

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Um Algoritmo Híbrido para o Problema de Roteamento de Veículos Estático e Dinâmico com Janela de Tempo

Guilherme Bastos Alvarenga

Tese apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para obtenção do Título de Doutor em Ciência da Computação.

Orientador: Prof. Geraldo Robson Mateus

Belo Horizonte, março de 2005

Resumo

O Problema de Roteamento de Veículos Estático com Janela de Tempo (PRVJT) é um dos problemas bem conhecidos em otimização combinatória que mais tem recebido atenção nos últimos anos. O objetivo do problema é planejar rotas para uma frota de veículos, sem violação das restrições de tempo e capacidade, minimizando custos. Os custos normalmente estão relacionados à distância total percorrida, o número de veículos necessário ao atendimento, o tempo total de espera nos consumidores ou combinação destes. O Problema de Roteamento de Veículos Dinâmico com Janela de Tempo (PRVDJT), por outro lado, é uma generalização do anterior onde parte das variáveis é conhecida somente após o início do processo de otimização.

Métodos exatos têm sido normalmente propostos para a versão estática do PRVJT. Melhores resultados com este tipo de método têm sido possível devido ao uso de modernas técnicas de planos de corte (*branch-and-cut*) e implementações paralelas. Entretanto, 23 dos 56 problemas de Solomon continuam em aberto. Adicionalmente, em muitos casos um tempo proibitivo é necessário para encontrar a solução exata. Muitas heurísticas têm sido desenvolvidas para possibilitar uma solução de boa qualidade dentro de um intervalo aceitável de processamento. Utilizando a distância total percorrida como único objetivo, uma abordagem híbrida é proposta para o PRVJT, utilizando um eficiente algoritmo genético combinado com uma formulação do problema por particionamento de conjunto. Testes foram realizados utilizando cálculos com dupla precisão e com inteiros, possibilitando comparar os resultados com aqueles anteriores, gerados por métodos exatos e heurísticas. Os resultados computacionais mostram que a heurística proposta supera todas as anteriores em termos de distância total percorrida. Os resultados utilizando cálculos com inteiros também são muito competitivos, comparados aos ótimos conhecidos na literatura.

Entretanto, um grande número de heurísticas utilizam a redução do número de veículos como primeiro objetivo e distância total percorrida somente como segundo. Uma fase adicional foi então proposta minimizando o número de veículos. Uma seleção baseada em um processo de torneio utilizando critérios hierárquicos foi proposta para o algoritmo genético. Todos os melhores resultados da literatura em termos de número de veículos para

as instâncias de Solomon foram alcançados. Depois de minimizado o número de veículos, a proposta inicial para minimização de distância é utilizada, sob uma população de indivíduos solução com número de veículos reduzido.

Finalmente, a proposta é aplicada ao problema dinâmico. Utilizando parte dos consumidores sendo incluídos ou cancelados após o início do processo de otimização. As instâncias de Solomon foram modificadas, possibilitando considerar vários graus de dinamismo. Com resultados desejados bem conhecidos para o problema dinâmico, foi possível uma melhor avaliação da qualidade dos resultados alcançados na proposta para o PRVDJT.

Abstract

The Vehicle Routing Problem with Time Windows (VRPTW) is a well-known and complex combinatorial problem, which has received considerable attention in recent years. The objective of the problem is to plan routes for vehicles, with no violated constraints, minimizing costs. The costs normally are related to the total travel distance, the number of vehicle or routes utilized, the total wait time in the customers or combination of these. The DVRPTW is the dynamic generalization of the VRPTW, where part of the information is only known after the initial of the optimization process.

Exact methods have normally proposed for the static version of the VRPTW. Results from this type of approach have been improved exploring parallel implementations and modern branch-and-cut techniques. However, 23 out of the 56 high order instances from Solomon's test set still remain unsolved. Additionally, in many cases a prohibitive time is needed to find the exact solution. Many efficient heuristic methods have been developed to make possible a good solution in a reasonable amount of time. Using travel distance as the main objective, in this thesis, a robust heuristic approach for the VRPTW using an efficient genetic algorithm and a set partitioning formulation is proposed. The tests were run using both, real numbers and truncated data type, making it possible to compare the results with previous heuristic and exact methods. Furthermore, computational results show that the proposed heuristic approach outperforms all previous known heuristic methods in the literature, in terms of the minimal travel distance.

However, a great number of heuristics has used the number of vehicles as the first objective and travel distance as the second, subject to the first. An additional phase is proposed to minimize the number of vehicles. Initially, a hierarchical tournament selection genetic algorithm is applied. It can reach all best results in number of vehicles of the 56 Solomon's problems explored in the literature. After then, the two phase approach, the genetic and the set partitioning, is applied to minimize the travel distance as the second objective.

Finally, the proposed framework is applied a dynamic version of the problem. Using part of customers being inserted or canceled after the initial of the optimization process, the

instances of Solomon are modified, making it possible to consider many degrees of dynamism. The fact of the desired results of the dynamic version of the proposed instances are the same of the static version, has make possible to evaluate the quality of the results from the proposed heurist for the DVRPTW.

Lista de Abreviações

Abreviação	Significado	Primeira ocorrência
Adap_CTRR	Adaptabilidade no AG_NV calculada sobre a carga total da rota receptora	96
Adap_DETCMR	Adaptabilidade no AG_NV quanto à dificuldade em eliminar todos os consumidores da menor rota	94
Adap_DETCRR	Adaptabilidade no AG_NV quanto à dificuldade de eliminar todos os consumidores da rota receptora	95
Adap_DEUCMR	Adaptabilidade no AG_NV quanto à dificuldade em eliminar um consumidor da menor rota	93
Adap_DEUCRR	Adaptabilidade no AG_NV quanto à dificuldade de eliminar um consumidor da rota receptora	94
Adap_DT	Adaptabilidade no AG_NV calculada sobre a distância total percorrida	95
Adap_NCMR	Adaptabilidade no AG_NV quanto ao número de consumidores na menor rota	93
Adap_NCRC	Adaptabilidade no AG_NV calculada sobre o número de consumidores na rota receptora	95
Adap_NV	Adaptabilidade no AG_NV quanto ao número total de veículos	94
Adap_SQNC	Adaptabilidade no AG_NV calculada como a soma dos quadrados dos números de consumidores das rotas	95
AG	Algoritmo Genético	38
AG_DT	Algoritmo genético desenvolvido para redução da distância total percorrida no PRVJT	59
AG_NV	Algoritmo genético desenvolvido para redução do número de veículos no PRVJT	92
AG_PARAM	Algoritmo genético desenvolvido para busca de parâmetros otimizados em AG_DT	120
CVRP	<i>Capacitated Vehicle Routing Problem</i> . Nome pelo qual o PRVC é mais conhecido na literatura.	5
DARP	<i>Dial-a-Ride Problem</i>	6
DT	Distância Total percorrida	27
ESPTCC	<i>Elementary Shortest Path Problem with Time Window and Capacity Constraint</i>	16
FIFO	<i>First In Last Out</i>	31
continua na página seguinte		

continuação da página anterior		
Abreviação	Significado	Primeira ocorrência
GA	<i>Genetic Algorithm</i>	38
GAP	<i>Generalized Assignment Problem</i>	28
GLPK	<i>GNU Linear Programming Kit</i> . Pacote de programação linear regido pelas regras de uso do grupo GNU.	58
GPDP	<i>General Pickup and Delivery Problem</i> . Nome pelo qual o PGCE é mais conhecido na literatura.	4
GRASP	<i>Greedy Randomized and Adaptive Search</i>	31
HGC_DT	Heurística de geração de colunas para minimização da distância total percorrida	3
HGC_NV	Heurística de geração de colunas para minimização do número de veículos	3
M_BETGC	Mutação de Busca Exaustiva por Troca com Ganho de Custo	81
M_ER	Mutação de Eliminação de Rota	76
M_ETCRR	Mutação de Eliminação de Consumidores da Rota Receptora	101
M_EUR	Mutação de Eliminação de Toda a Menor Rota	103
M_EUTC1	Mutação de Eliminação do Último Consumidor da Menor Rota com Troca de Um Consumidor na outra rota qualquer	101
M_EUTC2	Mutação de Eliminação do Último Consumidor da Menor Rota com Troca de Dois Consumidores em outra rota qualquer	102
M_IC	Mutação de Inserção de Consumidor	102
M_INT	Mutação λ -interchange	77
M_INVC	Mutação por Inversão de Consumidores	79
M_PFIHP	Mutação PFIH parcial	79
M_RC	Mutação de Retirada de Consumidor	77
M_RCRR	Mutação por Retirada de Consumidor da Rota Receptora	100
M_RDT	Mutação por Reinserção de Consumidores com ganho em DT	80
continua na página seguinte		

continuação da página anterior		
Abreviação	Significado	Primeira ocorrência
M_RDT	Mutação por Reinserção de Consumidores com ganho em DT	101
M_TC	Mutação por Troca de Consumidores em Rotas Distintas	78
M_TCDT	Mutação por Troca de Consumidores com Redução da Distância Total Percorrida	79
M_TCFIH	Mutação de Retirada e Troca de Consumidores seguida do PFIH	81
M_TCRAdap1	Mutação por Troca de Consumidores com Redução de Adap_DETCMR, Adap_DEUCMR e Adap_DEUCRR	99
M_TCRAdap2	Mutação por Troca Simples de Consumidores com Redução de Adap_DETCMR, Adap_DEUCMR	99
M_TSC	Mutação por Troca Simples de Consumidores em Rotas Distintas	78
M_TSCDT	Mutação por Troca Simples de Consumidores com Redução da Distancia Total Percorrida	79
M_TSCMR	Mutação por Troca Simples de Consumidores na Mesma Rota	98
NP	Non-deterministic Polinomial	12
NV	Número de Veículos	27
PCE	Problema de Coleta e Entrega	4
PCV	Problema do Caixeiro Viajante	6
PDP	<i>Pickup and Delivery Problem</i> . Nome pelo qual o PCE é mais conhecido na literatura.	4
PECMRJTC	Problema Elementar do Caminho Mínimo com Restrição de Janela de Tempo e Capacidade	16
PFIH	<i>Push-Forward Insertion Heuristic</i> . Heurística construtiva de inserção de consumidores proposta por Solomon [SM87]	66
PGCE	Problema Geral de Coleta e Entrega	4
PPC	Problema de Particionamento de Conjuntos	17
PRV	Problema de Roteamento de Veículos	4
PRVC	Problema de Roteamento de Veículos Capacitado	4
continua na página seguinte		

continuação da página anterior		
Abreviação	Significado	Primeira ocorrência
PRVDJT	Problema de Roteamento de Veículos Dinâmico com Janela de Tempo	2
PRVJT	Problema de Roteamento de Veículos Estático com Janela de Tempo	1
PRVJTDRT	Problema de Roteamento de Veículos com Janela de Tempo em Tempo Real	50
RTVRPTW	<i>Real Time Vehicle Routing Problem with Time Window</i>	50
SAT	<i>Satisfiability Problem</i> ou Problema da Satisfabilidade	12
SPP	<i>Set Partitioning Problem</i>	17
TPDP	Truckload Pickup and Delivery Problem ou Problema de Coleta e Entrega de Cargas Cheias	7
TSP	Traveling Salesman Problem. Nome pelo qual o Problema do Caixeiro Viajante é mais conhecido	6
VRP	<i>Vehicle Routing Problem</i> . Nome pelo qual o PRV é mais conhecido na literatura.	4
VRPTW	Vehicle Routing Problem Time Window. Nome pelo qual o PRVJT estático é mais conhecido na literatura	5

Lista de Símbolos

Símbolo	Descrição	Primeira ocorrência
A	Arestas de um grafo representando um problema de roteamento	10
a_i	Início da janela de tempo no consumidor i . Representa o momento inicial permitido para um veículo começar o serviço em um consumidor.	11
β	Igual a 0.1, constante calculada empiricamente por Solomon para a fórmula do <i>PFHI</i>	67
$B(s)$	representação de uma solução na proposta de Cordeau [CLM00]	35
β^*	penalidade associada ao número de janelas de tempo violadas, utilizada por Berger [BBB01]	45
b_i	Fim da janela de tempo no consumidor i . Representa o final do período aceitável para um veículo chegar neste consumidor.	11
b_j^i	tempo previsto para o consumidor j ser visitado no indivíduo i , utilizada por Berger [BBB01]	45
C	Conjunto de vértices do grafo \mathcal{G} , incluindo o depósito central (consumidores + depósito central)	11
c_i	Custo de inserção de consumidores propostos por Solomon [SM87] para a heurística <i>PFIH</i>	66
$c_i(r)$	custo mínimo de inserção de i em r , utilizado por Berger [BBB01]	47
$c_i(r^*)$	custo mínimo de inserção de i nas rotas da vizinhança de i , utilizado por Berger [BBB01]	47
c_{ij}	Custo associado a ir do consumidor i ao consumidor j . Neste trabalho é sempre distância percorrida entre os consumidores.	11
c_r	Custo da rota r , ou seja, distância euclidiana total percorrida, partindo do depósito central, passando pelos consumidores da rota r e retornando ao depósito central.	17
$d(s)$	violação de carga de uma solução sna proposta de Cordeau [CLM00]	34
d_{0i}	distância do depósito central ao consumidor i	67
d_i	distância total percorrida do indivíduo i , utilizada por Berger [BBB01]	45

continua na próxima página

continuação da página anterior		
Símbolo	Descrição	Primeira ocorrência
δ_{ri}	O parâmetro δ_{ri} é igual a 1 (um) se o consumidor i é atendido pela rota r e 0 (zero) caso contrário	17
D_k	Menor tempo de retardo causado aos consumidores de uma rota qualquer, quando inserimos o consumidor k na mesma	42
d_m	média da distância total percorrida dos indivíduos da primeira geração, utilizada por Berger [BBB01]	45
D_R	Soma dos mínimos tempo de retardos para todos os consumidores de uma rota	42
$f(s)$	função custo para definir a qualidade de uma solução para o PRVJT na proposta de Cordeau [CLM00]	34
\mathcal{G}	Grafo representando um problema de roteamento	11
i	Um consumidor qualquer; ou vértice do grafo	11
I	Um conjunto de pontos distribuídos no espaço multidimensional, representando a "região" viável do PRVJT original, incluindo as restrições de integralidade	23
I	Conjunto de soluções produzidas para o PRVJT conforme proposta de Rochat [RT95]	32
j	Um consumidor qualquer; ou vértice do grafo	11
K	Constante suficientemente grande, para permitir a linearização de uma restrição do modelo compacto do PRVJT	11
λ_i	multiplicadores de lagrange relacionados a restrições impostas ao atendimento do consumidor i	15
l_j	tempo limite para visitar o consumidor j , utilizada por Berger [BBB01]	45
K_1 e K_2	Duas constantes positivas, sendo $K_1 \gg K_2$	90
M	Um número de indivíduos selecionados para participar do torneio	72
N	Número de consumidores, excluindo-se o depósito central	10
$N(S)$	Vizinhança, ou seja, conjunto de soluções possíveis a partir de uma população de indivíduos S	61
$N(s)$	Vizinhança de uma solução única s	35
N_r	Número de consumidores da rota r .	95
NV	Número de veículos	90

continua na próxima página

continuação da página anterior		
Símbolo	Descrição	Primeira ocorrência
P	Um percentual da população que será renovada por cruzamentos (recombinação de indivíduos)	71
π	variáveis duais do PRVJT por particionamento de conjuntos	20
$p(s^*)$	penalidade de uma solução s^* vizinha de s , conforme trabalho de Cordeau [CLM00]	35
p_i	ângulo da coordenada polar do consumidor i , referente ao depósito central	67
Q	Capacidade dos veículos	10
$q(s)$	tempo total de viagem de uma solução utilizado na proposta de Cordeau [CLM00]	34
q_i	Demanda de carga associada à coleta ou à entrega em um consumidor i	11
R	Conjunto de todas as rotas viáveis para o PRVJT	55
\mathbf{R}	Politipo convexo que envolve a região I , definido pelas restrições do PRVJT sem as restrições de integralidade	23
r	Uma rota viável para o PRVJT	17
R'	Conjunto com parte das rotas viáveis para o PRVJT	55
r_i	Número de rotas no indivíduo i na proposta de Berger [BBB01]	45
r_j	Redução de custo relativo de uma variável não básica j (rota j)	20
r_m	limite inferior para número de rotas (relação da demanda total sobre a capacidade do veículo), utilizada por Berger [BBB01]	45
$RN(i)$	rotas da vizinhança do consumidor i , utilizado por Berger [BBB01]	47
R^s	representa as rotas de um indivíduo solução s .	90
s	Uma solução qualquer para o PRVJT	61
S	Conjunto de rotas selecionadas em T na proposta de Rochat [RT95]	32
s^*	Melhor indivíduo de uma população S	61
s_{iv}	Tempo de chegada do veículo v no consumidor i	11
ss_{ij}	variável de economia na união de duas rotas, através do consumidor final da primeira i com o inicial da segunda j	28
T	Conjunto de rotas presentes nas soluções I da proposta de Rochat [RT95]	32

continua na próxima página

continuação da página anterior		
Símbolo	Descrição	Primeira ocorrência
t_{ij}	Tempo de trajeto do consumidor i para o consumidor j	11
ts_i	Tempo de serviço necessário ao atendimento do consumidor i	11
V	Número de veículos idênticos+B12	11
\mathcal{V}	Conjunto de veículos idênticos, onde $ \mathcal{V} = V$	11
v	Um veículo do conjunto \mathcal{V}	10
$Viol_i$	número de janelas de tempo violadas pelo indivíduo i , utilizada por Berger [BBB01]	45
$w(s)$	violação no tempo total de retorno ao depósito central de uma solução s na proposta de Cordeau [CLM00]	34
w_i	tempo de espera em um consumidor, pela proposta de Cordeau [CLM00]	34
x_{ijv}	Variável de decisão (0 ou 1) na formulação compacta do PRVJT, representando se um veículo v percorre (1) ou não (0) o arco que vai do consumidor i para o consumidor j .	10
x_r	Variável de decisão (0 ou 1) no PPC relacionada a utilização ou não de uma rota r na solução. O valor 1 (um) significa que a rota é utilizada na solução, 0 (zero) caso contrário.	17
Z_D	Mínimo do problema de Lagrange ou função dual de Lagrange	15
Z_{LD}	Problema de maximização do dual de Lagrange, também conhecido como lagrangeano dual	15
α	Igual a 0.7, constante calculada empiricamente por Solomon para a fórmula do <i>PFHI</i>	66
α_j^*	penalidade associada a violação da janela temporal do consumidor j , utilizada por Berger [BBB01]	45
α', β' e γ'	parâmetros positivos utilizados empiricamente para definir uma função custo relativo a cada solução s , na proposta Cordeau [CLM00]	34
γ	Igual a 0.2, Constante calculada empiricamente por Solomon para a fórmula do <i>PFHI</i>	66
η	Número de iterações do algoritmo de Cordeau [CLM00]	36

