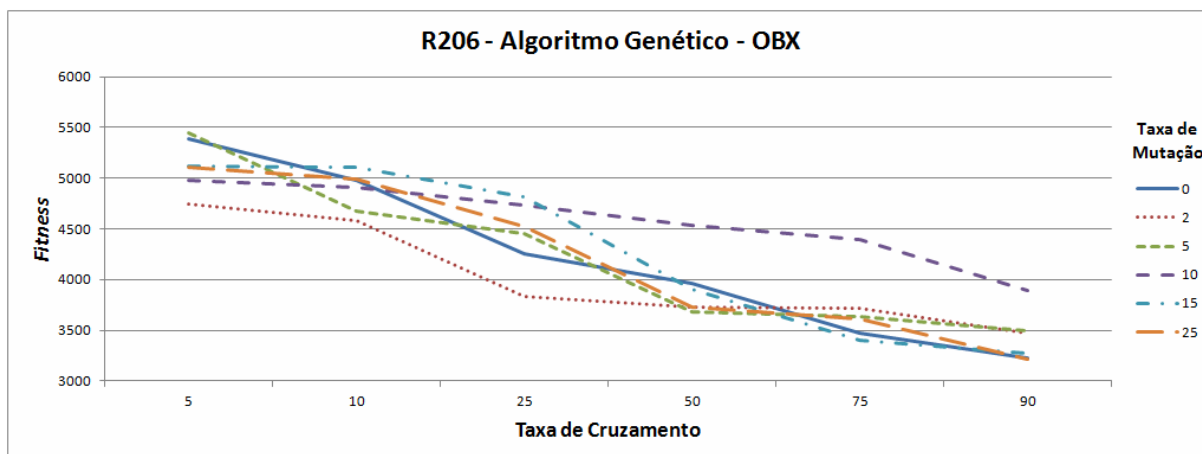


Figura 6.29 – Evolução do Algoritmo Genético com operador OBX e com a política GE para o problema R206

Observa-se na Figura 6.29, que ao aplicar o operador OBX à política de elitismo GE, a busca se torna mais consistente. Pode-se verificar que o aumento da taxa de cruzamento está diretamente ligado a melhora das soluções obtidas.

A seguir, aplica-se o operador PMX à política elitista GE. A Tabela 6.31, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador PMX e política GE. Os valores destacados na Tabela 6.31 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.31 – AG aplicada ao problema R206 com operador PMX e com a política GE

Problema: R206 – GE							
Operador de Cruzamento: PMX							
Taxa de Mutação		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4689,2	4430,6	4374,9	4252,4	4251,1	3053,8
	σ	538	548,5	378,8	502,7	500,5	391,4
2	FC	4848,8	4757,6	4728,6	3882,4	3303,6	3037,2
	σ	463,5	532,4	405,7	400,6	394,8	477,2
5	FC	4696,8	4107,6	4035,3	3997,7	3826,5	3708,4
	σ	379,9	505	396,7	368,5	498,6	518,8
10	FC	4827,2	4399,6	4253,9	4147,9	3994,8	3156,1
	σ	514	363,7	371	521,4	432,6	433,9
15	FC	4813,7	4523,2	4516,7	4280,5	4112,2	3956,1
	σ	457,7	453,9	466,3	432,4	353,6	415,4
25	FC	4880,9	4844,6	4737,6	3944,1	3779,6	3484,9
	σ	534,6	404	475,1	533,5	419,4	488,2

Observa-se na Tabela 6.31, que as soluções de melhor qualidade foram obtidas com altas taxas de cruzamento. Ao aumentar a taxa de mutação percebe-se uma melhora nos resultados obtidos, contudo, resultados de boa qualidade também foram obtidos com taxas menores. Isso se deve as características do operador de cruzamento PMX, que aliado a política elitista GE, não dependem exclusivamente da taxa de mutação para diversificar a busca e intensificá-la em áreas prósperas. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 30 e 50 segundos.

Para facilitar a análise dos dados obtidos para a Tabela 6.31, a Figura 6.30, apresenta o comportamento das médias obtidas para o problema R206, com o operador PMX e a política de elitismo GE. No eixo x mostra-se a variação da taxa de cruzamento enquanto que no eixo y mostra-se o valor da função objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutação.

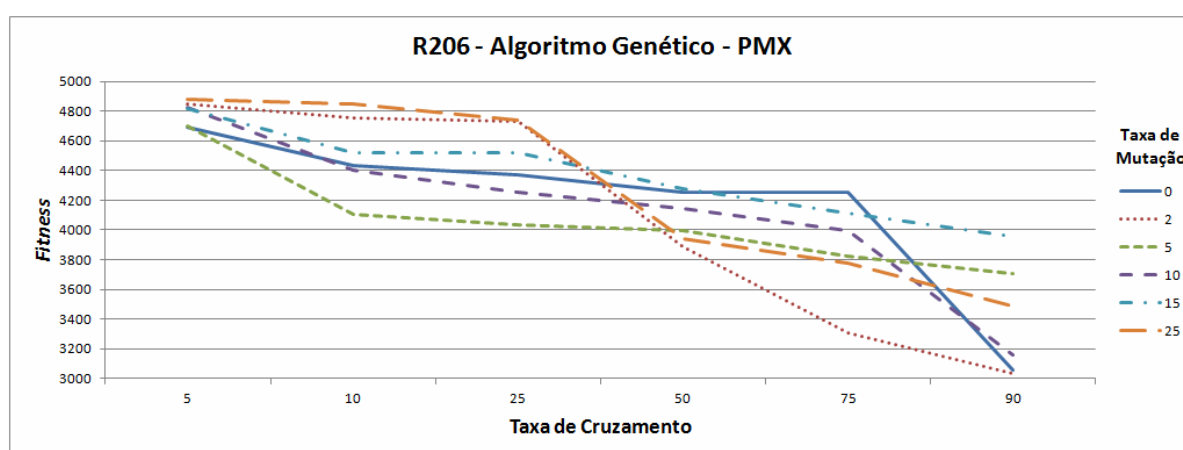


Figura 6.30 – Evolução do Algoritmo Genético com operador PMX e com a política GE para o problema R206

Podem-se verificar, analisando a Figura 6.30, que melhoras consideráveis foram obtidas quando aplicada a maior taxa de cruzamento. Conforme comentado anteriormente, o operador de cruzamento PMX não necessita exclusivamente da taxa de mutação para diversificar a população e explorar novas regiões do espaço de soluções. Devido a isso, observam-se soluções de boa qualidade foram obtidas, também, com baixas taxas de mutação.

Por fim, a Tabela 6.32, mostra a relação das taxas de cruzamento, nas colunas, e de mutação, nas linhas. As células mostram os valores médios obtidos para a Função Custo e seu respectivo desvio padrão, com o operador UBOX e

política GE. Os valores destacados na Tabela 6.32 mostram os melhores resultados obtidos, considerando todos os conjuntos de parâmetros.