Lista de Figuras

Figuras	Descrição	Página
2.1	Cruzamento de indivíduos, gerando dois outros descendentes	39
2.2	Processo de recombinação de três indivíduos, seguida de mutação, gerando um descendente.	42
3.1	Representação de uma vizinhança “pobre” de um indivíduo solução do AG	61
3.2	Representação de uma vizinhança mais promissora de um indivíduo solução no AG	61
3.3	Representação do indivíduo no AG_DT	65
3.4	Visualização do processo de seleção por roleta giratória	70
3.5	Processo de escalonamento das adaptabilidades para aumentar ou diminuir a pressão seletiva durante o processo de escolha de indivíduos	71
3.6	Visualização das etapas para gerar uma nova população, enfatizando o elitismo	73
3.7	Diagrama esquemático da Mutação de Eliminação de Rota (M_ER)	76
3.8	Diagrama esquemático da mutação M_RC	77
3.9	Diagrama esquemático da mutação M_IC	77
3.10	Diagrama esquemático da mutação M_TSC	78
3.11	Diagrama esquemático da mutação M_TC	78
3.12	Diagrama esquemático da mutação M_PFIHP	80
3.13	Diagrama esquemático da mutação M_RDT	80
3.14	Diagrama esquemático da mutação M_TGC	81
3.15	Diagrama do Algoritmo Genético implementado AG_DT	82
continua na página seguinte		

continuação da página anterior		
Figuras	Descrição	Página
3.16	Solução do AG_DT atual para o problema completo, as rotas aleatoriamente selecionadas e o problema reduzido gerado	85
3.17	Diagrama completo da solução utilizada para resolver o PRVJT (HGC_DT).	87
4.1	Diagrama esquemático da mutação M_TSC, aplicada em AG_NV	98
4.2	Diagrama esquemático da mutação M_TSCMR	99
4.3	Diagrama esquemático da mutação M_TCRAdap1	99
4.4	Diagrama esquemático da mutação M_TCRAdap2	100
4.5	Diagrama esquemático da mutação M_RCRR	101
4.6	Diagrama esquemático da mutação M_RDT	101
4.7	Diagrama esquemático da mutação M_ETCRR	101
4.8	Diagrama esquemático da mutação M_EUTC1	102
4.9	Diagrama esquemático da mutação M_EUTC2	102
4.10	Diagrama esquemático da mutação M_EUR	103
4.11	Diagrama completo de HGC_NV utilizado para resolver o PRVJT estático com o objetivo (NV, DT), nesta ordem	104
5.1	Diagrama da solução de HGC_DT adaptada ao problema dinâmico	114
6.1	Disposição espacial dos consumidores dos problemas da classe R1	117
6.2	Disposição espacial dos consumidores dos problemas da classe C1	118
6.3	Disposição espacial dos consumidores dos problemas da classe RC	119
8.1	Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema C201	155
continua na página seguinte		

continuação da página anterior		
Figuras	Descrição	Página
8.2	Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema R101	156
8.3	Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema R201, com 30 consumidores informados após 45 minutos de execução	157
8.4	Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema R201, com 50 consumidores informados após 45 minutos de execução	157
8.5	Resultado médio de HGC_DT (3 execuções), variando o número de consumidores	158
8.6	Resultados médios de HGC_DT e médios de AG_DT (re-otimização) com a variação do intervalo de inclusão de novos consumidores	159

Índice

Resumo	ii
Abstract (in English).....	iv
Lista de Abreviações	vi
Lista de Símbolos	x
Lista de Figuras	xiv
Índice	xvii
Capítulo 1	1
Introdução.....	1
1.1 MOTIVAÇÃO.....	1
1.2 PRINCIPAIS CONTRIBUIÇÕES DESTA TESE	2
1.3 ORGANIZAÇÃO DO TEXTO	3
Capítulo 2	4
Problemas de Roteamento de Veículos com Janela de Tempo	4
2.1 INTRODUÇÃO	4
2.2 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO ESTÁTICO	9
2.2.1 <i>Modelo matemático para o PRVJT</i>	10
2.2.2 <i>Complexidade do problema</i>	12
2.2.3 <i>Trabalhos correlatos utilizando Métodos Exatos</i>	15
2.2.4 <i>Trabalhos correlatos utilizando Métodos Heurísticos</i>	25
2.3 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS DINÂMICO COM JANELA DE TEMPO (PRVDJT)	48
2.3.1 <i>Modelo em tempo real para PRV dinâmico</i>	50
2.3.2 <i>Outros pontos críticos do roteamento dinâmico</i>	51
2.3.3 <i>Métodos Exatos para o Problema de Roteamento de Veículos Dinâmico</i>	52
Capítulo 3	54
Minimizando a Distância Percorrida no PRVJT Estático	54
3.1 INTRODUÇÃO	54
3.2 O MODELO DE PARTICIONAMENTO DE CONJUNTOS PARA O PRVJT	56
3.3 UMA HEURÍSTICA DE GERAÇÃO DE COLUNAS UTILIZANDO O ALGORITMO GENÉTICO	57
3.3.1 <i>Preprocessamento</i>	58
3.3.2 <i>Encontrando mínimos locais</i>	58
3.3.3 <i>Gerando os primeiros mínimos locais</i>	83
3.3.4 <i>Melhorando os mínimos locais através do problema reduzido</i>	83
3.3.5 <i>Encontrando soluções próximas ao ótimo global</i>	85
3.4 CONCLUSÕES	88
Capítulo 4	89
Minimizando o Número de Veículos no PRVJT Estático.....	89
4.1 INTRODUÇÃO	89
4.2 REDUZINDO O NÚMERO DE VEÍCULOS	90
4.2.2 <i>Atribuição de adaptabilidades hierárquicas aos indivíduos</i>	91
4.2.3 <i>Seleção por Torneio Múltiplo</i>	96

4.2.4	<i>Operadores para minimização de veículos</i>	97
4.3	ENCONTRANDO SOLUÇÕES EM NÚMERO DE VEÍCULOS PARA O PRVJT ESTÁTICO	103
4.4	CONCLUSÕES	105
Capítulo 5		106
Minimizando a Distância Percorrida no PRVJT Dinâmico.....		106
5.1	INTRODUÇÃO	106
5.2	DEFININDO AS VARIÁVEIS DINÂMICAS	107
5.3	MODIFICANDO O MACROFLUXO DE HGC_DT	111
5.4	CONCLUSÕES	114
Capítulo 6		115
Resultados Minimizando a Distância Percorrida no PRVJT Estático.....		115
6.1	O CONJUNTO DE PROBLEMAS TESTES DE SOLOMON	115
6.2	AVALIANDO A HEURÍSTICA HGC_DT PARA MINIMIZAÇÃO DE DISTÂNCIA TOTAL PERCORRIDA..	119
6.2.1	<i>Configuração dos parâmetros de HGC_DT</i>	120
6.2.2	<i>Resultados</i>	127
6.3	CONCLUSÕES	136
Capítulo 7		137
Resultados Minimizando Número de Veículos no PRVJT Estático		137
7.1	CONFIGURANDO OS PARÂMETROS EM AG_NV E HGC_NV	137
7.2	RESULTADOS.....	141
7.3	CONCLUSÕES	144
Capítulo 8		146
Resultados Minimizando Distância no PRVJT Dinâmico		146
8.1	INTRODUÇÃO	146
8.2	DEFININDO OS CENÁRIOS DINÂMICOS A PARTIR DAS INSTÂNCIAS DE SOLOMON.....	147
8.2.1	<i>Chegada de novos consumidores</i>	148
8.2.2	<i>Cancelamento de consumidores</i>	163
8.3	CONCLUSÕES	163
Capítulo 9		165
Conclusões.....		165
9.1	COMPARAÇÃO DOS RESULTADOS DE HGC_DT COM AS SOLUÇÕES EXATAS DA LITERATURA	165
9.2	RESULTADOS DE HGC_DT COM DUPLA PRECISÃO.....	166
9.3	MELHORANDO A DISTÂNCIA PERCORRIDA EM HGC_NV	166
9.4	ESTABELECIMENTO DE “PROBLEMAS TESTES” PARA O PRVDJT.....	167
9.5	DISTÂNCIA PERCORRIDA <i>VERSUS</i> NÚMERO DE VEÍCULOS	167
9.6	LIMITES INFERIORES PARA HGC_DT.....	167
9.7	CONCLUSÕES FINAIS	168
Referências Bibliográficas.....		170
Apêndice A – O Algoritmo AG_PARAM		181
Apêndice B – Novas Soluções Encontradas.....		183

Capítulo 1

Introdução

1.1 Motivação

O Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) tem sido amplamente estudado nas últimas décadas. Existem duas grandes motivações para tal fato. Primeiramente, os custos relacionados ao transporte de pessoas e mercadorias geralmente são elevados, com tendência ao crescimento, motivado pela expansão atual das fronteiras comerciais de todo tipo de negócio. Pesquisas sugerem que de 10% a 15% do valor final das mercadorias comercializadas no mundo correspondem ao custo de seu transporte [FJM97] [The98] [Cas93]. Segundo Bodin [BGA+83], o custo de distribuição nos Estados Unidos em 1980 foi estimado em 400 bilhões de dólares. Segundo Castro [Cas93], no Brasil o valor despedido no transporte de cargas é de 10.8% do Produto Interno Bruto (PIB). Adicionalmente, existem os fatores de poluição e ruído que ultimamente têm se tornado críticos, principalmente nas grandes cidades. Uma parcela destes custos e inconvenientes poderia ser reduzida com o tratamento de diversos problemas de roteamento, onde o PRVJT tem uma importante contribuição. A segunda grande motivação para o estudo do PRVJT está relacionada ao grande desafio que a solução deste problema representa. O PRVJT está conhecidamente na classe NP-difícil, e é um dos problemas mais difíceis em otimização combinatória, representando um grande desafio para os pesquisadores.

Entretanto, os problemas de roteamento tratados na literatura têm se limitado a modelos bem mais simples do que aqueles encontrados no mundo real, muitas vezes com excessiva simplificação. A principal simplificação está na variável tempo. Os problemas do mundo real apresentam variáveis que estão o tempo todo em alteração, tais como novos pedidos, cancelamentos, alteração nos custos previstos nos trajetos, entre outros. Quando não se conhecem todas as variáveis do problema no início da otimização, ou quando estas variáveis sofrem alteração ao longo do tempo, classificasse o problema como dinâmico.

Por outro lado, a variável tempo nunca foi tão importante como nos dias atuais. Atualmente, muitas empresas de distribuição de mercadorias trabalham sob consideráveis restrições de tempo. Isso inclui não só o intervalo de tempo para a entrega nos clientes, o que já é considerado no PRVJT tradicional, mas também o curto período disponível para chegada, carregamento, roteamento e distribuição das mercadorias. Conseqüentemente, um modelo considerando o tempo como variável do problema, vem ao encontro das necessidades atuais.

As poucas propostas atuais para solução do PRVJT dinâmico têm utilizado as mais diferentes formulações, uma vez que não há problemas testes (*benchmarks*) consolidados na literatura tratando-o de forma comum. Com isso, é muito difícil avaliar a qualidade das soluções produzidas em muitos trabalhos da literatura. Surge então a necessidade de uma proposta que obtenha ou supere os melhores resultados nos já consagrados problemas testes estáticos, mas que também possam tratar o desconhecimento parcial das variáveis, ou seja, os problemas dinâmicos.

1.2 Principais contribuições desta tese

Entre as principais contribuições desta tese, está o desenvolvimento e avaliação de uma heurística para o tratamento do PRVJT, capaz de abordar não só os problemas estáticos mais também aqueles dinâmicos, onde parte das variáveis são conhecidas após o início do processo de otimização.

Adicionalmente, buscou-se contribuir para algumas das principais lacunas encontradas na literatura atual para o PRVJT, a saber:

- Uma comparação entre as soluções produzidas por métodos exatos e uma heurística foi possibilitada, graças à padronização de critérios de arredondamento e da função objetivo utilizada.
- A obtenção de uma maior robustez na qualidade dos resultados foi possível graças à utilização de uma abordagem híbrida, utilizando técnicas de programação linear inteira e heurísticas.

- Novos critérios para orientação na direção de busca no espaço de soluções viáveis foram propostos, aumentando a confiabilidade e diminuindo o tempo necessário para redução do número de veículos no PRVJT.
- Resultados inéditos em distância total percorrida foram alcançados para as consagradas instâncias de Solomon do PRVJT.
- Modificações nas instâncias de Solomon tradicionais foram propostas para transformá-las em problemas dinâmicos, sendo possível avaliar de forma mais direta os resultados, comparando-os com aqueles alcançados na literatura para o problema estático.

1.3 Organização do texto

O capítulo 2 apresenta o estudo das principais propostas de solução para o PRVJT, incluindo métodos exatos e heurísticas. Um resumo das principais abordagens ao problema dinâmico também é feito neste capítulo. O capítulo 3 apresenta uma proposta de solução para o PRVJT estático, mas que já foi concebida objetivando também o problema dinâmico. Esta proposta consiste de uma heurística de geração de colunas baseada no algoritmo genético (HGC_DT). No capítulo 4, alterações são realizadas, a fim de permitir uma solução semelhante, mas desta vez com a prioridade na minimização do número de veículos (HGC_NV). No capítulo 5, ligeiras modificações são definidas em HGC_DT para possibilitar a abordagem do problema dinâmico, novamente minimizando distância total percorrida. O capítulo 6 apresenta os resultados obtidos para HGC_DT, e suas comparações com os principais trabalhos anteriores, utilizando dois conjuntos de problemas testes consagrados na literatura. No capítulo 7 os resultados para a minimização do número de veículos utilizando HGC_NV são apresentados. No capítulo 8 os resultados para o PRVJT dinâmico são apresentados, e finalmente, no capítulo 9 as conclusões desta tese.

CAPÍTULO 2.....	4
PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO	4
2.1	INTRODUÇÃO 4
2.2	O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO ESTÁTICO 9
2.2.1	<i>Modelo matemático para o PRVJT.....</i> 10
2.2.2	<i>Complexidade do problema.....</i> 12
2.2.3	<i>Trabalhos correlatos utilizando Métodos Exatos.....</i> 15
2.2.4	<i>Trabalhos correlatos utilizando Métodos Heurísticos</i> 25
2.3	O PROBLEMA DE ROTEAMENTO DE VEÍCULOS DINÂMICO COM JANELA DE TEMPO (PRVDJT) 48
2.3.1	<i>Modelo em tempo real para PRV dinâmico.....</i> 50
2.3.2	<i>Outros pontos críticos do roteamento dinâmico.....</i> 51
2.3.3	<i>Métodos Exatos para o Problema de Roteamento de Veículos Dinâmico.....</i> 52

Capítulo 2

Problemas de Roteamento de Veículos com Janela de Tempo

Este capítulo apresenta uma revisão bibliográfica sobre o PRVJT estático e dinâmico e as principais metodologias empregadas para sua solução. Na seção 2.1 os diversos problemas de roteamento são descritos, contextualizando o problema a ser tratado nesta tese. Na seção 2.2 o PRVJT estático é abordado, resumindo os principais trabalhos utilizando métodos exatos e heurísticas. Na seção 2.3 é feita uma revisão do tratamento dado ao PRVDJT, ou seja, o mesmo problema anterior, mas considerando variáveis dinâmicas.

2.1 Introdução

Entende-se por roteamento de veículos uma ampla gama de problemas relacionados à escolha de rotas a serem percorridas por veículos, objetivando melhorar o custo relacionado a entregas e ou coletas de mercadorias. Uma generalização bem ampla é a formulação do chamado Problema Geral de Coleta e Entrega (PGCE), *General Pickup and Delivery Problem* (GPDP) [SS95]. O PGCE consiste na escolha de rotas para veículos de uma frota, a fim de atender um conjunto de encomendas. Cada encomenda deve ser coletada em uma localidade e entregue em outra. Os veículos não estão necessariamente concentrados em um depósito ou garagem central, mas espalhados em qualquer localidade da região. Cada veículo tem um ponto de partida e um ponto de destino e durante a sua rota entre origem-destino, atenderá a um conjunto de coletas e entregas.

Savelsbergh [SS95] considera três problemas como casos particulares do PGCE: o Problema de Coleta e Entrega (PCE) ou *Pickup and Delivery Problem* (PDP), onde cada encomenda é coletada em uma única origem e possui um único destino, cada veículo deve retornar ao seu ponto de partida no final da rota; o *Dial-a-Ride Problem* (DARP), que ainda é um caso especial do PCE, onde a carga transportada são pessoas; e o Problema de

Roteamento de Veículos (PRV) ou *Vehicle Routing Problem* (VRP), onde todos os pontos de partida e destino final são idênticos para todos os veículos, um depósito central.

Similarmente, Xu [XCR+01] considera que os problemas de roteamento são basicamente focados no seguinte objetivo: “uso eficiente de uma frota de veículos que deverão coletar e entregar encomendas de mercadorias”. Adicionalmente, Xu enfatiza os principais problemas como sendo: o Problema de Roteamento de Veículos Capacitado (PRVC), *Capacitated Vehicle Routing Problem* (CVRP), que considera uma frota uniforme de veículos de capacidade limitada, localizados inicialmente no mesmo depósito e sem limitação de tempo na entrega; o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT), *Vehicle Routing Problem with Time Windows* (VRPTW), que é uma generalização do PRVC, incluindo uma janela de tempo como intervalo obrigatório para começar o atendimento do cliente; e o Problema de Coleta e Entrega (PCE), onde as encomendas dos clientes podem ser divididas em duas partes, a coleta em um local e a entrega em outro. Neste último caso as classificações em [XCR+01] e [SS95] são idênticas.

Gendreau [GP97] faz uma classificação dos problemas de despacho de veículos que inclui problemas que não necessariamente requerem roteamento. Sua classificação avalia inicialmente a necessidade ou não do roteamento. O autor considera roteamento como o planejamento seqüencial de diversas rotas. A Tabela 2.1 apresenta alguns exemplos de problemas e a classificação atribuída pelo autor.

Tabela 2.1: Classificação dos problemas de despacho de veículos quanto à necessidade de roteamento e abrangência da área de cobertura do serviço, segundo Gendreau [GP97].

	Área Restrita	Área Irrestrita
Com necessidade de roteamento	Entrega domiciliar	Transportadoras de carga fracionada
	Transporte de passageiros	
Sem necessidade de roteamento	Serviços de emergência (ambulância, polícia, bombeiro)	Transportadoras de carga cheia (um destino por viagem)

Gendreau [GP97] apresenta ainda outra possibilidade de classificação dos problemas que envolvem roteamento de veículos usando dois critérios: a existência ou não da capacidade de carga e o número de pontos de coleta e entrega, conforme Tabela 2.2.

Tabela 2.2: Classificação dos Problemas de Roteamento quanto à restrição de capacidade e o número de pontos de coleta e entrega de encomendas, segundo Gendreau [GP97].

	Muitos-para-muitos	Um-para-muitos
Com limite de capacidade	Transporte de passageiros	Entrega expressa de comida
Sem limite de capacidade	Serviço de correio (cartas)	Serviço de reparo

A classificação adotada por Gendreau [GP97] objetiva encontrar semelhanças na modelagem dos diversos problemas de roteamento e dos métodos de solução utilizados. O autor mostra que apesar das variações básicas para a adaptação a um modelo específico, os trabalhos têm seguido uma estrutura comum: “baseada em uma soma ponderada de uma variedade de componentes relacionados ao custo do sistema de despacho: distância total percorrida, capacidade utilizada, taxa de saída (throughput) e a qualidade dos serviços para os clientes”. Uma exceção está em [SPR+95] que propõe o despacho baseado em um sistema de aprendizagem.

Observa-se que os diversos problemas de roteamento e despacho de veículos sugeridos na literatura, PGCE, PCE, PRV, PRVC, Despacho de Veículos, DARP, entre outros, representam diferentes variantes de uma grande classe de problemas. Acredita-se que a generalização mais abrangente, capaz de contemplar muitas destas variantes, é o Problema Geral de Coleta e Entrega, PGCE, apresentado por Savelsbergh [SS95] como *General Pickup and Delivery Problem*. Entretanto, este fato não exclui a necessidade de tratar cada um de forma específica, com um modelo próprio, pois estes complexos problemas matemáticos geralmente requerem alto grau de especialização da técnica abordada. Conseqüentemente, uma avaliação dos diferentes tipos de métodos e seus resultados se faz necessária. Cada tipo de problema ou “subclasse” do PGCE em particular pode apresentar possibilidades diferentes quanto à utilização de determinadas metodologias de solução. Um exemplo é o amplamente conhecido Problema do Caixeiro Viajante (PCV) ou *Traveling Salesman Problem* (TSP). Este também é uma particularização do PGCE. Sob este enfoque de roteamento de veículos, no PCV só existe um veículo, não existe limitação de capacidade ou de tempo, todas as coletas são feitas na localidade de partida e existe uma entrega para cada localidade. Para o PCV já é viável a utilização de métodos exatos para

instâncias relativamente elevadas.. No entanto, outros casos especiais do PGCE, como o Problema de Roteamento de Veículos Capacitado a utilização de tais métodos já não são capazes de gerar soluções em tempo aceitável, mesmo com o número de consumidores bem menor comparado às utilizadas para o PCV.

Perante os objetivos deste trabalho, que concentra nos problemas de roteamento com múltiplos veículos, com restrição de capacidade e obedecendo uma janela de tempo em cada consumidor, as quatro classes de problemas apresentadas na Tabela 2.3 são de grande importância, pois sumarizam os principais problemas de roteamento correlatos, com ampla abordagem na literatura atual.

Tabela 2.3: Classificações dos problemas de roteamento, considerando as diferenças principais no âmbito deste trabalho: dinamicidade e tipo básico de serviço.

Quanto ao dinamismo Tipo de Serviço	Estático	Dinâmico
Problema de Coleta e Entrega (PCE)	(1) Transporte escolar com várias escolas	(2) Distribuição de dinheiro de várias agências para vários caixas automáticos 24 horas
Problema de Roteamento de Veículos (PRV)	(3) Entrega dos correios	(4) Entrega de bebidas a partir de pedidos a uma central

Primeiramente, é importante diferenciar o PRV do PCE. A comunidade de pesquisa operacional não tem empregado estes nomes de forma padronizada. Há quem considera o PRV como um caso específico do PCE [SS95], e há também o contrário [XCR+01]. Savelsbergh [SS95] considera que o PRV é um caso específico do Problema de Coleta e Entrega, onde todas as coletas são feitas em um depósito central. No entanto, Yang [YJM00] considera o que ele chamou de *Truckload Pickup and Delivery Problem (TPDP)*, que é um caso de coleta e entrega de uma carga cheia (todo o caminhão é carregado em um único ponto de origem com um destino também único), como sendo um caso específico do PRV. Conclui-se que se trata apenas de diferentes convenções adotadas por diferentes grupos de trabalho e não foi possível ainda chegar a um padrão consensual. O mais importante é que todos estes trabalhos apresentam diferentes variantes, que poderiam ser

generalizadas no Problema Geral de Coleta e Entrega (PGCE), e se assemelham muito. Entretanto existem diferentes abordagens na tentativa de obtenção da solução. Conseqüentemente, existem muitas possibilidades de metodologias, modelagens e algoritmos testados e disponíveis na literatura, seja qual for a variante do problema de roteamento em questão.

No contexto deste trabalho, considera-se como um Problema de Roteamento de Veículos (PRV) quando existe a necessidade de levar a mercadoria a um depósito central antes de ser entregue ao seu destino, ou, opcionalmente, quando são realizadas coletas em diferentes fornecedores a fim de centralizar as mercadorias em um depósito central. Isto é, no PRV, seja na situação onde são realizadas coletas ou entregas, considera-se uma restrição do tipo: a origem de partida e o destino final de todos os veículos devem ser a mesma. Já para o Problema de Coleta e Entrega (PCE) cada pedido tem independentemente uma origem e um destino qualquer. Além disso, os veículos não estão confinados inicialmente em um único local, podendo estar em qualquer vértice do grafo que representa o cenário do problema. Esta será a diferença considerada entre as classes de Problemas de Roteamento de Veículos (PRV) e Problema de Coleta e Entrega (PCE) no restante deste texto. Conseqüentemente, a expressão Problema de Roteamento de Veículos neste trabalho se refere a esta classe de problemas mais restrita, conforme foi descrito. Já a expressão problemas de roteamento de veículos, no plural, deve ser entendida como uma classe de problemas ainda mais ampla que o Problema Geral de Coleta e Entrega, que inclui o PRV, o PCE e ainda outros.

Embora o PRV e o PCE possuam muitas características em comum, existem algumas bem diferentes, que devem ser observadas. Por exemplo, podemos citar o padrão geográfico da solução encontrada no PRV: são regiões mais ou menos distintas e mais bem definidas sendo atendidas pelo mesmo veículo. Já no PCE, as soluções têm padrões totalmente diferentes, podendo um mesmo veículo percorrer uma rota de um extremo a outro da região atendida. Esta e outras diferenças são importantes e devem ser consideradas pelos diferentes métodos de solução propostos. Uma heurística, por exemplo, que faz uso da repartição inicialmente do espaço em sub-regiões, como é o caso de propostas baseadas no Método de Pétalas (*Petal Method*), pode ter um ótimo desempenho em um tipo de problema e um péssimo no outro.

Na Tabela 2.3 as classes pares se distinguem das ímpares pelo critério dinamicidade do problema. No mundo real a grande maioria dos problemas são dinâmicos. No entanto, por ser extremamente complicado tratar a dinamicidade, muitos dos modelos adotados na literatura desconsideram estes fatores, utilizando apenas um cenário estático.

2.2 O Problema de Roteamento de Veículos com Janela de Tempo Estático

O Problema de Roteamento de Veículos Estático, mais conhecido simplesmente como *Vehicle Routing Problem* ou *VRP*, foi introduzido por Dantzig e Ramser [DR59] em 1959 com o artigo intitulado “*The Truck Dispatching Problem*”. Em um depósito central ou garagem, utilizam-se um número de veículos idênticos V , com capacidade máxima uniforme Q , para atender a um conjunto de demandas, representadas por quantidades q_i , onde i é um dos N consumidores, $i \in C = \{0, \dots, N+1\}$, sendo 0 e $N+1$ um mesmo depósito central, onde partem e chegam os veículos. Cada trecho entre dois clientes i e j , possui um custo simétrico associado $c_{ij} = c_{ji}$, para $0 \leq i, j \leq N+1$. Atualmente, esta versão do PRV é mais comumente associada ao PRVC já citado na seção anterior, ou do inglês, *CVRP*, pois no referido trabalho o autor já considerou a restrição de capacidade do veículo. Uma generalização do PRVC é o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) ou *Vehicle Routing Problem with Time Window* (VRPTW) que, além da limitação de capacidade, inclui a restrição do intervalo de tempo para atendimento: o veículo tem que chegar a um consumidor dentro de um intervalo de tempo, denominada janela de tempo, *Time Window*. Normalmente, considera-se permitido ao veículo chegar antes do horário previsto em um consumidor, mas será necessário esperar o momento de iniciar o serviço, ou seja, esperar a “abertura” da janela de tempo para a carga ou descarga das encomendas. Todo veículo tem um tempo de serviço, que somente depois de transcorrido poderá partir para outro consumidor. Em alguns casos, costuma-se relaxar a janela de tempo, permitindo o início do serviço antes da abertura da janela ou após o fechamento da janela de tempo, somando-se um custo adicional por esta violação.

2.2.1 Modelo matemático para o PRVJT

O problema de roteamento de veículos com janela de tempo estático pode ser formulado como se segue. Um conjunto de V veículos idênticos, representado pelo conjunto $\mathcal{V}=\{1,\dots,V\}$, necessitam realizar entregas ou coletas em uma região, tendo como ponto de partida um único depósito central. Os N consumidores dentro desta região mais o depósito central estão representados pelo conjunto \mathcal{C} , que são vértices de um grafo $\mathcal{G}=(\mathcal{C},A)$ e A o conjunto de arestas do grafo, representando as distâncias entre os consumidores. Normalmente, para facilitar a modelagem utiliza-se dois vértices do grafo \mathcal{G} para representar o depósito central, o vértice 0 e o vértice $N+1$. No modelo, que será descrito a seguir, o vértice 0 representa o local de onde os veículos deverão iniciar suas rotas. Já o vértice $N+1$ representa o local onde todos os veículos deverão terminar suas rotas. As variáveis t_{ij} e c_{ij} representam respectivamente o tempo e a distância necessários para ir do vértice i para o vértice j . Cada consumidor i também está associado a uma demanda, ou seja, uma quantidade de encomenda q_i . Além disso, cada consumidor deverá ser atendido por um único veículo, não sendo permitido a divisão de uma encomenda por dois ou mais veículos. Quanto à janela de tempo, definida como o intervalo $[a_i,b_i]$, indica que a partir do instante inicial a_i é permitido o início da entrega ou coleta no consumidor i . Caso a chegada do veículo no consumidor i se dê antes do instante a_i o veículo deverá esperar. O veículo nunca poderá chegar depois do instante b_i , pois viola a restrição de tempo do problema. Este tipo de restrição de tempo é conhecido na literatura como janela de tempo rígida ou *hard time window*. Os veículos são idênticos e possuem uma capacidade máxima de carga Q . Os veículos irão consumir um tempo de serviço atendendo cada consumidor i igual a ts_i .

Pode-se agora definir o modelo matemático para o problema de otimização como

$$\text{Minimize } \sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} c_{ij} x_{ijv} \quad \text{sujeito à} \quad (2.1)$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{C}} x_{ijv} = 1 \quad \forall i \in \mathcal{C} \quad (2.2)$$

$$\sum_{i \in \mathcal{C}} q_i \sum_{j \in \mathcal{C}} x_{ijv} \leq Q \quad \forall v \in \mathcal{V} \quad (2.3)$$

$$\sum_{j \in C} x_{0jv} = 1 \quad \forall v \in \mathcal{V} \quad (2.4)$$

$$\sum_{i \in C} x_{ihv} - \sum_{j \in C} x_{h jv} = 0 \quad \forall h \in C, \forall v \in \mathcal{V} \quad (2.5)$$

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

$$s_{iv} + ts_i + t_{ij} - K(1 - x_{ijv}) \leq s_{jv} \quad \forall i, j \in C; \forall v \in \mathcal{V} \quad (2.7)$$

$$a_i \leq s_{iv} \leq b_i \quad \forall i \in C; \forall v \in \mathcal{V} \quad (2.8)$$

$$x_{ijv} \in \{0,1\} \quad \forall i, j \in C; \forall v \in \mathcal{V} \quad (2.9)$$

A equação (2.1) representa a soma de todos os custos associados a cada arco (i, j) utilizado na solução, percorrido por um determinado veículo v . Neste modelo o custo está associado unicamente a utilização ou não de cada arco, geralmente a distância percorrida do consumidor i para o j . A restrição da equação (2.2) garante que somente um veículo chega a cada consumidor i , ou seja, cada consumidor é atendido por um único veículo. A equação (2.3) garante que cada veículo v atenderá somente um conjunto de consumidores cuja demanda total não ultrapasse a sua capacidade Q . As equações (2.4) e (2.6) garantem, respectivamente, que cada veículo parte do depósito central e que ainda retornam ao mesmo ao final de sua rota. A equação (2.5) garante a continuidade das rotas, ou seja, se um veículo chega a um consumidor ele deverá sair do mesmo para o consumidor seguinte, perfazendo todos os trechos entre consumidores. A equação (2.7) garante que o instante de chegada de um veículo v a um consumidor j (s_{jv}) não poderá ocorrer antes do tempo de chegada no consumidor anterior i (s_{iv}) mais o tempo de serviço no primeiro (ts_i), mais o tempo de percurso no trecho (i, j) que é t_{ij} . É assumido uma velocidade constante tal que o tempo de percurso t_{ij} é igual a distância entre i e j . A constante K sendo suficientemente grande garante que a equação (2.7), ainda linear, seja somente uma restrição efetiva quando x_{ijv} seja igual a 1, ou seja, quando o veículo v percorre a trecho (i, j) . Adicionalmente este instante deve estar dentro da janela de tempo para o consumidor em questão, o que é garantido pela (2.8). Finalmente a equação (2.9) garante a integralidade binária das variáveis do problema.

Observa-se no modelo apresentado, conforme (2.1), um custo associado a cada arco do grafo, geralmente a distância do consumidor i ao consumidor j , sendo neste caso o objetivo a minimização da distância total percorrida.

2.2.2 Complexidade do problema

Um problema é dito tratável¹ se existe um algoritmo cuja complexidade de tempo e espaço seja polinomial, do tipo $O(n^k)$, para resolvê-lo, e intratável caso contrário. Entretanto o desconhecimento de tal algoritmo não significa que ele não exista. Em 1971, Cook [Coo71] deu uma contribuição significativa nesta matéria, estabelecendo a classe de problemas NP-completos. A classe de problemas cuja solução possa ser obtida por um algoritmo polinomial determinístico é chamada P. Muitos problemas, embora não possam ser resolvidos por algoritmos polinomiais determinísticos, uma solução pode ser testada em tempo polinomial, ou opcionalmente, a solução pode ser encontrada com algoritmos polinomiais não-determinísticos. Estes problemas são ditos da classe NP (*Non-deterministic Polynomial*). Cook prova a existência de uma sub-classe de problemas em NP correlacionados entre si, onde existe uma grande chance de serem realmente intratáveis, ou seja, de não existir um algoritmo polinomial determinístico para resolvê-lo. Conseqüentemente, não existe muita expectativa de se obter algum algoritmo exato eficiente capaz de resolvê-lo em tempo hábil, para instâncias de tamanho elevado. Atualmente, a complexidade dos algoritmos conhecidos para resolvê-los são exponenciais $O(k^n)$, fatoriais $O(n!)$, mas nunca polinomiais. Os problemas dessa classe contida em NP são chamados NP-completos. O primeiro problema NP-completo, provado por Cook, é o Problema da Satisfabilidade ou SAT, que consiste em dizer se existe ou não uma combinação de estados (sim/não) para variáveis de uma expressão booleana normal conjuntiva, que a tornem verdadeira. Cook provou que se houver um algoritmo determinístico capaz de resolver o SAT em tempo polinomial então $P=NP$. Através de transformações em tempo polinomial, o SAT já foi reduzido a uma série de outros problemas, como o Problema do Caixeiro Viajante, o Problema de Coloração de Grafos, o Problema da Clique, e muitos outros. Conseqüentemente, se algum destes problemas forem

¹ Não há uma definição firme do termo intratável. Aqui, se refere a problemas onde não existe (até o momento) algoritmo polinomial para resolvê-los.

resolvidos por algoritmos polinomiais, o SAT também estará resolvido indiretamente, resultando $P=NP$. Por isso, acredita-se que dificilmente tal algoritmo exista.

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

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

Obter a solução para o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) implica em obter simultaneamente a solução de vários problemas NP-difíceis, tais como o Problema do Caxeiro Viajante (PCV) e o Problema da Mochila, sendo por isso, também NP-difícil. Mesmo encontrar uma simples solução viável para o PRVJT, dispondo de um conjunto limitado de veículos é NP-difícil no sentido forte [Koh95]. Se o número de veículos é ilimitado, uma solução inicial viável é trivial, bastando atender cada consumidor com um veículo.

Conseqüentemente, as expectativas sobre os algoritmos exatos não devem ultrapassar a solução de instâncias de ordem reduzida. Embora a ordem máxima possível hoje seja ligeiramente diferente de alguns anos atrás; atualmente muitas instâncias de 100 consumidores já foram resolvidas enquanto a duas décadas atrás este número não passava de 25; o crescimento da capacidade dos computadores e da eficiência dos algoritmos está distante da curva exponencial representada por este problema NP-difícil. Hoje existem problemas reais que consideram milhares de consumidores. Exceto pelo fato improvável de que podemos ser surpreendidos pela constatação de que $P=NP$, os métodos exatos não são uma alternativa viável para muitos dos problemas reais de hoje.

Por outro lado, heurísticas e algoritmos de aproximação também têm sido estudados e propostos para o PRVJT. A classe de problemas NP-difícil, como não poderia ser diferente, também representa um grande desafio para as heurísticas. Enquanto nos métodos exatos só interessa o ponto ótimo global, para uma heurística é importante um limite superior de qualidade em tempo hábil. Adicionalmente, em uma boa heurística são desejáveis as seguintes características:

- Qualidade da solução em curto intervalo de tempo: deve ser capaz de encontrar soluções próximas ao ótimo, em tempo bem inferior ao necessário pelos métodos exatos disponíveis; o que justificaria sua utilização; e
- Robustez: a qualidade da solução não deve variar demasiadamente com diferentes instâncias de um mesmo tipo de problema, para o qual a heurística foi projetada. E o resultado não deve variar muito em qualidade quando a heurística é aplicada várias vezes para a mesma instância.

Os métodos de aproximação vêm ao encontro destas características almeçadas. Um método de aproximação nada mais é que uma heurística com garantia na qualidade no resultado. Isso é, a melhor solução encontrada por um algoritmo de aproximação está sempre a uma distância percentual previamente definida do ótimo desconhecido. A “distância” do ótimo é particular de cada algoritmo, podendo até mesmo não ser muito relevante em termos práticos. Um exemplo bem conhecido é o algoritmo PRIM, para árvore geradora mínima, que é capaz de oferecer um limite superior (solução viável) para o Problema do Caixeiro Viajante que é no máximo duas vezes o ótimo em distância total percorrida.

Para o PRVJT tem sido mais comum na literatura o desenvolvimento de heurísticas sem a preocupação com a garantia de qualidade (limite inferior), porém avaliadas em sua robustez e qualidade aplicando-as a conjuntos de problemas testes consagrados, como as instâncias de Solomon, 1987 [SM87]. Em [LO95] Laporte oferece uma boa bibliografia das principais heurísticas utilizadas nos problemas de roteamento de veículos.