Tabela 6.32 – AG aplicada ao problema R206 com operador UOBX e com a política GE

Problema: R206 – GE							
Operador de Cruzamento: UOBX							
Taxa de Mutaç�o		Taxa de Cruzamento					
		5	10	25	50	75	90
0	FC	4483,5	3850,8	3822,2	3486,5	3376,1	3021
	σ	471,3	451,7	407,2	380	321	332,8
2	FC	4470,7	4345,9	4285	3820	3769	3528,7
	σ	386,7	482	388,6	488,8	392,8	396,3
5	FC	4157,4	4138	4086,3	3677,6	3466,3	3258,9
	σ	464,3	362	352,5	379,3	387	369,4
10	FC	4390,1	4190,4	3587,4	3581,3	3231,3	3153,4
	σ	350,9	422,9	431,3	376,7	391,1	447
15	FC	4453,4	4352,6	3923,1	3874,3	3150,4	3078,1
	σ	496,9	397,2	429,5	374	327,5	499,4
25	FC	4276,6	4222,1	4149,6	3495,4	3109,3	3009,9
	σ	300,4	319	434,7	387,9	398,6	390,2

Na Tabela 6.32, observa-se que soluç es de boa qualidade foram obtidas, principalmente, com altas taxa de cruzamento e mutaç o. Isso se deve a caracter stica do operador de cruzamento UOBX, o qual direciona a busca para  reas espec ficas do espaço de soluç es, necessitando que o operador de mutaç o diversifique as soluç es, guiando a busca para novas  reas. Como informaç es adicionais, o tempo computacional necess rio para executar cada configuraç o de par metros variou entre 30 e 50 segundos.

Para melhorar a visualizaç o dos dados obtidos para a Tabela 6.32, a Figura 6.31, apresenta o comportamento das m dias obtidas para o problema R206, com o operador UOBX e a pol tica de elitismo GE. No eixo x mostra-se a variaç o da taxa de cruzamento enquanto que no eixo y mostra-se o valor da funç o objetivo. As linhas mostram os diferentes valores analisados para a taxa de mutaç o.

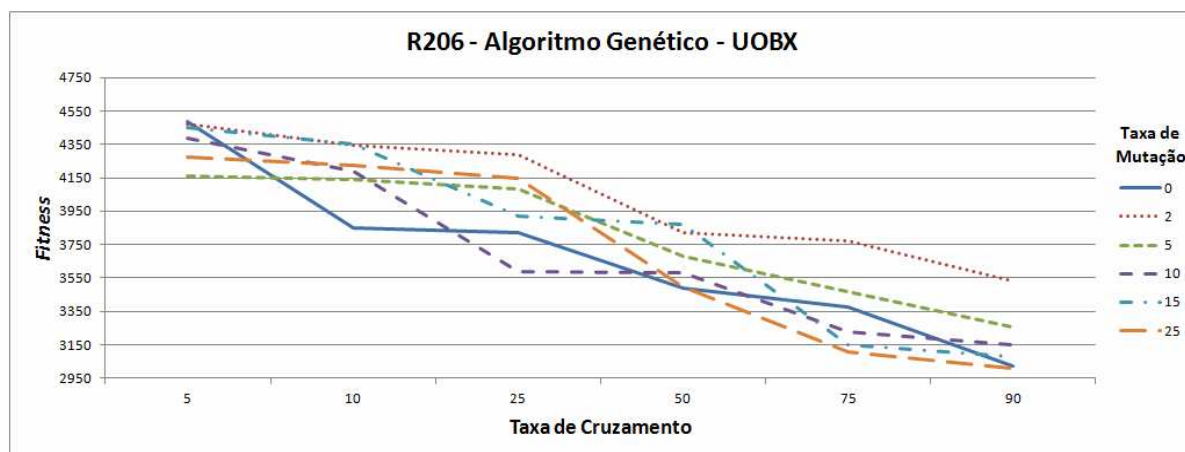


Figura 6.31 – Evolução do Algoritmo Genético com operador UOBX e com a política GE para o problema R206

Observa-se na Figura 6.31, que ao aplicar o operador UOBX à política de elitismo GE, a busca se torna mais eficiente. Percebe-se que o aumento da taxa de cruzamento está diretamente ligado a melhora das soluções obtidas. Além disso, vale lembrar que na política GE, um novo indivíduo é aceito na população somente se for mais apto que algum outro indivíduo presente na população. Essa característica dificulta a manutenção da diversidade da população, devido a isso, a maior parte dos resultados de boa qualidade foi obtida com altas taxas de mutação.

Visando analisar o processo de busca no espaço, realizado pelo Algoritmo Genético, aplicando a política de elitismos DJ e o operador OBX, a Tabela 6.33 mostra o número de incidências das melhores soluções obtidas com cada conjunto de parâmetros, analisando as Tabelas 6.15, 6.16 e 6.17.

Tabela 6.33 – Análise de busca com política DJ e operador OBX

Política DJ						
Operador de Cruzamento: OBX						
Taxa de Mutaç�o	Taxa de Cruzamento					
	5	10	25	50	75	90
0						2
2					3	3
5					1	1
10				1	3	1
15						1
25						1

Observa-se que as melhores solu es foram obtidas com taxas de cruzamento a partir de 50, mas com variadas taxas de muta o, na maior parte com taxas de muta o menores que 10. Essa ocorr ncia se deve pelas caracter sticas da pol tica DJ e pelo operador de cruzamento, onde a eleva o da taxa de muta o acaba guiando a busca para regi es n o exploradas do espa o de solu es.

A Tabela 6.34 mostra o n mero de incid ncias das melhores solu es obtidas com cada conjunto de par metros, analisando as Tabelas 6.21, 6.22 e 6.23, para a pol tica de elitismo DJ e operador PMX.

Tabela 6.34 – Análise de busca com política DJ e operador PMX

Política DJ						
Operador de Cruzamento: PMX						
Taxa de Mutaç�o	Taxa de Cruzamento					
	5	10	25	50	75	90
0	1	1		1		
2	1		1	1		
5		1		1	2	2
10				1		2
15			1	1	2	2
25						1

Na Tabela 6.34, verifica-se uma grande varia o nos conjuntos de par metros que obtiveram solu es de boa qualidade. Pode-se analisar que o operador PMX gerou maior varia o que o operador OBX, devido as suas caracter sticas.

A Tabela 6.35 mostra o n mero de incid ncias das melhores solu es obtidas com cada conjunto de par metros, analisando as Tabelas 6.27, 6.28 e 6.29, para a pol tica de elitismo DJ e operador UOBX.

Tabela 6.35 – Análise de busca com política DJ e operador UOBX

Política DJ						
Operador de Cruzamento: UOBX						
Taxa de Mutação	Taxa de Cruzamento					
	5	10	25	50	75	90
0		1		1		2
2		2		1		
5			1	1	1	2
10					2	1
15			1	2	1	1
25		1				2

Analisando a Tabela 6.35, verifica-se novamente uma grande variação nos conjuntos de parâmetros que obtiveram soluções de boa qualidade. Isso se deve as características da política de intensificação que não realiza uma exploração tão habilidosa no espaço de soluções.

A Tabela 6.36 mostra o número de incidências das melhores soluções obtidas com cada conjunto de parâmetros, analisando as Tabelas 6.18, 6.19 e 6.20, para a política de elitismo GE e operador OBX.

Tabela 6.36 – Análise de busca com política GE e operador OBX

Política GE						
Operador de Cruzamento: OBX						
Taxa de Mutaç�o	Taxa de Cruzamento					
	5	10	25	50	75	90
0					2	2
2						3
5						1
10						2
15					3	3
25					1	3

Analisando a Tabela 6.35, verifica-se que as melhores solu es foram obtidas com taxas de cruzamento entre 75 e 90. Ao comparar as Tabelas 6.36 e 6.33, o operador de cruzamento OBX gerou um comportamento semelhante, por m com a pol tica GE nenhum resultado de boa qualidade foi obtido com a taxa de cruzamento 50.