2.2.3 Trabalhos correlatos utilizando Métodos Exatos

Artigos dedicados a uma avaliação geral das soluções sobre o PRVJT podem ser encontrados em Bodin [BGA+83] [Bod90], Solomon [SM87], Solomon e Desrosiers [SD88], Desrochers [Des88] e Desrosiers [DDS+95]. Na classe de métodos exatos, os resultados mais efetivos têm sido encontrados utilizando um dos seguintes métodos: geração de colunas, relaxação lagrangeana, muitas vezes combinadas e aliadas a modernas técnicas de planos de corte (*cutting planes*), tais como *branch-and-cut* [BKY02], *branch-and-price* e *branch-and-cut-and-price* [FLP2004].

Relaxação Lagrangeana

Diferentes estratégias baseadas em relaxação lagrangeana são encontradas na literatura. Halse [Hal92] apresenta uma solução usando particionamento de variáveis (*variable splitting*), seguida de relaxação lagrangeana, Fisher [FJM97] utiliza uma solução baseada em árvores geradoras mínimas (*K-tree*) e relaxação lagrangeana. Mas talvez os trabalhos mais significativos considerando o sucesso na obtenção de vários ótimos nas consagradas instâncias de Solomon, utilizando relaxação lagrangeana, são Kohl [KM97] e Kallehauge [KLM01].

[KLM01] considera a relaxação sobre a restrição da equação (2.2) do modelo apresentado na seção 2.2.1, que estabelece que cada consumidor deverá ser atendido uma, e somente uma única vez. Para tal são introduzidos multiplicadores de Lagrange λ_i a λ_n , onde λ_i está associado a restrição (2.2) para o consumidor i . Desta forma, o problema relaxado e penalizado fica da seguinte forma, λ_i

$$Z_D(\lambda) = \text{minimize} \sum_{v \in V} \sum_{i \in G} \sum_{j \in G} c_{ij} x_{ijv} - \sum_{i \in G} \lambda_i \left(\sum_{v \in V} \sum_{j \in G} x_{ijv} - 1 \right) \quad (2.10)$$

sujeito as mesmas equações (2.3) a (2.9) apresentadas na seção 2.2.1.

O mínimo do problema de Lagrange, Z_D , é chamado de função dual. Z_D não possui a restrição (2.2) que mantêm as rotas de cada veículo dependente umas das outras, por isso pode ser tratado como diversos problemas independentes, um para cada veículo

$$Z_D(\lambda) = \sum_{v \in \mathcal{V}} Z_v(\lambda) - \sum_{i \in \mathcal{G}} \lambda_i \quad (2.11)$$

onde

$$Z_v(\lambda) = \text{minimize} \sum_{i \in \mathcal{G}} \sum_{j \in \mathcal{G}} (c_{ij} - \lambda_i) x_{ijv} \quad \forall v \in \mathcal{V} \quad (2.12)$$

sujeito equações (2.3) a (2.9) do modelo da seção 2.2.1.

O problema que se apresenta na equação (2.12) é o Problema Elementar do Caminho Mínimo com Restrição de Janela de Tempo e Capacidade (PECMRJTC) ou *Elementary Shortest Path Problem with Time Window and Capacity Constraint (ESPPTCC)* [La99]. Ressalta-se que o custo relativo a cada arco é o custo modificado, ou seja, $\hat{c}_{ij} = c_{ij} - \lambda_i$.

Z_D representa um limite inferior para o problema original a ser utilizado na técnica de poda, e por isso deve ser o mais apertado possível. Para isso, necessita-se encontrar os multiplicadores de Lagrange que maximizem o limite inferior Z_D . Este problema é conhecido como lagrangeano dual.

$$Z_{LD} = \text{maximize} Z_D \quad (2.13)$$

[KLM01] afirma que a utilização de métodos convencionais aplicáveis ao dual Lagrangeano, como subgradiente e *bundle method*, não são competitivos frente a geração de colunas (Dantzig-Wolfe) para o PRVJT. Por isso [KLM01] utiliza um algoritmo de plano de corte em minimização convexa para resolver o problema dual conhecido como *Trust-region*. Este tipo de algoritmo foi inicialmente proposto por Levenberg [Lev44] e Marquardt [Mar63].

Os resultados de [KLM01] foram bem expressivos. O método foi testado nas instâncias dos tipos 1 e 2 de Solomon. Estes conjuntos possuem problemas com 25, 50 e 100 consumidores. O conjunto do tipo 1 contém os subconjuntos R1, RC1 e C1, todos com restrições mais apertadas, com demanda de carga mais próxima ao limite dos veículos e as janelas de tempo menores. Isso deixa o algoritmo de poda com limites mais apertados e, conseqüentemente mais eficiente, conforme afirma o próprio autor. Com isso, no conjunto

do tipo 1, das 87 instâncias existentes foram resolvidas 80. Já no conjunto do tipo 2, que possui 81 instâncias, o número de ótimos encontrados cai para 46. Vale ressaltar que os conjuntos C1 e C2 não chegam a ser um desafio, sendo normalmente resolvidos na raiz da árvore de poda. Dos grupos R2 e RC2 (tipo 2) somente duas instâncias de 100 consumidores foram resolvidas (R201 e RC201). O autor reporta também a média do *gap* relativo as soluções inteiras encontradas e o ótimo da relaxação linear, $(IP_{opt} - LP_{opt}) / IP_{opt}$. Para os conjuntos R1, C1 e RC1 foram 1.2%, 0.2% e 5.3%, enquanto que para os conjuntos R2, C2 e RC2 foram 2.6%, 2.8% e 14% respectivamente.

Geração de Colunas

Formulação do PRVJT por Particionamento de Conjunto

Vários problemas em otimização combinatória podem ser descritos como um Problema de Particionamento de Conjunto (PPC) ou *Set Partitioning Problem (SPP)*. Em se tratando do PRVJT, cada coluna corresponde a uma rota viável candidata a pertencer a solução do problema. As linhas correspondem a cada consumidor, que deverão ser atendidos por uma única rota, uma única vez. O modelo matemático do PRVJT na forma de um PPC é mostrado abaixo:

$$\text{minimize } \sum_{r \in R} c_r x_r \quad \text{sujeito a} \quad (2.14)$$

$$\sum_{r \in R} \delta_{ir} x_r = 1, \quad \forall i \in C \quad (2.15)$$

$$x_r \in \{0,1\} \quad (2.16)$$

A equação (2.14) corresponde a função objetivo. O conjunto R representa todas as rotas possíveis para o problema. A cada rota r existe um custo associado, c_r . O objetivo do problema consiste em encontrar o conjunto de rotas de menor custo, sujeito às restrições do problema. A variável de decisão x_r é binária, conforme (2.16), sendo igual a 1 (um) se a rota r faz parte da solução e 0 (zero) caso contrário. O parâmetro δ_r é igual a 1 (um) se o consumidor i é atendido pela rota r e 0 (zero) caso contrário. Desta forma, a restrição em (2.15) assegura que cada consumidor será atendido por uma única rota, uma única vez.

O modelo do PPC tem sido adotado para tratar problemas de roteamento de veículos com sucesso por diversos autores: Agarwal [AMS89]; Desrosiers, Soumis e Desrochers [DSD+84]; Kohl [Koh95]; e Larsen [La99]. Sendo que [Koh95] e [La99] abordaram especificamente o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT), obtendo novos resultados ótimos para instâncias de Solomon [SM87] até então desconhecidos.

Cordeau [CDD+01] enfatiza a importância do modelo original por fluxo de rede multi-produto do PRVJT, conforme equações (2.1) a (2.9), mesmo quando a estratégia adotada para solução é baseada no método de geração de colunas. Como se sabe, o problema de particionamento de conjuntos (PPC) é o problema principal resultante da aplicação da decomposição de Dantzig-Wolfe no modelo original. Adicionalmente, é no estudo do modelo por fluxo de rede que se percebem as estratégias possíveis de poda e corte e somente depois devem ser “traduzidas” em restrições para o modelo PPC [La99].

No entanto, a utilização direta do modelo de fluxo de redes multi-produto não tem sido utilizada para o PRVJT. Isso acontece basicamente porque existe uma simetria muito grande na árvore de poda necessária na busca da solução inteira do problema. Havendo poucas podas, isto é, todo nó acaba por ser aberto novamente (*branching*), e sendo o espaço de busca muito grande, o algoritmo não termina em tempo hábil. Essas características já foram bem lembradas por Barnhart [BJN98] e algumas delas se encaixam perfeitamente no PRVJT, a saber:

- A formulação compacta do problema apresenta uma relaxação linear fraca. A representação com número elevado de variáveis, como é o caso do PPC, melhora significativamente o limite inferior produzido utilizando o problema relaxado, sem as restrições de integralidade;
- Conforme já mencionado, a árvore de poda resultante na formulação compacta apresenta uma simetria muito grande, fazendo que o desempenho da técnica de poda seja ruim, não alterando em nada os limites após a abertura de cada nó;
- A decomposição da formulação compacta em um problema principal e sub-problemas pode ter uma interpretação física adequada, isso acontece claramente

com o PRVJT, facilitando a consideração de diversas restrições importantes, pois estas são transparentes para o problema principal.

O modelo do problema por particionamento de conjuntos não impõe limitações perante peculiaridades e restrições que podem ser necessárias para gerar uma rota viável, conforme também lembra Xu [XCR+01]. Condições especiais, as mais diversas encontradas no mundo real, podem ser tratadas por um gerador de rotas independente, bastando garantir a viabilidade das rotas e a atribuição do custo de cada uma, c_r , conforme (2.14).

Buscando a solução inteira para o PPC

A busca pela solução inteira, utilizando o modelo das equações (2.14) a (2.16) diretamente, onde todas as rotas viáveis são inseridas no modelo de uma só vez, só é possível para pequenas instâncias. Suponha uma instância que possua 50 consumidores e restrições de carga e janelas de tempo de entrega suficientes para limitar o tamanho das rotas em até 10 consumidores cada. O número de rotas possíveis de n consumidores é o arranjo de 50 elementos n a n , ou seja,

$$\frac{50!}{(50-n)!} \quad (2.17)$$

uma vez que os mesmos consumidores em ordem diferente resultam rotas diferentes com custos diferentes. Como todas as rotas de 1 a 10 consumidores são candidatas, existem

$$\sum_{n=1}^{10} \left(\frac{50!}{(50-n)!} \right) \approx 3,8 \cdot 10^{16} \quad (2.18)$$

rotas possíveis, aproximadamente. As restrições de capacidade de carga dos veículos e janela de tempo podem reduzir significativamente o número de rotas viáveis. Mas o número de rotas cresce muito rapidamente com a ordem do problema e, adicionalmente, conforme descrito na seção 2.2.2 o problema é NP-difícil. Conseqüentemente, se torna uma tarefa impossível obter a solução para o problema de uma só vez, sendo mais conveniente a utilização do método de geração de colunas, conforme será descrito nas seções seguintes. É importante ressaltar que relaxando-se o problema quanto a integralidade das variáveis de decisão, descrito na seção 2.2.1, o problema se torna tratável por pacotes de programação linear disponíveis.

Geração de Colunas com Técnica de Poda (Branch-and-Price)

Muitos dos melhores resultados para o PRVJT na literatura têm sido alcançados utilizando o método de geração de colunas ou também conhecido por decomposição de Dantzig-Wolfe [La99] [KDM+99] [CR99]. O princípio de decomposição de Dantzig-Wolfe é, na realidade, uma generalização do método de geração de colunas. Como o PPC possui um número de variáveis muito grande, no caso do PRVJT se trata de todo o conjunto de rotas viáveis, impossível de ser tratado de uma só vez, inicia-se o problema com um pequeno conjunto de rotas. Na formulação de programação linear isso equivale a supor que todas as demais variáveis estão fora da base, ou seja, iguais a zero. Adicionalmente, considera-se também o problema sem a restrição de integralidade das variáveis de decisão. [La99] atribui o nome de *problema linear reduzido relaxado* ao problema neste estágio da solução. A partir deste ponto, inicia-se um processo de busca de novas variáveis, ou seja, novas rotas, que deveriam fazer parte do problema principal. Isso é feito resolvendo-se um outro problema, conhecido como sub-problema ou *pricing problem*, que é encontrar as variáveis com custo relativo negativo. Ou seja, encontrar rotas que ao fazer parte do conjunto considerado, irão reduzir o custo da solução atual. Sob o ponto de vista do problema linear formulado, esta tarefa corresponde em encontrar variáveis fora da base, que, assumindo um valor diferente de zero, dentro da base, irão reduzir o valor da função objetivo. Quando uma ou mais variáveis custo relativo negativo são encontradas, são inseridas no problema principal. Resolve-se novamente o problema linear e inicia-se mais uma vez a busca de variáveis a serem incluídas no problema principal, ou seja, a solução do sub-problema. Após não existir mais variáveis com custo relativo negativo, a solução do sub-problema será a solução do problema como um todo.

Na solução apresentada por Larsen [La99] o subproblema é o Problema Elementar do Caminho Mínimo com Restrição de Janela de Tempo e Capacidade (PECMRJTC) ou *Elementary Shortest Path Problem with Time Window and Capacity Constraint (ESPPTCC)*. A redução de custo r_j de uma variável não básica j é calculada sobre as variáveis duais π do problema, ou seja,

$$r_j = c_j - \pi^T \delta_j \quad (2.19)$$

onde c_j é o custo da rota j e δ_j é o vetor rota, ou seja, o vetor de zeros e uns indicando os consumidores que a rota j atende, conforme definido anteriormente para (2.15).

No entanto, não havendo mais variáveis com custo relativo negativo, a solução encontrada pode não ser inteira. Para encontrar a solução inteira é necessária uma técnica de poda em árvore de busca (*branch-and-bound*). Como é sabido, o algoritmo de poda em árvore divide o espaço de busca em várias pequenas regiões, incluindo restrições adicionais a cada nó. A estratégia básica é eliminar a necessidade de avaliar cada nó, que corresponde uma parcela do espaço de busca, comparando a informação do limite superior global do problema até então conhecido com o limite inferior da solução dentro do subespaço relativo ao nó em questão. Caso o limite superior global seja menor que o limite inferior do nó a região que ele representa está descartada.

Uma consideração importante é a função utilizada para encontrar o limite inferior do problema no subespaço referente a um determinado nó da árvore de busca. [La99] utilizou a relaxação linear do problema de particionamento de conjunto como limite inferior e por isso é necessária a técnica de geração de colunas em cada limite inferior calculado. A técnica de poda em árvore de busca (*branch-and-bound*) juntamente com geração de colunas é conhecida como *branch-and-price*.

A cada vez que um limite inferior corresponde a uma solução inteira, por isso viável para o problema original, este passa a ser o novo limite superior global do problema, fazendo com que o *gap* fique mais “apertado”, aumentando a possibilidade de descartar novos nós ou subespaços e melhorando o desempenho do algoritmo.

Até aqui, a técnica mostrada segue estritamente a utilizada por [La99] e não difere muito das utilizadas por outros autores dentro do paradigma principal da decomposição de Dantzig-Wolfe para o PRVJT. As diferenças mais significativas, embora não sejam muitas, nos trabalhos encontrados na literatura estão nas variáveis escolhidas para ramificação, no critério de seleção do nó a ser ramificado e na utilização de planos de corte, assunto, este último, que será mais explorado em outra seção desta tese.

Seleção

A um dado momento, os nós folhas da árvore de busca e poda correspondem aos diferentes subespaços a serem avaliados em busca da solução do problema. Aqueles que possuem um limite inferior maior que o limite superior global conhecido será completamente descartado, ou seja, haverá uma poda. No entanto, o critério de escolha do nó a ser ramificado, dentre os vários que apresentam a possibilidade de conter a solução, deve ser definido. Dentre os mais utilizados está a escolha pelo melhor limite inferior ou *Best First Search*, que consiste na escolha dos subespaços onde a solução relaxada (limite inferior local) apresenta o menor valor. A idéia é concentrar os esforços para encontrar limites superiores melhores o mais cedo possível, podendo resultar em podas mais freqüentes. As outras duas estratégias possíveis são os tradicionais algoritmos de varredura em grafos: busca em profundidade e busca em largura.

Ramificação

A escolha do critério de ramificação, ou seja, as variáveis escolhidas para definir os diversos subespaços da árvore de busca, podem resultar em grandes alterações no desempenho do algoritmo. Dentre as técnicas mais freqüentemente utilizadas estão:

- Ramificação sobre o número de veículos.
- Ramificação sobre as variáveis de fluxo.
- Ramificação sobre janela de tempo.

Para maiores detalhes sobre as estratégias de ramificação nos algoritmos de poda para o PRVJT ver [La99] ou [GDD+95].

Entretanto tudo isso não tem sido suficiente. Embora a formulação com grande número de variáveis tenha eliminado um pouco a simetria do problema, presente no modelo compacto de fluxo em rede, o número de eventuais podas seria insuficiente para o término do algoritmo em tempo hábil. Por exemplo, quase nenhuma ou poucas das 56 instâncias de 100 consumidores propostas por Solomon poderiam ser resolvidas somente com estes recursos. Para melhorar o desempenho se faz necessário o entendimento das características do polítopo convexo, possibilitando a inserção de planos de corte a fim de apertar

significativamente a relaxação linear, ou seja, a qualidade do *gap* durante o algoritmo de poda e, conseqüentemente, o tempo de execução do algoritmo.

Planos de corte

A utilização de planos de corte com a finalidade de diminuir o *gap* entre a solução inteira e a relaxação, facilitando o trabalho para o algoritmo de poda, tem sido comum em quase todos os trabalhos no PRVJT que fazem opção por métodos exatos de otimização. Primeiramente, é necessário constatar que embora existam diversas opções exatas para a solução de problemas de programação inteira, dificilmente se vê livre do algoritmo de ramificação e poda, *branch-and-bound*, em alguma etapa do algoritmo. Existem dois fatores fundamentais para o desempenho das estratégias de poda: o primeiro é a simetria ou não do problema; e o segundo é a qualidade dos limites inferior e superior, a fim de possibilitar podas mais freqüentes e, conseqüentemente, a redução do espaço de busca. Quanto à simetria, muitas vezes é inerente ao problema em si, o que pode ser melhorado um pouco com a escolha adequada do modelo matemático. Uma formulação com muitas variáveis pode ser um caminho, quando a formulação compacta apresenta uma simetria muito grande, o que é o caso do PRVJT. No que se refere à qualidade do limite inferior, diminuindo o tempo de convergência da estratégia de poda, algoritmos utilizando modernas técnicas de corte têm produzido bons resultados. A inclusão de restrições adicionais no problema relaxado de tal forma que o politopo convexo resultante se aproxime mais da geometria que envolve os pontos viáveis do problema inteiro é o principal objetivo.

O conjunto de restrições de um problema de programação linear define um politopo convexo. No caso do PRVJT original, existe um conjunto de pontos distribuídos no espaço multidimensional, uma vez que existem as restrições de integralidade $x_{ijv} \in \{0,1\}$, conforme modelo descrito, equações (2.1) a (2.9). Considere esta “região” definida pelas restrições do problema original como I . Por sua vez, quando relaxamos o problema, eliminando a restrição da equação (2.9) e aceitando que a variável pode assumir qualquer valor de 0 a 1, ou seja, $0 \leq x_{ijv} \leq 1$, estabelecemos um politopo convexo R que contém a região I , ou seja, $I \subseteq R$. A estratégia de métodos baseados em planos de corte consiste em descobrir restrições lineares que podem ser inseridas no problema relaxado que transforme

R no mais próximo possível da “casca convexa” completamente aderente a I . Isto é, encontrar as restrições que definam um polítopo convexo que se aproxime suficiente de I para que se possa usar uma técnica de programação linear comum, levando a mesma solução inteira do problema original. Como o polítopo ideal é muito difícil, busca-se pelo menos algum que se aproxime o suficiente para reduzir o *gap*. Isso requer muitas vezes um profundo estudo das características geométricas da região viável do problema e também características particulares do mesmo, o que geralmente não é uma tarefa fácil.

Normalmente este é um processo iterativo. Parte-se da solução do problema relaxado, definido pela região R . Se a solução é inteira, esta pertence a região I , e por isso é solução do problema original. Como isso geralmente não acontece, procura-se por uma desigualdade que possa deixar a solução atual de fora da nova região viável, gerando um outro polítopo R' que se aproxima mais de I . Como houve um corte da região viável, isso geralmente resulta em um aumento no valor da função objetivo, o que é bom, já que se trata do limite inferior do problema original. Existem técnicas gerais de corte que se aplicam a uma classe ampla de problemas, por exemplo o corte de *Gomory*, ver [NW88]. Entretanto, geração de cortes utilizando o conhecimento das características particulares de cada problema têm sido mais utilizados, principalmente depois do sucesso obtido por Grötschel e Padberg em 1985 para o Problema do Caixeiro Viajante (PCV).

Algoritmos de separação ou *separation algorithms* são aqueles capazes de encontrar cortes a serem incluídos no problema principal. Os problemas de separação estão para o problema principal assim como está o problema de busca de variáveis com custo relativo negativo, ou *pricing problem* na estratégia de geração de colunas. Métodos de ramificação e poda (*branch-and-bound*) em conjunto com técnicas de inserção de planos de corte são conhecidos como *branch-and-cut*.

A grande maioria dos trabalhos utilizando métodos exatos que lograram bons resultados para as instâncias de Solomon fizeram uso de algum tipo de corte. Cook [CR99], assim como Kohl [Koh95], utiliza a inserção de desigualdades que viabilizam toda solução inteira, mas que deixam de fora a solução atual linear do problema relaxado. A desigualdade utilizada, uma generalização de um método utilizado previamente para o PCV, é conhecida como *subtour elimination*. No PRVJT esta consiste em estabelecer um número mínimo de veículos k que devem visitar um subconjunto S de consumidores. Este número não deve

ser muito maior que o número real necessário para atender estes consumidores. Uma restrição fraca, mas que nem sempre é atendida pela PL, é fazer $k = 1$. Desta forma, considerando $\delta(S)$, o conjunto de arcos que começam em um dos consumidores de S e terminam fora de S .

2.2.4 Trabalhos correlatos utilizando Métodos Heurísticos

Conforme descrito na seção 2.2.2, o PRVJT já foi demonstrado ser NP-difícil, conseqüentemente não é possível considerar métodos exatos para instâncias de ordem elevada. O conjunto de problemas testes mais utilizado para avaliar alternativas de solução foi proposto por Solomon em [SM87], composto de 56 instâncias, cada uma com 3 dimensões diferentes, 25, 50 e 100 consumidores. As instâncias de Solomon têm sido normalmente escolhidas tanto por métodos exatos como heurísticas para avaliação de desempenho. Os problemas das classes C1 e C2, que representam cenários onde os consumidores estão divididos e agrupados (*clustered*) em algumas sub-regiões, não têm sido um grande desafio para métodos exatos ou heurísticas, pois somente a instância C204 com 100 (cem) consumidores ainda não foi resolvida pelos primeiros, de um total de 17 instâncias. Já os problemas de disposição randômica, classes R1 e R2, e os mistos, classes RC1, RC2 são poucos aqueles resolvidos por algoritmos de otimização. A Tabela 2.4 mostra o desempenho geral dos métodos exatos na obtenção do ótimo global, conforme [CR99], [KLM01], [KDM+99] e [La99].

Conforme pode ser observado na Tabela 2.4, os métodos exatos apresentam um ótimo desempenho para instâncias de 25 consumidores. Já em instâncias de 50 consumidores, o desempenho depende muito das restrições e dos tipos de problemas. A queda de desempenho é muito grande com o afrouxamento das restrições de janela de tempo e na capacidade de carga, conforme pode ser constatado pela alteração no resultado de RC1 para RC2, com 50 consumidores, que foi de 100% para 75%. De forma geral, exceto na classe de problemas com consumidores agrupados C1 e C2, os métodos exatos obtêm um resultado pouco expressivo em problemas de 100 consumidores.

Tabela 2.4: Resultados ótimos obtidos até fevereiro/2005 para as instâncias de Solomon do PRVJT.

Instâncias de Solomon Classes	Nº de Consumidores	Nº de Problemas	Nº de Prob. Resolvidos	Percentual resolvido
R1: disposição randômica e restrições de carga e janela de tempo apertadas.	25	12	12	100%
	50	12	12	100%
	100	12	9	75%
	subtotal	36	33	92%
R2: disposição randômica e restrições de capacidade e janela de tempo menos apertadas.	25	11	11	100%
	50	11	7	64%
	100	11	1	9%
	subtotal	33	19	58%
C1: agrupados em sub-regiões e restrições de capacidade e janela de tempo apertadas	25	9	9	100%
	50	9	9	100%
	100	9	9	100%
	subtotal	27	27	100%
C2: agrupados em sub-regiões e restrições de capacidade e janela de tempo menos apertadas	25	8	8	100%
	50	8	8	100%
	100	8	7	88%
	subtotal	24	23	96%
RC1 – disposição mista e restrições de capacidade e janela de tempo apertadas	25	8	8	100%
	50	8	8	100%
	100	8	4	50%
	subtotal	24	20	83%
RC2 – disposição mista e restrições de capacidade e janela de tempo menos apertadas	25	8	8	100%
	50	8	6	75%
	100	8	3	38%
	subtotal	24	9	71%

O fato de que no mundo real os problemas de roteamento de veículos apresentam cenários com número de consumidores muito maior que 100, e ainda que a capacidade de processamento disponível não apresenta possibilidade de acompanhar o crescimento exponencial requerido, têm motivado muitos trabalhos utilizando métodos heurísticos. As heurísticas apresentam uma possibilidade de encontrar uma solução de boa qualidade em um tempo de processamento aceitável. Por outro lado, o desafio de projetar heurísticas que se aproximam do ótimo e robustas é também muito grande.

Na classe de métodos heurísticos, destacam-se as meta-heurísticas, que se caracterizam pelo esforço na tentativa de “escapar” dos mínimos locais durante a busca. Dentre as que

têm obtido resultados mais promissores para os problemas de roteamento de veículos estão o Algoritmo de Tempera Simulada (*Simulated Annealing*), a Busca Tabu (*Tabu Search*) e os Algoritmos Genéticos (*Genetic Algorithms*).

As principais abordagens heurísticas para a solução do Problema de Roteamento de Veículos com Janela de Tempo estático serão abordadas a seguir.

Antes de avaliar os principais trabalhos abordando heurísticas para a solução do PRVJT, faz-se necessário alguns comentários a respeito dos objetivos e critérios de arredondamento normalmente utilizados. A grande maioria tem tratado as instâncias de Solomon com o principal objetivo de reduzir o número de veículos (NV). Somente como segundo objetivo, a distância total percorrida é considerada (DT). O objetivo principal de redução do número de veículos é considerado pela quase totalidade dos trabalhos envolvendo metaheurísticas, tais como [RT95], [HG99], [BBB01], [BH04], [LLH01] e [CC02]. Uma pesquisa realizada identificou somente [RGP99] [TBG+97] e [TLZ+01] utilizando a distância percorrida como objetivo principal. Por outro lado, quase todos os trabalhos envolvendo métodos exatos tratam a distância percorrida como primeiro e único objetivo.

Outra diferença ocorre quanto ao critério de arredondamento no cálculo de distâncias e tempo de percurso na rota, enquanto as heurísticas geralmente consideram operações com ponto flutuante (normalmente 64 bits de representação), os métodos exatos costumam fazer um arredondamento na segunda casa decimal, possibilitando a utilização de inteiros de 32 bits. Ou seja, todos os dados de entrada são multiplicados por 100 e truncados. A partir de então, todas as operações são truncadas em inteiros.

Heurísticas Clássicas

As heurísticas clássicas surgiram basicamente entre os anos de 1960 e 1990. Este tipo de heurística utiliza procedimentos de busca com forte apelo à natureza do problema, na tentativa de limitar a pesquisa a uma reduzida porção do espaço de soluções viáveis. A grande vantagem é que são capazes de produzir soluções razoáveis em curto período de tempo. A utilização de heurísticas clássicas perdeu considerável espaço para as metaheurísticas, que serão abordadas posteriormente. Atualmente, ainda são usadas de

forma complementar, como por exemplo, para produzirem uma solução inicial em outros tipos de algoritmos. Outra importância das heurísticas clássicas está no fato de que permanecem presentes como operadores de outros métodos baseados em busca local, ou seja, são responsáveis por encontrar grupos de soluções a serem avaliadas, conhecidos como vizinhança. Apenas as principais heurísticas clássicas que se destacaram na solução para os problemas de roteamento de veículos serão aqui resumidas. Laporte [LGP+00] resume as principais heurísticas, clássicas e modernas, para o PRV.

Algoritmo de Economia (Saving Algorithm)

Este algoritmo foi proposto inicialmente por [CW64]. Ele utiliza a idéia de unir duas rotas em uma mais econômica. Inicia-se o algoritmo com uma solução inicial que possui uma rota para cada ponto de entrega. Em seguida, avaliam-se os ganhos com a união de rotas utilizando-se a variável de economia.

$$ss_{ij} = c_{i0} + c_{0j} - c_{ij}, \quad \forall i, j \in C \quad (2.20)$$

onde c_{i0} e c_{0j} representam os custos dos trechos eliminados do consumidor i ao depósito central para a primeira rota e do depósito central ao consumidor j para a segunda rota, respectivamente. Entretanto, um custo adicional, c_{ij} , é necessário para unir o último consumidor da primeira rota ao primeiro consumidor da segunda rota, formando uma única rota resultante. O ganho obtido será então a diferença, conforme (2.20) As rotas que apresentarem a melhor economia de custo são unidas. O processo é repetido até que não exista mais fusão possível com redução de custo.

Embora este método não venha sendo mais empregado de forma pura, a sua idéia original vêm inspirando trabalhos que o utiliza como uma das etapas do processo.

Agrupamento seguido de roteamento (Cluster-first, route-second)

Neste algoritmo resolve-se, numa primeira fase, o Problema Geral de Associação, ou *Generalized Assignment Problem (GAP)*, objetivando-se a formação de agrupamentos de consumidores ou *Clusters*, que serão atendidos por veículos independentes. Em seguida aplica-se uma técnica de solução para o Problema do Caixeiro Viajante (PCV), ou seja, um

único veículo em uma única região, o que pode ser feito utilizando métodos exatos, pois estes agrupamentos são geralmente de pequena ordem. [Mil76] apresenta uma solução, na qual, determinadas as regiões, os problemas de roteamento são solucionados utilizando um método baseado em relaxação e poda. [BS95] utiliza uma heurística do mesmo tipo, porém em uma primeira fase algumas localidades podem ser deixadas de fora. Depois de encontradas as rotas principais, as localidades restantes vão sendo “encaixadas” nas rotas principais.

Heurísticas de Melhoria

O procedimento básico nas heurísticas de melhoria, *improvement procedures*, foi descrito por [Lin65], é baseado no mecanismo λ -opt. Neste procedimento, cada rota é dividida em λ segmentos. Em seguida estes são reconectados de todas as formas possíveis. O primeiro viável, ou o melhor custo, é adotado em lugar do anterior. O procedimento é interrompido assim que não é mais possível melhorar as rotas. A complexidade de tempo deste algoritmo é $O(n^\lambda)$. Varias metaheurísticas utilizam este mecanismo como um dos operadores presentes na estratégia de busca, como [BBB01] e [HG99].

Metaheurísticas

Diversos problemas de otimização apresentam uma dificuldade tal que o horizonte de tempo necessário para obter a solução exata inviabiliza a utilização de métodos com este fim, para instâncias de dimensão mais elevada. Nestes casos a opção por heurísticas, que deverão produzir o melhor resultado possível utilizando um tempo aceitável, poderá ser a única opção. As primeiras heurísticas que foram propostas são hoje conhecidas como *heurísticas gulosas* ou *greedy heuristics* e geralmente existem para qualquer tipo de problema em otimização combinatória. As heurísticas gulosas começam geralmente com uma solução viável qualquer. A partir de características percebidas ou intuídas do problema, são realizadas avaliações de algumas possibilidades de movimento, ou seja, de modificação da solução atual. A partir das opções avaliadas, a solução encontrada de melhor custo para função objetivo é escolhida, melhorando a solução atual. As heurísticas clássicas previamente apresentadas são todas consideradas heurísticas gulosas e, em geral, uma reduzida porção do espaço de busca é avaliada.

Uma outra classe de heurísticas, que diferem das heurísticas gulosas por realizarem uma pesquisa intensa em porções do espaço de busca viável, são os algoritmos de busca local. Nestes algoritmos, uma busca intensa é realizada na vizinhança da solução viável atual. Um conjunto de operadores, que aplicados à solução atual conduz a novas soluções viáveis “próximas”, o que define o conjunto de vizinhança. Em [AEO99], Ahuja e outros apresentam uma avaliação das principais técnicas de busca local para problemas com amplo espaço de busca. A movimentação, ou seja, a próxima solução escolhida ou mesmo um conjunto de soluções será escolhido por algum tipo de critério inspirado na função objetivo do problema. Este critério poderá ser diretamente o menor valor da função objetivo (este critério é conhecido como de descida em ou *steepest descent*), ou alguma estratégia heurística mais elaborada. Uma estratégia simples de descida irá fazer com que o algoritmo pare no primeiro mínimo local, ou seja, o primeiro ponto onde toda a vizinhança possui uma avaliação da função objetivo pior que a solução atual.

E por fim, existe uma terceira classe de heurísticas conhecida como metaheurísticas. As metaheurísticas basicamente constituem em estratégias de alto nível para condução de um ou mais algoritmos heurísticos secundários na tentativa de solução de problemas de otimização. Conforme Glover [Glo86], uma metaheurística “*se refere a uma estratégia principal que guia e modifica outras heurísticas para produzir soluções melhores que estas, que normalmente são geradas para uma busca pela otimalidade local*”. Há também a tentativa de definição de Voß [VMO+99]: “*Uma metaheurística é um processo iterativo principal capaz de guiar e modificar a operação de heurísticas subordinadas para eficientemente produzir soluções de alta qualidade*”.

Existem diversas outras tentativas de definição formal de metaheurísticas, entretanto o conceito formal do que é uma metaheurística ainda não está fechado. Uma característica importante relacionada às metaheurísticas é o de busca inteligente. Embora o entendimento do que é uma busca inteligente no espaço de soluções seja muito abstrato, existem alguns ingredientes que podem ser destacados na busca inteligente:

- Está presente algum tipo de memória, permitindo que soluções avaliadas anteriormente influenciem na decisão dos movimentos futuros dentro do espaço de busca; e

- A presença de algum procedimento principal, ou estratégia, para guiar e modificar o comportamento de heurísticas subordinadas.

As principais estratégias hoje identificadas como metaheurísticas, mas não se limitando à, conforme Voß [VMO+99] são: procedimentos com memória adaptativa, busca tabu, sistemas de formigas, busca gulosa adaptativa e randômica ou *Greedy Randomized and Adaptive Search (GRASP)*, busca de vizinhança variável, algoritmos evolutivos, algoritmos genéticos, busca disseminada (*scatter search*), redes neurais, tempera simulada e seus híbridos.

Busca Tabu

Introdução

A idéia principal do algoritmo de busca tabu é a memória dos locais já visitados no espaço de busca, evitando busca repetidas em locais já visitados. Como as regiões já visitadas entram em uma lista, chamada de lista tabu, que como o nome sugere indica regiões proibidas que devam ser evitadas, o algoritmo atualiza a posição atual de busca mesmo para pontos com custo deteriorado (não proibidos), possibilitando escapar de mínimos locais. Estes movimentos ocorrem quando toda a vizinhança possui custo mais elevado ou toda a vizinhança de melhor custo se encontra na lista tabu. Adicionalmente pode ser utilizado algum critério adicional de aceitação, chamado de *critério de aspiração*, que ocasionalmente vai permitir movimentos para pontos mesmo presentes na lista tabu.

O método é considerado estático, quando a lista tabu, que é do tipo *FIFO (First In First Out)*, tem um comprimento fixo. Já o tabu reativo modifica o tamanho da lista ao longo da busca, utilizando algum critério do comportamento do problema em questão. Alterações dinâmicas do tamanho da lista como do critério tabu (raio de influência, vizinhança, etc.) permitem um adequado balanceamento entre diversificação (pesquisa por regiões inexploradas).

Trabalhos Anteriores

Alguns bons resultados têm sido encontrados para PRVJT utilizando algoritmo de busca tabu, dentre os principais estão Cordeau [CLM00], Taillard [TBG+97] e Rochat [RT95].

O trabalho de Rochat [RT95] talvez tenha sido o que mais trouxe novos resultados de uma só vez para o conjunto de problemas de Solomon, muitos dos quais ainda não foram superados. Do conjunto de 56 instâncias de 100 consumidores 40 resultados foram superados na ocasião. Atualmente este trabalho ainda não foi superado em 19 instâncias, embora 17 correspondam as classes inteiras C1 e C2, que como foi dito, os mesmos resultados têm sido alcançados pela maioria dos trabalhos atuais.

Algumas das contribuições de Rochat [RT95] foram sua proposta de paralelização do algoritmo tabu e a robustez obtida pelo algoritmo. A principal motivação da proposta, segundo o próprio autor, é baseada na seguinte constatação: *“o uso inteligente da busca randômica, diferentemente de uma busca cega uniforme, que considera a história e a medição da atratividade, é um tipo útil de diversificação que evita o uso mais complexo de memória”*.