A Tabela 6.37 mostra o n mero de incid ncias das melhores solu es obtidas com cada conjunto de par metros, analisando as Tabelas 6.24, 6.25 e 6.26, para a pol tica de elitismo GE e operador PMX.

Tabela 6.37 – Análise de busca com política GE e operador PMX

Política GE						
Operador de Cruzamento: PMX						
Taxa de Mutação	Taxa de Cruzamento					
	5	10	25	50	75	90
0						2
2					3	3
5						2
10						3
15						1
25					1	3

Na Tabela 6.37, pode-se verificar que a política de elitismos GE aplicada ao operador de cruzamento PMX obteve resultados com taxas de cruzamento entre 75 e 90. Esse comportamento pode ser verificado também na Tabela 3.36, com o operador OBX. Ao comparar as Tabelas 6.37 e 6.34, percebe-se uma grande diferença nos conjuntos de parâmetros que obtiveram soluções de boa qualidade.

A Tabela 6.38 mostra o número de incidências das melhores soluções obtidas com cada conjunto de parâmetros, analisando as Tabelas 6.30, 6.31 e 6.32, para a política de elitismo GE e operador UOBX.

Tabela 6.38 – Análise de busca com política GE e operador UOBX

Política GE						
Operador de Cruzamento: UOBX						
Taxa de Mutação	Taxa de Cruzamento					
	5	10	25	50	75	90
0						2
2					1	1
5					1	3
10					1	3
15					1	2
25					2	3

Analisando a Tabela 6.38, pode-se verificar que a política de elitismos GE obteve resultados com altas taxas de cruzamento. Percebe-se esse comportamento em todos os dados analisados com base na política GE. Isso ocorre devido as características da política de elitismo, onde um indivíduo somente é aceito se for mais apto que outro indivíduo da população.

A Figura 6.32 mostra o percentual de melhora obtido comparando a melhor solução obtida com cada um dos operadores em relação ao indivíduo mais apto da população inicial, *fitness* de 6071,8, para o problema R206. As melhores soluções obtidas pelo operador OBX foram de 4756,64 e 2800,92 para as políticas DJ e GE, respectivamente. O operador PMX obteve as soluções 3517,39 e 2518,58 para as políticas DJ e GE, respectivamente, e o operador UOBX obteve as soluções 3550,78 e 2619,98 para as políticas DJ e GE, respectivamente.

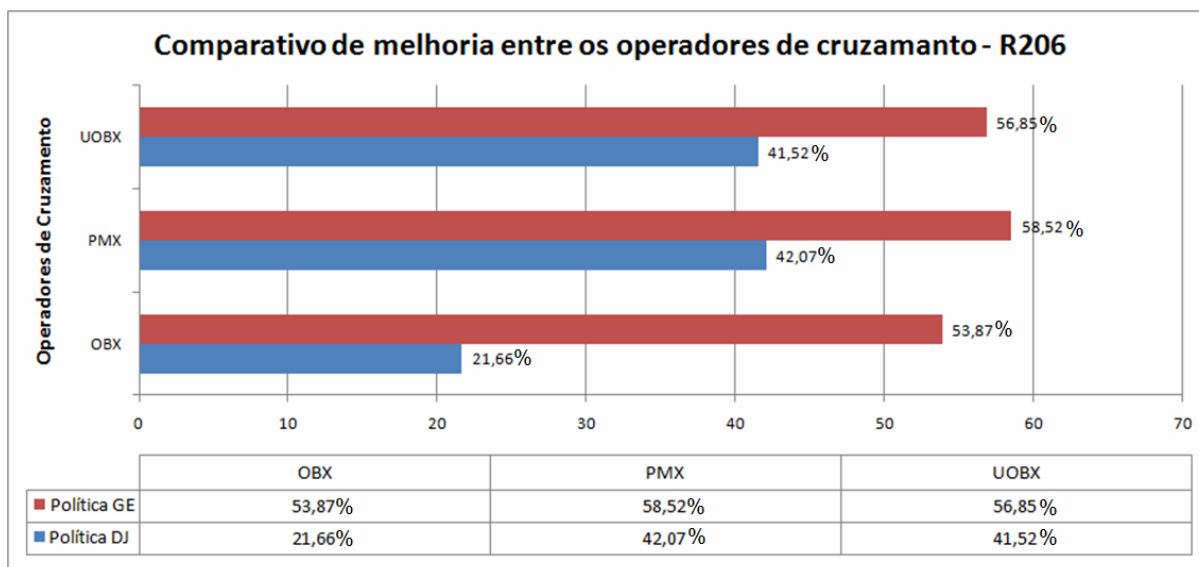


Figura 6.32 – Comparativo de melhora do para o problema R206.

Na Figura 6.32 pode-se verificar que a política elitista GE obteve resultados de boa qualidade, se comparados com a política elitista DJ. Além disso, o operador PMX obteve resultados de melhor qualidade comparados com os demais operadores, quando aplicado a política de elitismo GE, somente para o problema R206. Isso ocorre devido a política de elitismo GE tem a característica de intensificar a busca em uma região promissora.

6.3 Algoritmo Híbrido

A terceira fase de experimentos tem o objetivo de analisar a influencia de parâmetros no módulo do Algoritmo Híbrido. Com essa variação de parâmetros espera-se estudar o módulo e obter resultados de melhor qualidade em relação aos outros módulos com menor tempo computacional.

Nessa fase analisa-se a variação das taxas de cruzamento e mutação, do nbmax e do tamanho da Lista Tabu. O tamanho da população será de 100 indivíduos e o nbmax será de 200 iterações sem melhora da função objetivo. A Tabela 6.39 mostra os valores propostos para a taxa de cruzamento, a Tabela 6.40 mostra os valores propostos para a taxa de mutação e a Tabela 6.41 mostra os valores propostos para o tamanho da Lista Tabu.

Tabela 6.39– Relação de valores para a taxa de cruzamento

Taxa de cruzamento
75
95

Tabela 6.40 – Relação de valores para a taxa de mutação

Taxa de mutação
25
50

Tabela 6.41 – Relação de valores para o tamanho da Lista Tabu

Tamanho da Lista Tabu
25
50

Os resultados obtidos serão apresentados em tabelas seguindo o padrão da Tabela 6.42 onde a primeira linha irá designar o problema que está sendo utilizado nos experimentos. As colunas mostram a variação da taxa de cruzamento e as linhas mostram a variação da taxa de mutação em relação a variação do tamanho da Lista Tabu. Os resultados exibidos são o valor médio obtido para a Função Custo (FC) e o desvio padrão (σ) em torno dessa média, com base nos parâmetros mostrados.

Tabela 6.42 - Modelo de tabela de resultados

Problema: R206				
Lista Tabu	Taxa de Mutação		Taxa de Cruzamento	
			75	95
25	25	FC σ		
	50	FC σ		
50	25	FC σ		
	50	FC σ		

Os melhores e os piores resultados são destacados, em negrito, para facilitar a análise. Por fim, são traçados comparativos entre os melhores resultados obtidos por cada módulo, analisando a qualidade das soluções obtidas, o desvio padrão em torno dessas soluções e o tempo computacional necessário para obter tais resultados.

Iniciam-se os experimentos analisando o problema RC207. Nesse problema os clientes estão localizados parcialmente agrupados e parcialmente esparsos. A Tabela 6.43 apresenta os resultados obtidos para o problema RC207. As colunas apresentam a taxa de cruzamento enquanto que as linhas apresentam a taxa de mutação, aplicadas para um determinado tamanho de Lista Tabu, nas células são exibidas os valores médios obtidos para o valor da Função Custo e o desvio padrão em torno dessa média. Os valores destacados mostram a melhor média obtida e seu desvio padrão.

Tabela 6.43 – Algoritmo Híbrido aplicada ao problema RC207

Problema: RC207				
Lista Tabu	Taxa de Mutação		Taxa de Cruzamento	
			75	95
25	25	FC	1150,8	1002,4
		σ	101,2	39,5
25	50	FC	1143,1	993,8
		σ	98,6	30,9
50	25	FC	1134,2	1007,2
		σ	94,7	24,1
50	50	FC	1103,1	1015,5
		σ	56,4	45,7

Observa-se na Tabela 6.43, que os resultados de melhor qualidade foram obtidos utilizando altas taxas de cruzamento e mutação, além de uma Lista Tabu limitada. Esse comportamento se deve a dinâmica do Algoritmo Híbrido. Em seu processo de iteração ocorrem processos constantes de intensificação, forçando a população a convergir rapidamente. Com base nisso, faz-se necessário que os operadores genéticos gerem novos indivíduos causando várias mutações, de modo a guiar a busca para novas regiões do espaço de soluções. Uma vez que a busca é guiada para novas áreas do espaço de soluções, o processo de intensificação tem a possibilidade de explorar, de forma eficiente, locais promissores. Além disso, o valor elevado da Lista Tabu força a busca para novas áreas do espaço de soluções,

dificultando o processo de intensificação. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 45 e 75 segundos.

No problema R205, os clientes estão aleatoriamente distribuídos e as janelas de tempo são flexíveis. Desse modo, existe uma dificuldade intrínseca na exploração do espaço de soluções.

A Tabela 6.44 apresenta os resultados obtidos para o problema R205. As colunas apresentam a taxa de cruzamento enquanto que as linhas apresentam a taxa de mutação, aplicadas para um determinado tamanho de Lista Tabu, nas células são exibidas os valores médios obtidos para o valor da Função Custo e o desvio padrão em torno dessa média. Os valores destacados mostram a melhor média obtida e seu desvio padrão.

Tabela 6.44 – Algoritmo Híbrido aplicada ao problema R205

Problema: R205				
Lista Tabu	Taxa de Mutação		Taxa de Cruzamento	
			75	95
25	25	FC	1201,8	977,3
		σ	124,9	27,3
	50	FC	1129,4	960,9
		σ	107,0	11,1
50	25	FC	1160,4	1094,6
		σ	100,1	111,7
	50	FC	1135,3	971,5
		σ	97,9	21,7

Assim como para o problema RC207, os melhores resultados foram obtidos utilizando elevadas taxas de cruzamento e mutação, com um tamanho reduzido de Lista Tabu. Observa-se que essa característica se repete para o problema R206.

Pode ser verificado que o Algoritmo Híbrido necessita da diversificação proporcionada pelos operadores de cruzamento e mutação, enquanto que no processo de intensificação um tamanho elevado de Lista Tabu impede que a busca seja ainda mais intensificada e direciona-a para outras regiões do espaço de soluções. Como informações adicionais, o tempo computacional necessário para executar cada configuração de parâmetros variou entre 50 e 80 segundos.

Por fim, analisando o problema R206, a Tabela 6.45 apresenta os resultados obtidos para o problema R206. As colunas apresentam a taxa de cruzamento enquanto que as linhas apresentam a taxa de mutação, aplicadas para um determinado tamanho de Lista Tabu, nas células são exibidas os valores médios obtidos para o valor da Função Custo e o desvio padrão em torno dessa média. Os valores destacados mostram a melhor média obtida e seu desvio padrão.

Tabela 6.45 – Algoritmo Híbrido aplicada ao problema R206

Problema: R206				
Lista Tabu	Taxa de Mutação		Taxa de Cruzamento	
			75	95
25	25	FC	1203,7	1167,5
		σ	154,3	103,6
50	50	FC	1207,0	1118,2
		σ	149,8	106,4
50	25	FC	1234,5	1191,2
		σ	179,6	171,0
50	50	FC	1229,3	1174,6
		σ	197,7	155,9

Na Tabela 6.45, pode-se verificar que, novamente, os melhores resultados foram obtidos utilizando taxas de mutação e cruzamento elevadas, enquanto que o tamanho da Lista Tabu ficou reduzido. Desse modo, pode-se constatar que durante o processo de busca, realizado pelo Algoritmo Híbrido, na etapa de diversificação, quando maiores recombinações genéticas forem realizadas, de modo a guiar a buscar para regiões ainda não exploradas, maiores serão as chances de a etapa de intensificação obter soluções de boa qualidade. Como informações adicionais, o tempo computacional necessário para executar cem vezes cada configuração de parâmetros variou entre 50 e 75 segundos.

7 Conclusão

O presente trabalho aborda o Problema de Roteamento de Veículos Capacitado com Janelas de Tempo, onde devem ser atendidas as restrições de capacidade do veículo e as janelas de tempo de atendimento do cliente. Para resolver tal problema, além da utilização das metaheurísticas Busca Tabu e Algoritmos Genéticos, desenvolveu-se um Algoritmo Híbrido baseado nas duas metaheurísticas.

Ao analisar os experimentos verifica-se que a parametrização correta dos métodos influencia nas soluções obtidas. Para a Busca Tabu, observou-se que o aumento exagerado do tamanho da Lista Tabu acaba guiando a busca para regiões pouco promissoras do espaço de soluções. Por outro lado, o aumento do nbmax, permite que a Busca Tabu prolongue seu processo de exploração, muitas vezes, consumindo tempo computacional sem prover nenhum benefício. Ainda assim, equívocos na parametrização do algoritmo pode gerar uma busca superficial, a qual abrange um grande espaço de soluções, porém não realiza uma busca aprofundada em regiões promissoras. Além disso, fica claro que a aplicação da política de intensificação produz resultados de melhor qualidade, se comparados com a não utilização da política, sem comprometer o tempo computacional.

Analisando a parametrização do Algoritmo Genético, verificou-se que com o aumento da taxa de cruzamento, resultados de melhor qualidade são obtido sem uma mudança considerável no tempo computacional. Esse aumento promove uma convergência mais acentuada da população, promovendo uma intensificação da busca em uma região promissora. Em contra partida, o aumento excessivo da taxa de mutação faz com que o Algoritmo Genético acabe diversificando sua busca, retardando a convergência e permitindo que novas regiões possam ser exploradas. Pode-se verificar, também, que os operadores genéticos estão diretamente ligados à obtenção de soluções de melhor qualidade. Com a aplicação do operador UOBX, obtiveram-se resultados de melhor qualidade em grande parte dos experimentos, se comparados aos operadores OBX e PMX. Ainda assim, analisando as políticas de

elitismo, verifica-se que a política GE produz resultados de melhor qualidade, em comparação com a política DJ. A política de elitismo GE força o processo de intensificação do Algoritmo Genético, direcionando a busca para uma região promissora do espaço de soluções, contudo, ela necessita que o operador genético de mutação realize uma manutenção consistente da diversidade da população. Por outro lado, a política DJ promove uma busca mais ampla no espaço de soluções, uma vez que sua população sempre é reconstruída. Esse processo de reconstrução retarda o processo de convergência ocasionado pela política GE, mas também não consegue promover uma busca mais eficiente em áreas promissoras.