O algoritmo de Rochat [RT95] inicialmente produz um conjunto I de diferentes soluções para o PRVJT, diferentes uma das outras, utilizando um algoritmo de busca local. Uma constatação de Rochat foi que as diferentes soluções inicialmente geradas, embora de qualidade razoável, poderia ser trabalhadas para produzir uma solução final de melhor qualidade. Esta idéia de Rochat é parecida com a utilizada neste trabalho, conforme será mostrado no capítulo seguinte. Então, na primeira fase, as diferentes soluções presentes em I produzem um conjunto de rotas T . Na segunda fase do algoritmo, as melhores rotas deverão ser selecionadas e melhoradas. Baseando-se no princípio que rotas de qualidade (não individualmente, mas para a solução como um todo) são aquelas presentes nas melhores soluções de I , cada rota em T é relacionada com o valor da solução da qual esta foi extraída. As rotas são então ordenadas pelo valor da solução relacionada e aquelas com apenas um consumidor são eliminadas. As rotas são então aleatoriamente escolhidas, dando preferência por rotas com menor valor da solução relacionada (o autor atribui uma probabilidade de ser escolhida, utilizando uma função do valor da solução associada à rota), formando um novo conjunto de rotas S . As rotas escolhidas que contêm consumidores que já estão em S são desprezadas. O algoritmo é interrompido quando não há mais rotas possíveis em S . O conjunto de rotas em S podem não formar uma solução completa. Neste caso o algoritmo de busca inicial é novamente utilizado para completar a solução. O algoritmo completo utilizado está resumido abaixo:

Inicialização

- a. *Produção de I soluções diferentes utilizando a busca tabu;*

Atualização do Conjunto de Rotas

- b. *Relacionada cada rota produzida com o valor da solução (função objetivo) de onde a mesma foi retirada;*
- c. *Eliminação das rotas com um único consumidor;*
- d. *Inserção das demais rotas em um vetor T;*
- e. *Ordenação do vetor T conforme os valores da solução de onde a rota foi retirada;*

Diversificação e Intensificação

- f. *Inicia $T'=T$; e $S = \emptyset$;*
- g. *Enquanto T' não vazio faça:*
- *Escolhe uma rota r de T' baseando-se na probabilidade de cada uma, calculada sobre o valor relativo da qualidade da solução de onde foi retirada;*
 - *Insere a rota r em S ;*
 - *Elimina todas as rotas de T' que possuem os consumidores presentes na rota r ;*
- h. *Completa a solução S , se houver alguns consumidores que não foram atendidos;*
- i. *Melhora a solução S utilizando a busca tabu;*
- j. *Se tempo limite alcançado termina, senão, insere as novas rotas em T e vai para o passo “b.”*

A etapa de diversificação e intensificação é executada repetidas vezes, até algum critério de parada for satisfeito, como tempo ou número máximo de iterações. A implementação paralela do tipo mestre e escravo foi implementada. As etapas mais consumidoras de tempo, que são “a” e “i” foram executadas de forma paralela e independente em diferentes máquinas (processo escravo). Após a finalização, todas as rotas são enviadas para o

processo principal, onde todas as demais etapas são realizadas de forma centralizada (processo mestre).

Adicionalmente, Rochat [RT95] realiza uma última etapa, chamada de técnica de pós-otimização. Esta etapa consiste em resolver o modelo de particionamento de conjuntos, conforme (2.14) a (2.15), sendo o conjunto de rotas R todas as rotas produzidas em T . Foi utilizada precisão dupla para todos os cálculos realizados. Os resultados de Rochat [RT95] foram muito relevantes, conforme descrito inicialmente. Em termos de número de veículos, principal objetivo também adotado, houve alguns resultados piores que aqueles então conhecidos na literatura. Na classe R1 quatro instâncias apresentaram resultado com 1 veículo a mais; e na classe R2 duas instâncias. Nas instâncias RC1 e RC2, 3 e 1 instâncias apresentaram um veículo a mais, respectivamente. No entanto, em todas as outras instâncias o número de veículos foi alcançado ou reduzido. O seu principal avanço foi mantendo o número de veículos já conhecido e reduzindo a distância percorrida. Um total de 27 resultados nas 56 instâncias de Solomon com 100 consumidores foram alcançados ou melhorados.

Cordeau [CLM00], no uso da busca tabu, não se limita ao espaço de soluções viáveis, mas permite a violação de restrições de capacidade e janelas de tempo. Esse espaço é denotado por S . Isso significa que é permitido um veículo atender um conjunto de consumidores com soma total de carga superior a sua capacidade, violação das restrições (2.3), e chegar ao consumidor após a janela de tempo, $s_{iv} > b_i$. é aceitável nas restrições (2.8). No entanto, se um veículo chega antes do início da janela de tempo, este deverá esperar um tempo $w_i = a_i - s_{iv}$. Para uma dada solução qualquer $s \in S$, o autor estabelece as seguintes variáveis $c(s)$, $q(s)$, $d(s)$ e $w(s)$ como sendo o tempo de viagem total, violação de carga, violação no tempo total de retorno ao depósito central e violação das janelas de tempo, respectivamente. O autor então usa a função custo $f(s) = c(s) + \alpha'q(s) + \beta'd(s) + \gamma'w(s)$, onde α' , β' e γ' são parâmetros positivos. Os parâmetros exercem um papel inicial de relaxação e são ajustados dinamicamente. Quanto mais elevado são estes multiplicadores maior será a pressão para reduzir a violação de carga e janelas de tempo, conduzindo a soluções viáveis.

Uma solução s é definida pelo simples conjunto ordenado de pares $B(s) = \{(i, k) : i \text{ consumidor atendido e } k \text{ é o veículo utilizado}\}$. A vizinhança $N(s)$ de uma solução s é definida pela simples troca de atendimento de um consumidor, ou seja, (i, k) é substituído por (i, k') , na posição que minimiza $f(s)$. A lista tabu é representada pela impossibilidade de um consumidor voltar para uma mesma rota k após ter sido retirado, durante as próximas θ iterações. O critério de aspiração é utilizado sempre que $f(s) > f(s')$, permitindo o movimento para uma região proibida pela lista tabu. Adicionalmente, com a finalidade de aumentar a diversificação, Cordeau adiciona uma penalidade sobre qualquer vizinhança $\bar{s} \in N(s)$, que representa uma deteriorização da função objetivo, ou seja, se $f(\bar{s}) > f(s)$. Esta penalidade é calculada sobre a frequência que um determinado consumidor i é inserido em uma rota k , atributo (i, k) , e também o que o autor chama de fator de escala. A frequência de inserção de um atributo (i, k) é $\rho_{(i, k)}$ é calculado como a soma do número de vezes que cada par (i, k) foi inserido na solução durante todo o processo de busca. Já o fator de escala é calculado como o tempo total de viagem, definido previamente, multiplicado pela raiz quadrada do número de consumidores vezes o número de veículos, $c(s)\sqrt{nm}$. A penalidade completa então é

$$p(\bar{s}) = \begin{cases} \lambda c(\bar{s})\sqrt{nm} \sum_{(i, k) \in B(\bar{s})} \rho_{ik} & \text{se } f(\bar{s}) > f(s) \\ 0 & \text{caso contrário} \end{cases} \quad (2.21)$$

O fator λ é utilizado dinamicamente para ajustar a intensidade da diversificação.

O autor testou o algoritmo em uma série de instâncias por ele mesmo criadas e para as instâncias de Solomon [SM87]. Foram usadas instâncias com múltiplos depósitos centrais, do chamado Problema de Roteamento de Veículos com Janela de Tempo e Múltiplos Depósitos (PRVJTMD) e também do Problema Periódico de Roteamento de Veículos com Janela de Tempo (PPRVJT). Entretanto, pelo que se tem conhecimento, as suas próprias instâncias não foram posteriormente reutilizadas, não permitindo comparação, e por isso esses resultados não serão aqui comentados.

Quanto às instâncias de Solomon utilizadas alguns bons resultados foram encontrados pelo autor. Quanto aos objetivos, o autor tem dificuldade para minimizar o número de veículos

e por isso fixou este valor no mínimo encontrado na literatura. Como foi descrita, sua formulação permite soluções inicialmente inviáveis, possibilitando tal procedimento. O algoritmo realizará η iterações. Se transcorrido 10% do algoritmo ($\eta/10$ iterações), nenhuma solução viável foi encontrada, o algoritmo do autor permite um veículos a mais nas soluções, continuando o processo de busca. O segundo objetivo foi o tempo total de viagem, o que difere um pouco do normalmente utilizado, que é a distância total percorrida. Entretanto, nas instâncias de Solomon, é estabelecido que o tempo para percorrer uma unidade do espaço euclidiano é uma unidade de tempo, por isso, os resultados são próximos. A diferença se dá quando o veículo espera a abertura de uma janela de tempo, consumindo mais tempo de viagem.

Os testes foram realizados para as 56 instâncias de Solomon de 100 consumidores. Conforme já era previsto, para os problemas C1 e C2 todos os melhores resultados conhecidos foram alcançados. Na ocasião, 11 novos resultados foram obtidos e 19 alcançados. Atualmente, o algoritmo tabu descrito ainda não foi superado em apenas uma instância, RC104. Em termos do primeiro objetivo, minimização do número de veículos, somente em dois casos o melhor resultado atual não foi alcançado, nas instâncias R104 e R112.

Ainda utilizando a busca tabu, Taillard [TBG+97] mantém três resultados não superados para o conjunto de instâncias de Solomon. Outros trabalhos utilizando busca tabu são [Car95], [PKG+96] e [Bra98].

Algoritmo de Têmpera Simulada (Simulated Annealing)

O algoritmo de têmpera simulada tem como idéia principal permitir que a movimentação dentro do espaço de busca possa ir para um ponto de custo superior, ou seja, possibilita a deteriorização da solução atual para que seja possível evitar mínimos locais.

O procedimento é inspirado no fenômeno físico de resfriamento de metais. Inicialmente, com o metal quente, as partículas possuem muita energia e podem se locomover aleatoriamente sem restrições. A partir do momento que a temperatura vai esfriando, as moléculas tendem a se acomodar na estrutura cristalina natural do material em uso. Esta acomodação é baseada no princípio de minimização da energia total do metal. Enquanto a

temperatura é alta, mesmo atingindo uma região mais estável (melhor valor da função objetivo) ainda é possível um salto aleatório para qualquer outra posição. Estudos mostram que se a temperatura é resfriada vagarosamente, a estrutura final tende a ser a de menor energia, definida pelo tipo de material, que é o ótimo global da função. Se a temperatura não for reduzida suficientemente lenta, um mínimo local poderá ser atingido.

Na simulação de problemas de otimização é realizado um processo análogo. Inicia-se com uma solução viável inicial para o problema. Um movimento aleatório baseado na utilização de operadores diversos indica uma outra suposta solução dentro do espaço de busca. Enquanto uma variável de controle que simula a temperatura está elevada, a nova solução é aceita indiscriminadamente. Isso representa a possibilidade de sair de mínimos locais. A medida que a temperatura vai esfriando, ou seja, o valor da variável de controle fica menor, a nova solução terá menor probabilidade de ser aceita, dependendo de um valor randômico calculado e da elevação no valor da função objetivo. Todo movimento que resulta na redução do valor da função objetivo é aceito. A probabilidade de ser aceita uma nova solução é dada por $e^{\Delta/T}$, onde Δ representa a variação na função objetivo e T o valor da temperatura. Esta equação é oriunda da distribuição de Boltzmann, que define a probabilidade de uma partícula estar em um determinado nível de energia, dada uma temperatura.

Trabalhos Anteriores

Dentre trabalhos utilizando o algoritmo de temperatura simulada para a solução do PRVJT estão [CR96], [TOS94], [CC02] e [LLH03].

Um trabalho recente utilizando o algoritmo de temperatura simulada para o PRVJT é o de Li e outros [LLH03], testado com as instâncias de Solomon. [LLH03] utiliza a heurística *Push Forward Insertion Heuristic* (PFIH) de Solomon [SM87] para gerar uma solução inicial. Também foi usada uma estratégia de k-reinicialização para evitar mínimos locais e aumentar a robustez do algoritmo. Três operadores foram utilizados para implementar a busca local: o operador de deslocamento (*shift operator*), o operador de troca (*exchange*) e operações de reinserção, todos baseados nos operadores já utilizados por Salvesberg [Sav92] e Taillard [TBG+97].

Li [LLH03] obteve 9 resultados inéditos para as 56 instâncias de Solomon com 100 consumidores, um ainda é mantido (R103). Também foram alcançados os melhores conhecidos em 19 instâncias.

Algoritmos Evolutivos

Os primeiros algoritmos a utilizarem a evolução natural como idéia central para a resolução de problemas apareceram durante a década de 50, Fraser [Fra57] e Box [Box57]. Em 1966 Fogel [FOW66], dentre outros, propôs um método denominado programação evolutiva ou *evolutionary programming* e em 1973, Rothenberg [Rec73] introduziu o método denominado estratégia de evolução ou *evolution strategies*. O algoritmo genético (AG), *genetic algorithm* (GA), foi proposto por Holland [Hol75] em 1975. As estratégias de todas estas variantes se baseiam principalmente nas regras de reprodução e seleção de Darwin de 1859 [Dar59]. O algoritmo genético tem se tornado muito popular por encontrar boas soluções nos problemas matematicamente complicados, como os problemas de roteamento de veículos.

O AG é baseado em uma população de cromossomos (seqüência de *bits* ou valores numéricos), onde cada indivíduo representa uma proposta para a solução do problema. Novos cromossomos, conseqüentemente novas sugestões para a solução do problema, são gerados combinando os cromossomos, ou indivíduos, já existentes. Ao conjunto de indivíduos de uma geração dá-se o nome de população. A característica mais importante do indivíduo para o algoritmo genético é a sua adaptabilidade ou *fitness*. O *fitness* de um indivíduo está relacionado ao valor da função objetivo quando avaliada com os parâmetros neste codificados. Quanto maior o valor da adaptabilidade do indivíduo maior a chance de ser combinado para formar outros indivíduos descendentes. Isto assegura que os melhores, que representam os melhores parâmetros para o problema, estarão mais bem representados na geração futura, influenciando a busca em alguma direção do espaço de soluções. Na operação de cruzamento, ou *crossover*, dois indivíduos previamente selecionados têm partes de sua *string* de bits trocadas. A forma mais simples é quebrando as seqüências em um único ponto e permutando as partes geradas, ver Figura 2.1.

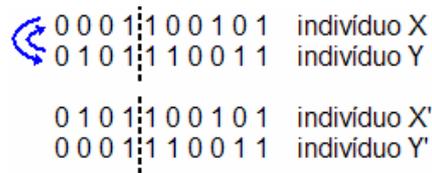


Figura 2.1: Cruzamento de indivíduos, gerando dois outros descendentes.

Gerações sucessivas vão sendo criadas até que algum critério de parada seja satisfeito: geralmente o estacionamento da população em algum nível ou pelo número máximo de avaliações permitidas.

O algoritmo básico descrito acima pode ser apresentado da seguinte forma:

```

procedimento AG; {
  Num_gerações = 1;
  Gera_População_Inicial;
  Avalia_Indivíduos;
  enquanto (não convergiu) {
    incrementa (Num_gerações);
    Seleciona_Indivíduos_Nova_Geração;
    Cruzamento;
    Mutação;
    Avalia_Indivíduos;
  }
}

```

Algoritmo 2.1: Algoritmo genético básico

É muito importante avaliar a forma de codificar as variáveis de um problema específico no AG, que pode ter uma influência muito grande no desempenho do algoritmo em si. O uso da representação dos indivíduos como seqüências de bits, assim como na Figura 2.1, pode não ser adequada. O tipo específico do problema pode indicar uma forma mais direta e simples. É comum o cromossomo ser considerado um vetor de valores reais [MM91].

Trabalhos Anteriores

Dentre os trabalhos utilizando algoritmos evolutivos para o PRVJT, alguns apresentam resultados significativos para os problemas de Solomon [HG99] [BBB01] [BG01] [Mes02].

Homberger [HG99] propõe duas estratégias evolutivas ES1 e ES2 para o PRVJT. O autor caracteriza seus algoritmos como metaheurísticas evolutivas, segundo o qual, é uma generalização dos algoritmos genéticos, pois não necessitam de codificação das soluções nos indivíduos. Segundo o critério utilizado pelo autor, as metaheurísticas evolutivas podem fazer uso de indivíduos e das operações de recombinação e mutação diretamente no formato de soluções para o problema, sem necessidade de codificação.

Na estratégia ES1, um indivíduo que sofre uma mutação passa a fazer parte da nova geração sem nenhum processo de recombinação. A representação de um indivíduo é um vetor de inteiros, *SVector*, representando seqüencialmente cada consumidor visitado. O inteiro zero é utilizado para representar o depósito central. ES1 trabalha somente com soluções viáveis. Como os objetivos são minimização de veículos e distância total percorrida, nesta ordem de prioridade, [HG99] utiliza um parâmetro de controle para direcionar em duas fases distintas a busca. Este parâmetro binário *RElimination* define se há necessidade de reduzir o número de rotas ou somente distância percorrida, o que é feito em duas fases distintas. Adicionalmente, um outro parâmetro utilizado, *NMoves*, define a quantidade efetiva de variações (movimentos) a ser realizada no procedimento mutação. Durante a mutação é realizado um movimento aleatório para a vizinhança do indivíduo solução. A vizinhança é definida como o conjunto $N(SVector)$. O conjunto vizinhança é a união de possíveis soluções geradas a partir de operadores já amplamente utilizados na literatura: *Or-opt*, *2-opt**, e *1-Interchange*. Logo o conjunto vizinhança é $N(SVector) = \{Or-opt(SVector), 2-opt*(SVector), 1-Interchange(SVector)\}$. O operador utilizado é escolhido aleatoriamente. Na fase de eliminação de rotas, um operador *Or-opt* modificado é utilizado. O operador *Or-opt* modificado tenta eliminar uma rota de um indivíduo solução *SVector*, tentando retirar mais um consumidor da menor rota e inseri-lo em outra. O critério de escolha da posição na nova rota do consumidor *i* retirado da menor rota é o menor deslocamento na partida do consumidor seguinte, ou na chegada ao depósito central, quando avaliado na última posição. Somente em caso de empate em

diferentes posições, o critério mínima distância percorrida é empregado. Caso o consumidor retirado não possa ser inserido em outra rota sem violação de restrições, a menor rota permanece inalterada. Por fim, [HG99] controla a intensificação e diversificação variando os valores dos dois parâmetros de controle *NMoves* e *RElimination*. Geralmente altos valores de *NMoves* são usados quando *RElimination* indica hora de reduzir a distância percorrida e baixos valores durante a redução do número de veículos.

Diferentemente da estratégia ES1, os indivíduos na estratégia ES2 são recombinados (*crossover*) antes de passarem pelo processo de mutação. Um conjunto de descendentes λ é gerado por recombinação de indivíduos (pais) escolhidos de forma randômica, sem influência da adaptabilidade. O número de descendentes inicialmente é maior que o número de indivíduos da população μ ($\lambda > \mu$). Em seguida, μ elementos serão escolhidos, desta vez a escolha é feita considerando os de melhor função adaptabilidade. Adicionalmente ao vetor solução *SVector* cada indivíduo possui uma regra de mutação ou código de mutação, *mutation code MC*. Um vetor solução, *SVector*, tem uma relação direta com seu código de mutação, pois foi gerado através deste. Desta forma, o processo de recombinação considera não só o vetor solução, mas também a regra de mutação ou código de mutação *MC*. O processo se dá com a escolha inicial aleatória de 3 indivíduos (pais). Um descendente intermediário será gerado primeiramente pela cópia direta do vetor solução do indivíduo 3. Este descendente irá receber um código de mutação resultante da recombinação dos códigos de mutação dos indivíduos 1 e 2, MC^* . Adicionalmente, o descendente intermediário também recebe o *RElimination* do terceiro indivíduo, *RE3*. O descendente final irá resultar da aplicação da mutação ao vetor solução do indivíduo 3, conforme o código de mutação encontrado, MC^* . O processo pode ser visualizado na Figura 2.2.

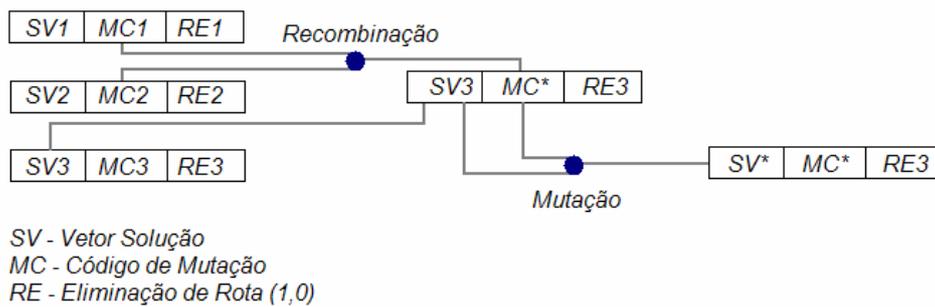


Figura 2.2: Processo de recombinação de três indivíduos, seguida de mutação, gerando um descendente. Este procedimento foi utilizado por [HG99] na sua estratégia ES2.

O código de mutação consiste de um vetor de dimensão duas vezes o de *SVector*, cada consumidor aparecendo duas vezes, em uma ordem qualquer. Durante o processo de mutação, a primeira incidência corresponde a ação de remover aquele consumidor de *SVector*, enquanto a segunda incidência corresponde a ação de inserir novamente o consumidor. A remoção é feita de forma direta. Na inserção dos consumidores, a melhor posição de atendimento corresponde àquela que resultar no melhor valor da função adaptabilidade: se *RElimination* for 0, a função adaptabilidade corresponde à menor distância total percorrida; se *RElimination* for 1, a função adaptabilidade corresponde ao menor deslocamento no tempo saída provocado no consumidor atendido em seguida.

Conforme dito, os μ melhores descendentes do total de λ previamente gerados ($\lambda > \mu$) são escolhidos. [HG99] testou inicialmente a escolha baseada primeiramente no número de veículos de cada descendente (primeiro critério) e depois a distância percorrida (segundo critério) como critério de desempate. Os resultados não foram satisfatórios. Segundo o autor porque a menor distância percorrida não necessariamente conduz ao menor número de veículos e vice-versa. Conseqüentemente, foi necessário inserir outros critérios na seleção dos indivíduos. O primeiro e simples critério adicionado foi o número de consumidores na menor rota, C_r . O segundo foi chamado de tempo de retardo mínimo ou *minimal delay* da menor rota de uma solução, D_R . O tempo de retardo mínimo é a soma da menor violação de tempo causada pela inserção de cada consumidor k da rota R , em outra rota qualquer, isto é,

$$D_R = \sum_{k \in R} D_k \quad (2.22)$$

onde D_k é calculado da seguinte forma:

- $D_k = 0$ se o consumidor k pode ser inserido em qualquer outra rota $R' \neq R$, sem nenhuma violação de janela de tempo ou capacidade do veículo;
- $D_k = \infty$ se a inserção de k em qualquer outra rota $R' \neq R$ causa uma violação da capacidade de carga; e
- $D_k = \text{mínimo tempo de violação}$, se a inserção de k causa, para uma ou mais rotas, violação na de janela de tempo, mas não de capacidade do veículo. Neste caso, a menor violação de tempo encontrada em uma rota, somada ao atraso causado no consumidor subjacente, é o valor de D_k .

[HG99] avaliou a seleção dos indivíduos em ordem lexicográfica, ou seja, número de veículos como primeiro parâmetro, para soluções com mesmo número de veículos a distância percorrida é considerada, depois o número de consumidores na menor rota C_k e, por último, o retardo D_R . Entretanto, melhores resultados foram obtidos executando o processo em duas etapas. As duas etapas são:

- Em uma primeira etapa somente o número de veículos, seguido do número de consumidores da menor rota e então do valor do retardo D_k são considerados em ordem lexicográfica;
- Em uma segunda fase, após o número de veículos ter sido minimizado, a busca é redirecionada para a menor distância total percorrida. Nesta fase, o número de veículos continua sendo o primeiro objetivo, seguido da menor distância percorrida.

As estratégias ES1 e ES2 foram testadas utilizando as 56 instâncias de Solomon contendo 100 consumidores e as instâncias D-417 e E-417 propostas por Russell [RR95], contendo ambas 417 consumidores. ES1 e ES2 obtiveram redução do número de veículos em duas instâncias de R1 (R104 e R112). ES1 melhorou também o resultado de R109, mas neste caso a redução foi no segundo critério, distância total percorrida. Neste quesito ES2 também foi responsável por dois novos resultados, R105 e R107. Nas classes R2, cinco instâncias tiveram seus resultados superados por ES1 e três por ES2, mantendo o número de veículos inalterado. Nas classes C1 e C2 todos os resultados foram equivalentes aos resultados anteriores. Na classe RC1 somente a estratégia ES2 foi capaz de melhorar dois resultados, enquanto que na classe RC2 duas novas instâncias distintas foram superadas

por cada estratégia, ES1 e ES2. Ao todo 20 novos resultados foram alcançados na ocasião. Atualmente, somente dois ainda não foram superados (R109 e R201). Entretanto, [HG99] ainda detêm um dos menores valores para a quantidade total de veículos utilizados, 406 veículos para as 56 instâncias.

Berger [BBB01] utiliza uma implementação onde duas populações com objetivos diferentes são executadas em paralelo. Berger utiliza o termo algoritmo genético, embora também não exista codificação da solução nos indivíduos, se referindo a estes como indivíduos solução. A primeira população, chamada de Pop1, tem como primeiro objetivo a redução da distância total percorrida, para um número fixo de veículos, enquanto a segunda, Pop2, trabalha para reduzir a violação das janelas de tempo, dado um número também fixo de veículos, mas inferior em uma unidade. O objetivo principal global é a redução do número de veículos, seguido da distância total percorrida. A principal estratégia utilizada por Berger [BBB01] para escapar dos mínimos locais na minimização do número de veículos foi trabalhar com um problema relaxado em uma das populações (Pop2), assim como o fez Cordeau [CLM00] utilizando a busca tabu. Assim que a segunda população (Pop2) obtém uma solução viável para um certo número de veículos, esta solução é passada para a Pop1, que busca a menor distância total percorrida com aquele número de veículos. O número de veículos fixado em Pop2 é então reduzido novamente em uma unidade, recomeçando a tentativa de encontrar uma solução viável. Nesta ocasião, toda a população de Pop1 é também substituída pela população de Pop2, reiniciando o processo de redução de distância percorrida. Isso possibilita Pop1 trabalhar sempre com pelo menos uma solução viável. Desta forma, as populações trabalham sempre com o número de rotas (veículos) diferindo de uma unidade, R_{min} e $R_{min} - 1$, em Pop1 e Pop2 respectivamente.

Berger [BBB01] utiliza um algoritmo genético em regime, conhecido também na literatura como *steady-state*, o qual, gera descendentes que vão sendo incorporados na população atual até atingir um número igual a n_p . Neste momento, os n_p piores indivíduos são eliminados. A função de avaliação dos indivíduos para realizar esta eliminação, que difere da utilizada na seleção, é definida nas equações (2.23) a (2.25).

$$Eval_i = E_i + CV_i \quad (2.23)$$

$$E_i = r_i - r_m + \frac{d_i}{\max(d_i, d_m)} \quad (2.24)$$

$$CV_i = \sum_{j=1}^n \alpha_j^* \max(0, b_i^j - l_j) + \beta^* Viol_i \quad (2.25)$$

onde

r_i = número de rotas no indivíduo i ,

r_m = limite inferior para número de rotas (relação da demanda total sobre a capacidade do veículo),

d_i = distância total percorrida do indivíduo i ,

d_m = média da distância total percorrida dos indivíduos da primeira geração,

n = número de consumidores,

α_j^* = penalidade associada a violação da janela temporal do consumidor j ,

b_i^j = tempo previsto para o consumidor j ser visitado no indivíduo i ,

l_j = tempo limite para visitar o consumidor j ,

β^* = penalidade associada ao número de janelas de tempo violadas, e

$Viol_i$ = número de janelas de tempo violadas pelo indivíduo i .

Os indivíduos vão sendo selecionados dois a dois para recombinação, aleatoriamente, porém com probabilidades diferentes, conforme adaptabilidades definidas pelas equações (2.26) e (2.27) abaixo. A seleção é feita utilizando a técnica de roleta giratória proposta por Goldberg [Gol89].

As funções adaptabilidades utilizadas na seleção para Pop1 e Pop2, respectivamente, são:

$$fitness_i = d_i + \sum_{i=1}^n \alpha_j^* \max(0, b_i^j - l_j) + \beta^* Viol_i \quad (2.26)$$

$$fitness_i = \sum_{i=1}^n \alpha_j^* \max(0, b_i^j - l_j) + \beta^* Viol_i \quad (2.27)$$

Onde as variáveis são as mesmas definidas para as equações de (2.23) a (2.25).

Berger [BBB01] utiliza dois tipos distintos de cruzamento, $IB_X(k)$ e $IRN_X(k)$. O primeiro tipo, IB_X realiza o seguinte procedimento:

1. Escolhe algumas rotas $\{r_1\}$, do primeiro indivíduo selecionado para o cruzamento P1, com probabilidade proporcional ao número de consumidores e à distância média entre consumidores consecutivos;
2. Escolhe algumas rotas $\{r_2\}$ do segundo indivíduo selecionado P2, baseado em um critério de vizinhança ao conjunto previamente selecionado $\{r_1\}$;
3. Remove alguns consumidores de $\{r_1\}$, baseados na violação da janela de tempo e também aleatoriamente;
4. Insere os consumidores restantes de $\{r_1\}$, independentes de sua rota original, nas rotas selecionadas $\{r_2\}$, utilizando a heurística de inserção proposta por Solomon [SM87] modificada, gerando sempre soluções viáveis;
5. Para consumidores cuja inserção em $\{r_2\}$ foi inviável, são criadas novas rotas utilizando a heurística de inserção de Solomon modificada;

O procedimento de recombinação IB_X é utilizado somente na Pop1.

Já o procedimento IRN_X é utilizado em Pop2, onde a evolução direciona a busca na tentativa de minimizar as restrições de tempo violadas para um número fixo de veículo. A estratégia inicial é semelhante a IB_X , removendo alguns consumidores, baseados no mesmo critério, de um dos indivíduos selecionados. Rotas dos indivíduos que participam do cruzamento são também escolhidas, da mesma forma. Entretanto, na etapa final do procedimento IB_X , rotas novas são criadas para evitar violação das restrições. Diferentemente, IRN_X , insere os consumidores, podendo causar violação se necessário, para manter o número de veículos (rotas) fixos. O algoritmo utilizado para definir a rota e a posição de inserção dos consumidores restantes foi proposto por Liu e Shen [LS99], que define estas escolhas com base no máximo valor do “custo de arrependimento” ou *regret cost*, conforme (2.28). No entanto, o autor estende a idéia do *regret cost* para incluir a penalidade pela violação da janela de tempo dada pela equação (2.25). Liu introduz o conceito de rotas da vizinhança de um consumidor i , que se baseia na proximidade para

definir o subconjunto de rotas onde há maior chance de uma troca de posição resultar em ganho positivo na função custo, reduzindo o espaço de busca de forma heurística (ver [LS99] para maiores detalhes).

$$regret\ cost = \sum_{r \in RN(i)} \{c_i(r) - c_i(r^*)\} \quad (2.28)$$

onde

$RN(i)$ = rotas da vizinhança do consumidor i ,

$c_i(r)$ = custo mínimo de inserção de i em r ,

$c_i(r^*)$ = custo mínimo de inserção de i nas rotas da vizinhança de i .

Berger [BBB01] utiliza seis tipos de operadores para realizar as mutações nos indivíduos, AC_M, LNSB_M, EE_M, RS_M, RSS_M e RC_M. O primeiro, AC_M (*ant colony mutation*) utiliza o sistema de formigas proposto por Gambardella [GTA99]. LNSB_M (*Local Neighborhood Search Based Mutation*) utiliza a estratégia de busca local proposta por Shaw [Sha98], que consiste em um processo repetitivo de retirada e re-inserção de consumidores utilizando uma árvore baseada em restrições, ou seja, programação por restrições. EE_M (*Edge Exchange Mutation*) é baseada no operador λ -interchange, que faz troca de consumidores em duas rotas vizinhas. RS_M (*Repair Solution Mutation*) também é baseada no mesmo operador, mas se restringe aos consumidores responsáveis pelas violações das janelas de tempo no momento de realizar trocas de consumidores. RSS_M (*Reinsert Shortest Solomon Mutation*) tenta eliminar a rota com menor número de consumidores, fazendo uma tentativa de inserção destes nas rotas já existentes, onde a escolha da posição utiliza novamente a função custo de Liu [LS99]. Por fim, o operador de mutação RC_M (*Reordering Customers Mutation*) tenta reduzir a distância percorrida de soluções viáveis, reordenando os consumidores de cada rota, utilizando o procedimento de inserção I1 de Solomon [SM87].

Berger [BBB01] então avalia o seu algoritmo, comparando também com os resultados recentes sobre as 56 instâncias de 100 consumidores de Solomon [SM87]. Embora se trate de duas populações com evolução paralela, o autor utiliza uma única máquina (Pentium 400 Mhz) para executar todo o algoritmo. A condição de parada do algoritmo foi o tempo

máximo de 1800 segundos para cada instância, executada três vezes cada. Os parâmetros do algoritmo, como número de vezes que cada operador seria aplicado a cada geração foi ajustado intuitivamente, realizando algumas tentativas e testes antes de iniciar o processo de medição dos resultados apresentados. Os parâmetros foram mantidos constantes, exceto para as instâncias C1 e C2, que, segundo o autor, são classes bem distintas (consumidores agrupados) e não têm representado um grande desafio para as diversas alternativas de solução na literatura. Berger [BBB01] obteve seis novos melhores resultados (R108, R110, RC105, RC106, R210 e R211), atualmente, três ainda não foram superados (R108, RC105 e RC106). Entretanto, o principal resultado do trabalho foi relativo ao resultado médio, que é o menor em termos de número de veículos, com um total de 405 para todas as 56 instâncias.

2.3 O Problema de Roteamento de Veículos Dinâmico com Janela de Tempo (PRVDJT)

Na grande maioria das condições reais, os problemas não são estáticos e sim dinâmicos. As informações referentes ao problema, como novos pedidos, cancelamento de entregas e coletas, alteração nos tempos e custos de trechos, mudam a todo instante. Considerar esta dinamicidade na solução do problema pode ser essencial na redução dos custos envolvidos em um processo do mundo real. Entretanto, o número de trabalhos desenvolvidos considerando o problema de roteamento de veículos dinâmico (PRVD) é bem mais reduzido, comparado ao caso estático. Alguns deles são [Psa80] [Psa95] [GP97] [Psa98] [SS98] [LAR00]. Mas avaliando os diversos trabalhos da literatura, conclui-se que não tem existido consenso quanto à formulação do problema ou dos objetivos almejados.

Para diferenciar melhor os problemas estáticos dos dinâmicos, no contexto desta tese, as definições de Larsen são bem adequadas [LAR00]:

Definição 1: Problemas de Roteamento Estáticos

- A. Assume-se que todas as informações relevantes são conhecidas antes do início do processo de roteamento.
- B. Não há alteração das informações relevantes depois de iniciado o processo de roteamento.

Definição 2: Problemas de Roteamento Dinâmicos

- A. Assume-se que não são conhecidas todas as informações relevantes antes do início do processo de roteamento.
- B. As informações relevantes depois de iniciado o processo de roteamento podem sofrer alterações

Uma forma básica para considerar um novo consumidor é reiniciar o algoritmo da situação atual. Esta técnica parece ser a mais simples e natural, quando se faz necessário alterar os dados do problema, devido à alterações ou inclusão de novas informações relevantes. Esta abordagem é freqüentemente chamada de reotimização. Entretanto, mesmo a inclusão de um novo pedido pode ser uma tarefa complicada. Primeiramente, existem restrições nesta segunda fase, que antes não existiam, como por exemplo, as encomendas já podem estar alocadas em determinados veículos, havendo necessidade de se verificar quais são os veículos que podem atender a esta nova demanda. Outro ponto é a própria modelagem adotada. Os veículos podem não mais estarem localizados no depósito central, alterando significativamente o modelo matemático. Adicionalmente, recomeçar a otimização pode requerer um tempo não disponível. Por isso, tem sido comum a utilização de duas técnicas distintas, a primeira adotada para solução do problema estático, onde todos os veículos estão localizados no depósito central, e outra técnica mais adequada à fase seguinte.

Larsen [LAR00] propõe diferentes estratégias para o problema de roteamento de veículos dinâmico, PRVD, baseado no grau de dinamismo. Segundo o autor, cada grau de dinamismo deve ser considerado de forma específica para cada tipo de problema. O autor enfatiza as distintas necessidades do PRVD com ou sem janela de tempo.

Uma outra abordagem é o problema de roteamento de veículos estocástico (PRVE). Este considera as variáveis do problema, como pedidos, custos e disponibilidade de veículos

com uma determinada distribuição de probabilidade. Pode-se considerar que esta é uma opção para tratar alguns tipos de dinâmismos de forma antecipada. Algumas dessas abordagens podem ser encontradas na literatura, como por exemplo, [CR78] [DT86] [Ber92] [BSL96]. No entanto, o modelo estatístico utilizado deve ser muito bem validado, para ser aceitável enviar um veículo onde há uma possibilidade de um consumidor, e não uma demanda real. Este tipo de tratamento é bem adaptado a problemas com alto grau de dinamismo, como por exemplo, onde 90% dos consumidores surgem após o início do roteamento. Os métodos que utilizam esta abordagem são conhecidos como *a-priori methods*.

2.3.1 Modelo em tempo real para PRV dinâmico

Considera-se um modelo em tempo real para o problema de roteamento quando este contempla a chegada e o tratamento, a qualquer momento, de novas informações; adicionalmente, o resultado considerando tais informações deve estar disponível em tempo hábil para a tomada de decisão. Zhu [ZO01] considera uma generalização do PRVJT denominada problema de roteamento de veículos com janela de tempo dinâmico em tempo real (PRVJTDRT) ou *RTVRPTW*. Na sua generalização o autor considera o tempo de trajeto entre os consumidores variando ao longo do tempo por condições do tráfego. Adicionalmente a saída dos veículos do depósito central podem ocorrer a qualquer instante. O autor utiliza uma abordagem por melhoria contínua, fazendo uso de alguns operadores comuns na literatura. Uma solução básica inicial é continuamente melhorada por heurísticas de busca local. Sempre que as condições estão estáveis, ou seja, enquanto não há novas informações a serem processadas, a heurística de busca local trabalha para aperfeiçoar as rotas. Os operadores utilizados incluem a troca dos consumidores entre diferentes rotas, troca de posição entre dois consumidores (*swap*) e o conhecido operador *3-opt*. Uma heurística de retirada e re-inserção de consumidores, utilizando uma função custo que inclui a prioridade de atendimento entre os consumidores, também foi utilizada. Novos consumidores são inseridos a qualquer momento, utilizando a mesma função custo.