Analisando os resultados obtidos, verifica-se que a Busca Tabu obteve soluções de melhor qualidade que o Algoritmo Genético para o PRVCJT com desvios padrões menores. Isso se deve pela política de intensificação, a qual promove uma busca local em regiões promissoras. Apesar disso, a Busca Tabu vai se mostrando ineficiente à medida que o número de clientes e a complexidade aumentam.

Objetivando uma técnica que unisse o poder de busca local intensificada proporcionado pela Busca Tabu e o poder de diversificação proporcionado pelo Algoritmo Genético, desenvolveu-se um Algoritmo Híbrido. Esse algoritmo une o processo de Busca Tabu, com a aplicação da política de intensificação, ao processo do Algoritmo Genético com a aplicação da política de elitismo DJ, a qual demonstra realizar uma exploração mais ampla do espaço de soluções.

Os experimentos mostram que o Algoritmo Híbrido possui maior eficiência na obtenção de soluções de melhor qualidade, se comparado a Busca Tabu e ao Algoritmo Genético, e ainda se mostrou mais eficiente, gerando desvios padrões menores. Além disso, apesar do Algoritmo Híbrido unir as características das duas técnicas, o tempo computacional não sofre um aumento relevante, uma vez que a diferença é apenas alguns segundos em relação à Busca Tabu e ao Algoritmo Genético.

Ao comparar os resultados obtidos pelas diferentes técnicas, verifica-se que o Algoritmo Genético, conforme comentado anteriormente, não obteve soluções de tão boa qualidade, se comparado à Busca Tabu e o Algoritmo Híbrido. A Busca Tabu mostra-se eficiente em alguns casos, superando alguns resultados obtidos pelo

Algoritmo Híbrido, em problemas de menor complexidade. Contudo, ela se mostra ineficiente, se comparada ao Algoritmo Híbrido, quando ocorre um aumento da complexidade dos problemas. Em contra partida, o Algoritmo Híbrido desenvolvido se mostra flexível e eficiente, ao obter soluções de boa qualidade para todos os tipos de problemas analisados. Vale ressaltar que o Algoritmo Híbrido obteve muitas soluções conhecidas, *a priori*, como ótimas, além de obter soluções de melhor qualidade na maioria dos problemas se comparado com as melhores soluções heurísticas.

7.1 Trabalhos Futuros

No que tange a Busca Tabu, pode-se estudar a utilização de diferentes técnicas de intensificação e diversificação, bem como novas estruturas de vizinhança, voltadas ao PRVJT. Quanto ao Algoritmo Genético, pode-se utilizar uma taxa de mutação adaptativa, a qual poderá fazer a manutenção adequada da diversidade da população, proporcionando que busca convirja para áreas promissoras ao mesmo tempo em que diversifique quando necessário. Ainda assim, quanto ao Algoritmo Híbrido, novas técnicas de intensificação podem ser utilizadas no lugar da Busca Tabu, como, por exemplo, métodos exatos. A diferença entre a utilização de um método exato e a Busca Tabu estudada está na obtenção do ponto ótimo em relação ao tempo computacional. A Busca Tabu pode gerar soluções de boa qualidade, próximas ao ótimo com um tempo computacional baixo, entretanto, não há garantia da otimalidade desse ponto, em contra partida aos métodos exatos, os quais irão obter esse ponto ótimo. Ainda assim, o Algoritmo Híbrido pode ser aplicado a problemas com mais clientes, de modo a mostrar todo seu poder de busca.

8. Referências Bibliográficas

AARTS, E.H.L., LENSTRA, J.K. **Local Search in Combinatorial Optimization**. Wiley, Chichester, 1997.

ALFA, A.S., HERAGU, S.S., CHEN, M. **A 3-opt based simulated annealing algorithm for vehicle routing problems**. Computers & Industrial Engineering, v.21, p.635-639, 1991.

ARCHETTI, C., BERTAZZI, L., LAPORTE, G., SPERANZA, M. G. **A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem**. Transportation Science, v.41, p. 382–391, 2007.

BAKER, B. M., AYECHIEW, M. A. **A genetic algorithm for the vehicle routing problem**. Computers and Operations Research, v.30, p.787-800, 2003.

BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos – Planejamento, Organização e Logística Empresarial**, 5ª edição, Porto Alegre: Bookman, 2006.

BARBAROSOGLU, G.; OZGUR, D. **A tabu search algorithm for the vehicle routing problem**. Computers and Operations Research v.26, p.255-270, 1999.

BARTHOLDI III, J.J., PLATZMAN, L.K., LU, C.R., WARDEN, W.W. **A minimal technology routing system for meals on wheels**. Interfaces v.13, p.1-8, 1983.

BELLMANN, R. R. **On a Routing Problem**. Quarterly Applied Mathematics, v.16, p.87-90, 1958.

BENJAMIN, A. M., BEASLEY, J. E. **Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities**. Computers and Operations Research, v.37, p. 2270-2280, 2010.

BENT , R., HENTENRYCK , P. V., **A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows.** Technical Report CS-01-06, Department of Computer Science, BrownUniversity, 2001.

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.

BIANCHESSI, N., RIGHINI, G. **Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery.** Computer and Operations Research, v.34, p.578-594, 2007.

BODIN, L.D., GOLDEN, B.L., ASSAD, A.A., BALL, M.O. **Routing and Scheduling of vehicles and crews: The State of the Art.** Computers and Operations Research v.10, p.69-211, 1983.

BOLDUC, M. C., LAPORTE, G., RENAUD, J., BOCTOR, F. F. **A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars.** European Journal of Operational Research, v.202, p.122–130, 2010.

BRANDÃO, J. **A new tabu search algorithm for the vehicle routing problem with backhauls.** European Journal of Operations Research, v.173, p.540-555, 2006.

BRANDÃO, J. **A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem.** Computers and Operations Research, v.38, p.140-151, 2010.

BREEDAM, A. V. **Comparing descent heuristics and metaheuristics for the vehicle routing problem.** Computer & Operations Research v.28, p.289-215, 2001.

BREEDAM, A. V. **Improvement heuristics for the Vehicle routing Problem based on Simulated Annealing.** European Journal of Operational Research, v.86, p.480-490, 1995.

BREMERMANN, H. J., Rogson, J., Salaff, S. **Global properties of evolution process.** In: Pattee H. H., **Natural Automata and Useful Simulations**, p.3-42, 1964.

CHABRIER A. **Vehicle Routing Problem with Elementary Shortest Path based Column Generation.** Computers and Operations Research, v.33, p. 2972-2990, 2005.

CEDER, A., STERNY, H.I. **Deficit Function Bus Scheduling with Deadheading trip insertion for fleet size reduction.** Transportation Science v.15, p.338-363, 1981.

CHEN, C. H., TING, C. J. **A Hybrid Ant Colony System For Vehicle Routing Problem With Time Windows.** Eastern Asia Society for Transportation Studies, v.6, p. 2822-2836, 2005.

CHEN, C. H., TING, C. J. **Applying Two-Stage Ant Colony Optimization to Solve the Large Scale Vehicle Routing Problem.** Eastern Asia Society for Transportation Studies, v.7, p.1-17, 2009.

CHIYOSHI, F., GALVÃO, R. D. **A statistical analysis of simulated annealing applied to the p-median problem.** Annals of Operations Research, v.96, p.61–74, 2000.

CHRISTOFIDES, N., MINGOZZI, A., TOTH, P. **The vehicle routing problem.** In **Christofides, N., Mingozi, A., Toth, P., Sandi, C., Combinatorial Optimization**, Wiley, Chichester, UK, 1979, pp. 315-338.

CLARKE, G., WRIGHT, J. W. **Scheduling of vehicles from a central depot to a number of delivery points.** Operations Research, v.12, p.568-581, 1964.

COOK, W., RICH, J. L. **A parallel cutting plane algorithm for the vehicle routing problem with time windows.** Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

COOK, W. J; CUNNINGHAM, W. H; PULLEYBLANK, WILLIAM R; SCHRIJVER, A. **Combinatorial Optimization.** Willey, 1998.

CORDEAU, J. F., DESAULNIERS, G., DESROSIERS, J., SOLOMON, M. M., Soumis, F. **VRP with Time Windows.** In: **Toth, P. Vigo, D., The Vehicle Routing Problem**, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

CORDEAU, J-F., Gendreau, M., LAPORTE, G., POTVIN, J.-Y., SEMET, F. **A guide to vehicle routing heuristics.** Journal of the Operational Research Society, v.53, p.512-522, 2002.

CORDEAU, J.-F., LAPORTE, G., MERCIER, A. **A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows.** Working Paper CRT-00-03, Centre for Research on Transportation, Montreal, Canada, 2000.

CHRISTOFIDES, N. ELION, S. **An Algorithm for the vehicle Dispatching Problem.** Operational Research Quarterly, v.20, n.3, 1969.

CHRISTOFIDES, N., MINGOZZI, A., TOTH, P. **Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations.** Mathematical Programming, v.20, p.255-282, 1981

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, p. 376-383, 2002.

DANNA, E., LE PAPE, C. **Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows.** In: Column Generation, G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), 99-130, Kluwer Academic Publishers, 2005.

DANTZIG, G.B., RAMSER, J.H. **The truck dispatching problem.** Management Science, v.6, p.80-91, 1959.

DAVIS, I. **Handbook of Genetic Algorithms.** Van Nostrand Reinhold, New York, 1991.

DE JONG, K. A. **Genetic Algorithms in Search, Optimization, and Machine Learning,** Addison-Wesley, Reading, Massachusetts, 1975.

DELL'AMICO, M., MONACI, M., PAGANI, C., VIGO, D. **Heuristic Approaches for the Fleet Size and Mix Vehicle Routing Problem with Time Windows.** Transportation Science, v.41, p. 516–526, 2007.

DESAULNIERS, G., LESSARD, F., HADJAR, A. **Tabu Search, Partial Elementarity, and Generalized k-Path Inequalities for the Vehicle Routing Problem with Time Windows.** *Transportation Science*, v.42, p. 387–404, 2008.

DRIDI, I. H., Kammarti, R, Ksouri, M., Borne P. **A Genetic Algorithm for the Multi-Pickup and Delivery Problem with time windows.** *Information Control Problems in Manufacturing*, v.13, p.1-17, 2010.

DUECK,G. **New optimization heuristics: The great deluge algorithm and the record-top-record travel.** *Journal of Computational Physics*, v.104, p.86-92, 1993.

DUECK,G., SCHEURER,T. **Threshold accepting: A general purpose optimization algorithm.** *Journal of Computational Physics*, v.90, p.161-175, 1990.

EASWARAN, G., USTER, Ü. **Tabu Search and Benders Decomposition Approaches for a Capacitated Closed-Loop Supply Chain Network Design Problem.** *Transportation Science*, v.43, p. 301–320, 2009.

ERGUN, O., ORLIN, J. B., STEELE-FELDMAN, A. **Creating very large scale neighbourhoods out of smaller ones by compounding moves.** *Journal of Heuristics*, v.12, p.115–140, 2006.

FISHER, M. L., JAIKUMAR, R. **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*. Department of Decision Sciences, The Wharton School, University of Pennsylvania, Philadelphia, VOL. 15:2, p. 10-21, 1985.

FISHER, M. L., **Optimal solution of vehicle routing problems using minimum k-trees.** *Operations Research* v.42, n.4, p.626-642, 1994.

FRANCESCHI, R. D., FISCHETTI, M., TOTH, P. **A new ILP-based refinement heuristic for Vehicle Routing Problems.** *Mathematical Programming*,v.105, p.471–499, 2006.

GAMBARDELLA, L. M., TAILLARD, E., AGAZZI, G. **MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows.** In *New Ideas in Optimization*, D. Corne, M. Dorigo and F. Glover (eds), McGraw-Hill, London, p. 63-76, 1999.

GEIGER, M. **Genetic Algorithms for multiple objective vehicle routing.** *Proceedings of the Metaheuristics International Conference*, p.349-353, 2008.

GENDREAU, M., POTVIN, J.-Y. **Metaheuristics in Combinatorial optimization.** *Annals of Operations Research*, v.140, p.189-213, 2005.

GENDREAU, M., LAPORTE, G., POTVIN, J. Y. **Metaheuristics for the Capacitated VRP.** In: **Toth, P. Vigo, D., The Vehicle Routing Problem**, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

GENDREAU, M., LAPORTE, G. POTVIN, J-Y. **Metaheuristics for the vehicle routing problem.** Technical Report, Centre de Recherche sur les transports, Université de montreal, 1999.

GENDREAU, M., HERTZ, G., LAPORTE, G., STAN, M. **A generalized insertion heuristic for the traveling salesman problem with time windows.** *Operations Research*, v.43, p.330-335, 1998.

GENDREAU, M., HERTZ, A., LAPORTE, G. **A Tabu Search Heuristic for the Vehicle Routing Problem.** *Management Science* v.40, p.1276-1290, 1994.

GHAZIRI, H. **Solving routing problems by a self-organizing map.** In **Kohonen, T., Makisara, K., Simula, O., Kangas, J., Artificial Neural Networks, North-Holland, Amsterdam**, p. 829-834, 1991.

GILLET, B. E., MILLER, L. R. **A heuristic algorithm for the vehicle dispatch problem.** *Operations Research*, v.22, p.240-349, 1974.

GLOVER, F., KOCHENBERGER, G. A., **Handbook of Metaheuristics.** Kluwer Academic Publishers, 2003.

GLOVER, FRED. **Future paths for integer programming and links to artificial intelligence.** Computers and Operations Research v.1, p.533-549, 1986.

GOLDBARG, M.C.,LUNA, H.P. **Otimização Combinatória e Programação Linear: Modelos e Algoritmos.** Campus, 2000.

GOLDBERG, D. E., LINGLE, R. **Alleles, loci and the traveling salesman problem.** In: **Grefenstette, J. J., Proceedings of an International Conference on Genetic Algorithms and Their Applications.** Lawrence Erlbaum Associates, Hillsdale, New Jersey, p.154-159, 1985.

GOLDEN,B.L; MAGNANTI, T.L.; NGUYEN, H.Q. **Implementing Vehicle Routing Algorithms.** Networks v.7, p.113-148, 1977.

GARG, N., HOCHBAUM, D. S. **An $O(\log k)$ -Approximation Algorithm for the k Minimum Spanning Tree Problem in the Plane.** Algorithmica, vol. 18(1), p. 111-121, 1997.

HADJICONSTANTINOU, E. CHRISTOFIDES, N., MINGOZZI, A. **A new Exact algorithm for the vehicle routing problem based on q -Path and k -Shortest Path Relaxation.** Annals of Operations Research, v.61, p.21-44, 1995.

HO, S.C, HAUGLAND, D. **A tabu search heuristic for the vehicle routing problem with time windows and split deliveries.** Computers and Operations Research v.31, p.1947-1964, 2004.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems.** University of Michigan Press, Ann Arbor, Michigan, 1975.

HOMBERGER, J., GEHRING, H. **Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows.** INFOR, v. 37, p. 297-318, 1999.

HOMBERGER, J. **Verteilt-parallele Metaheuristiken zur Tourenplanung.** Gaber, Wiesbaden, 2000.

HU, X. B., DI PAOLO, E. **An Efficient Genetic Algorithm with Uniform Crossover for the Multi-Objective Airport Gate Assignment Problem.** Evolutionary Computation, p.55-62, 2007.

IBARAKI, T., KUBO, M., MASUDA, T., UNO, T., YAGIURA, M. **Effective Local Search Algorithms for the Vehicle Routing Problem with General Time Windows.** Working Paper, Department of Applied Mathematics and Physics, Kyoto University, Japan, 2001.

IORI, M., GONZÁLEZ, J. J. S., VIGO, D. **An Exact Approach for the Vehicle Routing Problem with Two-Dimensional Loading Constraints.** Transportation Science, v.41, p.253-264, 2007.

IRNICH, S., VILLENEUVE, D. **The shortest path problem with k-cycle elimination ($k \geq 3$): Improving a branch-and-price algorithm for the VRPTW.** INFORMS Journal of Computing, 2005.

JIANG, H., BARNHART, C. **Dynamic Airline Scheduling.** Transportation Science, v.43, p. 336-354, 2009.

JOHNSON, D. S. **Approximation algorithms for combinatorial problems.** J. Comput. System Sci., vol. 9, p. 256-278, 1974.

Kallehauge, B., Larsen, J., Madsen, O.B.G. **Lagrangian duality and non-differentiable optimization applied on routing with time windows - experimental results.** Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.

KAWAMURA, H., YAMAMOTO, M., MITAMURA, T., SUZUKI, K., OHUCHI, A. **Cooperative search on pheromone communication for vehicle routing problems.** IEEE Transactions on Fundamental, E81-A.1089-1096, 1998.

KIRKPATRICK, S., GELLATT, C.D. Jr. VECCHI, M.P. **Optimization by simulated annealing.** Science, v.20, n 4598, p.671-680, 1983.

KOHL , N., DESROSIERS , J., MADSEN , O. B. G., SOLOMON , M. M., SOUMIS, F. **2-Path Cuts for the Vehicle Routing Problem with Time Windows.** *Transportation Science*, v. 33 (1), p. 101-116, 1999.

KULCAR,T. **Optimizing Solid Waste Colletion in Brussels, European Journal of Operational Research.** v.90, p.71-77, 1996.

KYTÖJOKI, J., NUORTIO, T., BRAYSY, O., GENDREAU, M. **An efficient variable neighborhood search heuristic for very large scale vehicle routing problems.** *Computers and Operations Research*, 2005.

LAPORTE, G. **Fifty Years of Vehicle Routing.** *Transportation Science.* v43, p. 408–416, 2009.

LAPORTE,G., GENDREAU,M., POTVIN,J., SEMET, F. **Classical and modern heuristics for the vehicle routing problem.** *Intl.Trans. in Op. Res* v.7, p.285-300, 2000.

LAPORTE, G. SEMET, F. **Classical Heuristics for the Vehicle Routing Problem.** *Les Cahiers du GERAD, G98-54, Group for Research in Decision Analysis, Montreal, Canada, 1998.*

LAPORTE, G. **The Vehicle Routing Problem: An overview of exact and approximate algorithms.** *European Journal of operational Research* v.59, p.345-458, 1992.

LAPORTE, G., NOBERT, Y. **A cutting planes algorithm for the m-salesman problem.** *Journal of the Operational Research Society*, v.31, p.1017-1023,1980.

Larsen , J. **Parallelization of the vehicle routing problem with time windows.** Ph.D. Thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.

LENSTRA,J.K, RINNOOY Kan, G. **Complexity of Vehicle Routing and Scheduling Problems.** *Networks* v.11, p.221-227, 1981.

LI, H., LIM, A., HUANG, J. **Local Search with Annealing-like Restarts to Solve the VRPTW**. Working Paper, Department of Computer Science, National University of Singapore, 2001.

LIN, S. **Computer solution of the traveling salesman problem**. Bell System Technical Journal 44,2245-2269, 1965.

LIU, S. C., CHUNG, C. H. **A heuristic method for the vehicle routing problem with backhauls and inventory**. Journal of Intelligent Manufacturing, v.20, p.29–42, 2009.

LYSGAARD, J., LETCHFORD, A. N., EGLESE, R.W., **A new branch-and-cut algorithm for the capacitated vehicle routing problem**. Mathematical Programming, v.100, p.423–445, 2004.

MALMBORG, C.J., SIMONS, G.R. **Integrating logistical and Processing Functions through mathematical modeling**. Applied Mathematical Modelling v.13, n6, p.357-364, 1989.

MARINAKIS, Y., MIGDALAS, A. **Heuristic Solutions of Vehicle Routing Problems in Supply Chain Management**. Combinatorial and Global Optimization, P.M. Pardalos et al. (ed.s), Scientific World, p. 205-235, 2002.

MARTELLO, S., TOTH, P. **Knapsack Problems: Algorithms and Computer Implementations**, Wiley, Chichester, UK, 1990.

MARTINHON, C., LUCENA, A., MACULAN, N. **Stronger K-tree relaxations for the vehicle routing problem**. European Journal of Operational Research, v.158, p.56-76, 2004.

MATSUYAMA, Y. **Self-organization via competition, cooperation and categorization applied to extended vehicle routing problem**. In: **Proceeding of International Joint Conference on Neural Networks**, p.385-390, Seattle, WA, 1991.

MESTER, D. **An Evolutionary Strategies Algorithm for Large Scale Vehicle Routing Problem with Capacitate and Time Windows Restrictions**. Working Paper, Institute of Evolution, University of Haifa, Israel, 2002.

MESTER, D., BRAYSY, O. **Active-guided evolution strategies for large-scale capacitated vehicle routing problems.** Computers & Operations Research v.32, p.1593-1614, 2005.

MLADENOVIC, N., HANSEN, P. **Variable neighborhood search.** Computers and Operations Research, v.24, p.1097–1100, 1997.

MOLE, R. H., JAMENSON, S. R. **A sequential route-building algorithm employing a generalized savings criterion.** Journal Chemical and Physical, v.27, p.503-511, 1976.

MONTANÉ, F. A. T., GALVÃO, R. D. **A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service.** Computers and Operations Research, v.33, p.595–619, 2006.

MONTEMANNI, R., BARTA, J., MASTROLILLI, M., Gambardella, L. M. **The Robust Traveling Salesman Problem with Interval Data.** Transportation Science, v.41, p. 366–381, 2007.

NALLUSAMY, R., DURAISWAMY, K., DHANALAKSMI, R., PARTHIBAN, P. **Optimization of Multiple Vehicle Routing Problems Using Approximation Algorithms.** International Journal of Engineering Science and Technology, v.1, p.129-135, 2009.

NEWTON, R, THOMAS,W. **Bus Routing in Multischoll System.** Computers and Operational Research, v.1, p.213-222, 1974.

NOWAK, M., Ergun, O., White, C. C. III. **Pickup and Delivery with Split Loads.** Transportation Science, v.42, p32-43, 2008.

OSMAN, I.H. **Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem.** Annals of Operations Research, 41:421-451, 1993.

OSMAN, I.H., LAPORTE,G. **Metaheuristics: A bibliography.** Annals of Operations Research 63, 513-628, 1996.

PAUL, G. **Comparative Performance of Tabu Search and Simulated Annealing Heuristics for the Quadratic Assignment Problem.** Operations Research Letters, v.38, p.577-581, 2010.

PERLS, J., DASKIN, M.S. **A warehouse location-routing problem.** Transportation Research, v.19B, p.381-396, 1985.

PRINS, C. **A simple and effective evolutionary algorithm for the vehicle routing problem.** Computers and Operations Research v.31, p.1985-2002, 2004.

RALPH, T.K., KOPMAN, L., PULLEYBLANK, W.R., TROTTER, L.E. **On the Capacitated Vehicle Routing Problem.** Mathematical programming, v.94, p.343-359, 2001.

RAYWARD-SMITH, V., ORMAN, I., REEVES, C., SMITH, G. **Modern heuristic search methods.** Chichester: Wiley, 1996.

REEVES, C. **Genetic Algorithms.** In: Glover, F., Kochenberger, G. A., **Handbook of Metaheuristics**, Kluwer Academic Publishers, 2003.

REGO, C. **A subpath ejection method for the vehicle routing problem.** Management Science, v.44, p.1447-1459, 1998.

REGO, C., ROUCAIROL, C. **A parallel tabu search algorithm using ejection chains for the vehicle routing problem.** In: Osman I.H. and Kelly J.P (eds) **Meta-Heuristics: Theory and Applications.** Kluwer:Boston, 661-675, 1996.

REINELT, G. **The travelling salesman problem, Computational solutions for TSP Applications.** Springer-Verlag, 1994.

REYES, L. C., ORTA, J. F. D., BARBOSA, J. J. G., JIMENEZ, J. T., HUACUJA, H. J. F., CRUZ, B. A. A. **An Ant Colony System Algorithm to Solve Routing Problems Applied to the Delivery of Bottled Products.** Foundations Of Intelligent Systems, Lecture Notes in Computer Science, v.4994, p.329-338, 2008.

ROBUSTÉ, F., DAGANZO, C.F., SOULEYRETTE II, R. **Implementing vehicle models.** Transportation Research, 24B.263-286, 1990.

ROCHAT, Y., TAILLARD, E.D. **Probabilistic diversification and intensification in local search for vehicle routing.** Journal of Heuristics, v.1, p.147-167, 1995.

ROPKE, S., CORDEAU, J.F. **Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows.** Transportation Science, v.43, p.267-286, 2009.

ROPKE, S., PISINGER, D. **An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows.** Transportation Science, v.40, p. p. 455–472, 2006.

ROUSSEAU, L.M., GENDREAU, M. PESANT, G. **Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows.** Journal of Heuristics, v8, p. 43-58, 2002.

SCHRIMPF, G., SCHNEIDER, J., STAMM-WILBRANDT, H., DUECK, G. **Record Breaking Optimization Results Using the Ruin and Recreate Principle.** Journal of Computational Physics, v. 159, p. 139-171, 2000.

SCHUMANN, M., RETZKO, R. **Self-organizing maps for vehicle routing problems minimizing an explicit cost function.** In F. Fogelman-Soulie, editor, Proceedings of the International Conference on Artificial Neural Networks, Paris, p.401-406, 1995.

SHAW, P. **A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems.** Working Paper, University of Strathclyde, Glasgow, Scotland, 1997.

SHAW, P. **Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems.** In Principles and Practice of Constraint Programming - CP98, Lecture Notes in Computer Science, 417-431, M. Maher and J.-F. Puget (eds), Springer-Verlag, New York, 1998.

SIMAS, E. L. P. **Utilizando Busca Tabu na Resolução do Problema de Roteamento de Veículos.** Dissertação (Mestrado em Computação Aplicada) – Programa interdisciplinar de Pós-Graduação em Computação Aplicada. Universidade do Vale do Rio dos Sinos, São Leopoldo, RS, 2007.

SIMAS, E. P. L., GÓMEZ, A. T. **Comparing a tabu search process - using and not using and intensification strategy to solve the vehicle routing problem.** ICINCO-ICSO, p.77-84, 2007

SOLOMON, M. M. **Algorithms for the vehicle routing and scheduling problem with time window constraints.** Operations Research, vol. 35(2), p. 254–265, 1987.

SYSWERDA, G. **Schedule optimization using genetic algorithms.** In: Davis, I. **Handbook of Genetic Algorithms.** Van Nostrand Reinhold, New York, p.332-349, 1991.

TAILLARD, É.D. **Parallel iterative search methods for the vehicle routing problems.** Networks v.23, p.661-673, 1993.

TAILLARD, E., BADEAU, P., GENDREAU, M., GEURTIN, F., POTVIN, J.Y. **A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows.** Transportation Science, v. 31, p. 170-186, 1997.

TARANTILIS, C.D., IOANNOU, G., PRASTACOS, G. **Advanced vehicle routing algorithms for complex operations management problems.** Journal of Food Engineerig, v.70, p.455-471, 2005

THANGIAH, S.R; PETROVIC, P. **Introduction to Genetic Heuristics and vehicle Routing Problems with Complex Constraints.** In: Woodruff, David, L. **Advances in Computacional and Stochastic Optimization, Logic programming , and Heuristic search: Interfaces in Computer Science and Operations research.** Kluwer Academic Publishers, 1997.

THOMPSON, P. M., PSARAFTIS, H. N. **Cyclic Transfer algorithms for the multivehicle routing and scheduling problems.** Operations Research, v.41, p.935-946, 1993.

TIMLIN, M.T., PULLEYBLANK,W.R. **Procedence Constrained Routing and Helicopter Scheduling: Heuristic Design.** Departament of Combinatoric and Optimization University of Waterloo, N2L 3G1, Canadá, 1990.

TOTH, P., & VIGO, D. **The granular tabu search and its application to the vehicle-routing problem.** INFORMS Journal on Computing, v.15, n.4, p.333-346,2003.

TOTH, P., VIGO, D. **Models, relaxations and exact approaches for the capacitated vehicle routing problem.** Discrete Applied Mathematics v.123, p.487-512, 2002.

TYAGI, M., **A Pratical Method for the Truck Dispatching Problem.** Journal of Operations Research Society of Japan, v.10, p.76-92, 1968.

VAIDYANATHAN, B., AHUJA, R. K., ORLIN, J. B. **The Locomotive Routing Problem.** v.42, p. 492–507, 2008.

VIGO, D. **A heuristic algorithm for the Assimetric Capacitated Vehicle Routing Problem.** European Journal of Operational Research, p. 108-126, 1994.

XU, J., KELLY, J.P. **A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem.** Transportation Science, v.30, p.379-393, 1996.