Uma metodologia comumente utilizada no problema dinâmico é denominada inserção e melhoramento ou *insert and improve*. Estes métodos, por construção, se adaptam naturalmente ao surgimento de nova informação ou pedidos ao sistema. A solução é

continuamente melhorada, a cada surgimento de um novo pedido. Atualmente, seus resultados para os problemas estáticos são considerados medianos se comparados com metodologias mais atuais, como as metaheurísticas. Entretanto ainda é muito utilizada como parte da solução.

[KPS98] descreve a vantagem de se utilizar uma estratégia de melhoramento contínuo da solução, ou seja, no seu estudo, uma instância de 100 visitas ao dia, poderá ser resolvida utilizando todo o tempo do dia disponível. Neste mesmo artigo é descartado o uso de uma metaheurística para o problema dinâmico, sob a argumentação de que é indesejado um retrocesso no valor da função objetivo, fato inevitável, segundo o autor, utilizando tais métodos. Neste trabalho será demonstrado que isso pode ser evitado, conseqüentemente, as metaheurísticas continuam sendo uma opção para os problemas dinâmicos.

Na estratégia de [KPS98], do tipo inserção e melhoria, é utilizada uma inclusão de 10 localidades a cada passo e em seguida aplicam-se as técnicas de melhoria *2-Opt* [Lin65], *Or-opt*, *Relocate* [OR76], *Exchange* [TAI94] e *Cross* [PR95]. No processo dinâmico de chegada de novos pedidos, o autor utiliza o conceito de passo de tempo. Durante um passo de tempo não há inserção de novos pedidos. Se no final do intervalo houver novos pedidos, os anteriores são fixados na melhor solução conhecida. O autor avalia a qualidade de sua estratégia adaptando ao caso dinâmico, problemas testes estáticos amplamente adotados na literatura, como em Taillard [TAI94], Christophides [NB84] e Fisher [FJW81]. A maior instância utilizada é de 385 localidades. No entanto, não é possível avaliar, pelos dados apresentados, a performance do algoritmo em termos de distância da solução ótima ou outros trabalhos que tenham utilizado as mesmas instâncias. A contribuição mais significativa está na descrição do comportamento do custo da entrega com o aumento da dinamicidade do problema. Segundo os resultados ali apresentados, o custo da entrega aumenta significativamente com o aumento da dinamicidade, ou seja, mais entregas surgidas após o início do roteamento, não conhecidas *a priori*.

2.3.2 Outros pontos críticos do roteamento dinâmico

Existem também alguns outros fatores importantes no tratamento dos problemas dinâmicos, a saber:

Coleta e Entrega: Quando o tipo de serviço no PRVJTD são coletas, estas podem ser realizadas a qualquer momento, mesmo o veículo já em trânsito, desde que exista ainda capacidade de carga, pois não há necessidade de voltar ao depósito central. Situação bem diferente ocorre em novas entregas. Neste caso, dependendo do tipo de serviço, poderá haver a necessidade de coletar a encomenda primeiro, geralmente no depósito central.

Conhecimento das Demandas: o conhecimento de novas demandas pode ocorrer em tempo real ou em intervalos específicos. Por exemplo, um veículo pode estabelecer contato com a central somente durante intervalos pré-definidos, havendo temporariamente desconhecimento de parte dos consumidores a serem atendidos.

Atualização Cíclica ou em Tempo Real: novos pedidos, novas variáveis ou restrições podem surgir a qualquer momento. É comum, no entanto, aguardar um determinado período de tempo fixo antes de atualizar as variáveis e mudar a busca para uma nova solução. Isso pode ser feito visando agrupar uma certa quantidade de novas variáveis e novos pedidos, para que as alterações possam justificar as alterações nas rotas atuais.

Preemptivo e Não-preemptivo: é a possibilidade ou não de um veículo alterar a sua tarefa no meio do caminho. Imagine que um veículo sai de uma localidade em direção a outra, com uma entrega específica. O surgimento de uma necessidade (uma coleta, por exemplo) em uma localidade nas margens do trecho onde se encontra o veículo, poderá ou não ser atendida antes do atual destino: se este atendimento é considerado e possível, o sistema é dito preemptivo.

Um aspecto não trivial em problemas dinâmicos é o tratamento da função objetivo. Para simplificar, geralmente considera-se o problema dinâmico como uma seqüência de problemas estáticos.

2.3.3 Métodos Exatos para o Problema de Roteamento de Veículos Dinâmico

Larsen [LAR00] na conclusão de sua tese de doutorado resume bem a situação dos métodos exatos para o PRV dinâmico. Segundo o autor, “o rápido desenvolvimento no hardware disponível somado ao grande avanço dos software de otimização, tem alterado as

fronteiras do que pode ser resolvido por métodos de otimização em quantidade razoável de tempo. No entanto, mesmo problemas estáticos de roteamento do mundo real têm sido computacionalmente difíceis para tais métodos”. Com isso, hoje é difícil imaginar um método de otimização para problemas dinâmicos tratados em tempo real. A menos que o desafio da classe NP-difícil seja quebrado com algum fato novo que hoje não é previsto ou que P seja igual a NP, o que parece improvável, não é razoável, por enquanto, desejar a solução ótima para o problema dinâmico em instâncias mesmo de tamanho moderado.

Capítulo 3

Minimizando a Distância Percorrida no PRVJT Estático

Este capítulo apresenta uma heurística baseada no modelo de particionamento de conjunto e no método de geração de colunas para o PRVJT estático, minimizando a distância total percorrida pelos veículos. Na seção 3.1 a heurística proposta é contextualizada na literatura atual e suas principais motivações são apresentadas. Na seção 3.2 o modelo de particionamento de conjuntos utilizado é apresentado. A seção 3.3 contém a descrição da heurística e suas partes, incluindo o algoritmo genético utilizado e seus operadores de busca local. Encerrando o capítulo, as conclusões são apresentadas na seção 3.4.

3.1 Introdução

O problema de roteamento de veículos com janela de tempo (PRVJT) tem sido abordado na literatura utilizando métodos exatos e heurísticas. O conjunto de instâncias de Solomon [SM87] tem sido amplamente adotado para avaliação do desempenho das propostas em ambas as alternativas. Entretanto, enquanto nos trabalhos utilizando métodos exatos a função objetivo tem sido a minimização da distância total percorrida (DT) pelos veículos, naqueles utilizando métodos heurísticos a minimização do número total de veículos utilizados na solução (NV) tem sido prioridade. Neste último caso DT segue como segundo objetivo hierarquicamente após NV ter sido minimizado. Quanto ao critério de precisão nos cálculos também não tem havido consenso. Novamente, as heurísticas seguem uma estratégia, realizando todos os cálculos com dupla precisão em ponto flutuante, e os métodos exatos outra, arredondando ou truncando os cálculos para números inteiros, geralmente após a multiplicação dos dados de entrada por dez (obtendo uma casa decimal de precisão).

A inviabilidade de se optar por um método exato para solução do PRVJT tem impulsionado diversos estudos de diferentes heurísticas, na tentativa de encontrar uma solução próxima do ótimo em tempo aceitável. Entretanto, poucos estudos utilizando heurísticas foram publicados minimizando a distância total percorrida. Os únicos que se têm conhecimento são [RGP99] [TBG+97] e [TLZ+01]. Conseqüentemente, os resultados das heurísticas não foram efetivamente comparados com os ótimos produzidos pelos métodos exatos.

Um outro ponto a ser considerado é que a distância total percorrida pode ser bem menor quando não se utiliza um número de veículos já minimizado. Conseqüentemente, os custos relacionados à entrega de mercadorias pode ser menor se a minimização da distância for tratada como prioridade. Isso acontece, por exemplo, quando já existe uma frota de veículos com quantidade superior ao número mínimo necessário para uma dada demanda de entregas ou coletas.

Além dos fatores mencionados acima, um outro importante, que justifica o empenho do estudo de uma nova heurística para o tratamento do PRVJT, é a necessidade de um método robusto, onde a qualidade dos resultados não varie demasiadamente de uma instância para outra, ou mesmo em duas execuções diferentes para um mesmo problema. Neste sentido a proposta híbrida a seguir apresenta um dos seus pontos mais promissores.

Por fim, a proposta apresentada deverá ter resultados positivos tanto para o problema de roteamento estático quanto para o dinâmico. No problema dinâmico, conforme discutido na seção 2.3, não se conhecem todas as variáveis do problema no início do processo de otimização. Embora já existam algumas propostas para o problema dinâmico a avaliação dos resultados é incipiente. Primeiro porque é muito difícil medir a qualidade dos resultados neste caso, e também pela falta de “problemas teste” com resultados já bem trabalhados (*benchmarks*). A heurística aqui proposta, HGC_DT será avaliada neste capítulo exclusivamente para o caso estático. Uma vez que seus resultados tenham sido positivos, justificar-se-á aplicá-la ao problema dinâmico.

3.2 O Modelo de Particionamento de Conjuntos para o PRVJT

A abordagem proposta para minimização da distância total percorrida no PRVJT estático utiliza o modelo do problema por particionamento de conjuntos (PPC) já descrito, conforme equações (3.1) a (3.3) abaixo:

$$\text{Minimize } \sum_{r \in R} c_r x_r \quad \text{sujeito a} \quad (3.1)$$

$$\sum_{r \in R} \delta_{ir} x_r = 1, \quad \forall i \in C \quad (3.2)$$

$$x_r \in \{0,1\} \quad (3.3)$$

Onde

c_r = distância euclidiana total percorrida, partindo do depósito central, passando pelos consumidores da rota r e retornando ao depósito central;

x_r = variável de decisão. Igual a 1 se a rota r faz parte da solução e 0 caso contrário;

δ_{ir} = parâmetro auxiliar, que define os consumidores presentes na rota r . É igual a 1 se o consumidor i está na rota r e 0 caso contrário.

R' = conjunto de rotas viáveis. $R' \subseteq R$, onde R é o conjunto de todas as rotas viáveis para o problema. Se $R' = R$ então a solução para o problema acima conduz a solução exata para o PRVJT.

O método exato de geração de colunas permite que rotas (colunas) sejam deixadas de fora do problema principal, equação (3.1), porém, deverá haver garantia de que todas aquelas com redução de custo relativo, no contexto de Dantzig-wolf, já foram inseridas no problema. Entretanto, o subproblema de encontrar tais rotas, também conhecido como *pricing problem*, já foi provado ser também NP-completo no sentido forte [KO95]. Conseqüentemente, também a abordagem por geração de colunas é inviável para instâncias com um número maior de consumidores, conforme já foi discutido no Capítulo 2. Nestes casos, uma das opções é fazer uma busca heurística por rotas que podem gerar redução de

custo da função objetivo, não oferecendo garantia que todas as rotas com redução de custo negativa serão inseridas no problema, o que torna todo o algoritmo uma heurística.

3.3 Uma Heurística de Geração de Colunas utilizando o Algoritmo Genético

A heurística desenvolvida foi baseada no método de geração de colunas, denominada por isso HGC_DT (Heurística por Geração de Colunas para minimização da Distância Total percorrida). A principal motivação é a constatação de que um mínimo local para o PRVJT produzido por uma heurística qualquer possui algumas rotas muito semelhantes ou até mesmo idênticas às aquelas presentes na solução ótima. Esta comparação pode ser feita observando soluções ótimas conhecidas por métodos exatos e mínimos locais produzidos por diferentes heurísticas. Se as rotas que fazem parte da solução ótima podem ser produzidas, mesmo que isoladamente em diferentes mínimos locais, todas estas rotas podem formar um conjunto R' de tamanho moderado e um algoritmo de programação linear inteira poderá resolver o modelo de particionamento de conjunto previamente descrito de forma exata, produzindo uma solução ótima. Não há nenhuma garantia real que tais rotas serão encontradas nos mínimos locais, porém caso poucas rotas da solução ótima procurada fiquem de fora, ter-se-á produzido um ótimo local de boa qualidade. Outra característica importante observada é que diferentes mínimos locais produzidos por diversas execuções de métodos heurísticos não produzem resultados idênticos. A princípio este fato é considerado simplesmente falta de robustez da heurística. No entanto, pode também ser explorado de forma positiva no PRVJT, pois as características destes mínimos permitem a geração de um conjunto de rotas que pode ser adequado ao modelo por particionamento de conjunto proposto, pois aumentam a possibilidade das rotas desejadas serem incluídas no conjunto R' .

Rochat [RT95] já verificou estas características do PRVJT, porém utilizou o modelo de particionamento de conjunto somente em um pós-processamento, segundo o qual foi responsável apenas por melhorar suavemente as soluções previamente encontradas por sua heurística baseada na busca Tabu. As rotas usadas por Rochat foram aquelas produzidas ao longo da busca Tabu. De forma diferente, a solução apresentada neste trabalho se baseia no

fato de que é preferível gerar uma quantidade maior e diversificada de diferentes mínimos locais, sem tomar muito tempo na tentativa de escapar desses mínimos, e deixar o restante do trabalho para os pacotes disponíveis de programação linear inteira. Para geração dos diferentes mínimos locais foi escolhido o Algoritmo Genético (AG), por ter apresentado bons resultados isoladamente para o PRVJT, que será apresentado nas seções seguintes. Para solução do PPC, o software livre GLPK foi utilizado.

3.3.1 Preprocessamento

O processamento prévio, evitando ao máximo o uso de CPU para operações repetitivas, limitou-se a uma pequena modificação na construção da matriz de adjacência, com o cálculo das distâncias entre os consumidores do problema. Existem pares de consumidores que a presença simultânea em uma mesma rota é inviável, devido a restrições das janelas de tempo impostas pelos mesmos. As janelas de tempo e as distâncias entre todos os pares de consumidores foram avaliadas, procurando por pares de consumidores incompatíveis. A saída de um consumidor i no instante de tempo $a_i + ts_i$ representa a melhor condição, uma vez que a_i representa o início da janela de tempo para o consumidor i ser atendido e ts_i o tempo mínimo necessário naquele consumidor. Aquele par de consumidores $[i,j]$ onde o instante ideal de partida do veículo $a_i + ts_i$, somada ao tempo de trajeto t_{ij} supera o limite superior da janela de tempo em j , ou seja, $(a_i + ts_i + t_{ij}) > b_j$, representa um arco ou trecho inviável. Nestes casos a distância foi substituída por uma constante grande, *BIG_NUMBER*. Ao encontrar qualquer distância *BIG_NUMBER*, a verificação da viabilidade de qualquer rota é interrompida, reduzindo alguns cálculos posteriores. Embora represente uma pequena economia em tempo de processamento, estas são operações muito repetitivas durante a avaliação de novas rotas, geradas a todo o momento por diversos operadores de busca local no AG.

3.3.2 Encontrando mínimos locais

O principal desafio, adequando ao modelo do PPC utilizado, equações (3.1) a (3.3), é gerar rotas de boa qualidade para o problema principal. No entanto, a tarefa de gerar rotas de boa qualidade não é fácil. Várias heurísticas clássicas, geralmente chamadas de heurísticas construtivas para o problema de roteamento, se prestam a gerar e aperfeiçoar rotas. Um

exemplo é o *PFIH* [SM87], que será descrito posteriormente. No entanto, nota-se que quando rotas individuais são produzidas, sem considerar o problema como um todo, existe pouca chance de haver uma combinação adequada das mesmas, reduzindo efetivamente a função objetivo. Conforme já é amplamente conhecido, a grande dificuldade do PRVJT está na restrição da equação (2.2) do modelo original [LA99] [KO95], que cria a interdependência entre os diversos veículos e diferentes consumidores. Por isso não existe um conceito de rota de boa qualidade para o PRVJT isoladamente, uma vez que um custo adequado em uma rota, mesmo com muitos consumidores poderá ocasionar condições inadequadas para o atendimento dos demais consumidores. Uma rota só pode ser avaliada no conjunto das outras. Conseqüentemente, as rotas de utilidade, que devem ser inseridas no conjunto R' do PPC, são aquelas que fazem parte de soluções completas: mínimos locais.

Algumas heurísticas têm sido propostas para geração de colunas [LLS98], mas não se tem conhecimento da utilização do algoritmo genético com esta finalidade para o PRVJT. Este algoritmo, conforme foi discutido, tem sido um dos métodos responsáveis pelos melhores resultados na minimização do número de veículos para as instâncias de Solomon.

O objetivo é a utilização de um algoritmo genético para geração rápida de mínimos locais de qualidade razoável, que uma vez utilizado em conjunto com um pacote disponível de programação linear inteira, como o GLPK, poderá fornecer a qualidade e robustez necessária, possibilitando a obtenção de novos resultados para as instâncias de Solomon no critério minimização da distância total percorrida.

O Algoritmo Genético

O algoritmo desenvolvido, denominado AG_DT, por ter como objetivo a redução de DT, se baseia nos princípios básicos dos algoritmos genéticos introduzidos no capítulo 2. Uma população de I indivíduos soluções para o PRVJT é gerada utilizando uma versão modificada estocástica do algoritmo *PFIH* proposto por Solomon [SM87]. Todos os indivíduos correspondem a soluções viáveis para o problema. Por isso, todas as operações como mutação e recombinação produzem sempre indivíduos também válidos. O algoritmo foi desenvolvido para corresponder, de forma mais completa possível, às seguintes expectativas descritas a seguir.

1) Operadores capazes de permitir movimentos de forma mais completa possível para todo o espaço de busca

A capacidade de “escapar” dos mínimos locais é a principal característica almejada pelas metaheurísticas. O algoritmo genético possui basicamente duas estratégias para efetivamente possibilitar isso. A primeira é manter múltiplas soluções sendo trabalhadas simultaneamente. Uma vez que a melhor solução atual, representada pelo indivíduo de maior adaptabilidade, encontra um mínimo local, sua capacidade de gerar descendentes com qualidade pode ficar dificultada. Neste caso, outros indivíduos, que até então eram medianos, podem tomar a frente na geração de descendentes de qualidade, representando a busca em outros locais do espaço de soluções. A segunda estratégia é a possibilidade de sair de mínimos locais através das mutações. Uma mutação modifica alguma característica de forma aleatória ou heurística, na tentativa de melhorar a solução, ou mesmo iniciar a busca em locais ainda não explorados do espaço de busca.

A primeira estratégia descrita é natural no algoritmo genético e para ser aproveitada deve haver diversidade suficiente na população de indivíduos, representando regiões promissoras diferentes. Uma das formas de controlar a diversidade é através da pressão seletiva sobre a evolução, que corresponde a maior ou menor chance de um indivíduo mediano ou mesmo ruim ser selecionado para participar da geração seguinte ou para um cruzamento. Se somente os indivíduos com elevada adaptabilidade são selecionados, diz-se que a pressão seletiva está muito forte, podendo ocasionar uma rápida convergência no mínimo local atual encontrado.

Quanto aos operadores utilizados na mutação, um cuidado maior deve ser tomado. O conjunto de operadores deve mapear de forma mais completa possível o espaço de busca. Se este mapeamento é incompleto, uma região de boa qualidade pode ficar inatingível a partir da população atual, o que representa uma maior possibilidade de estagnar o processo de evolução.

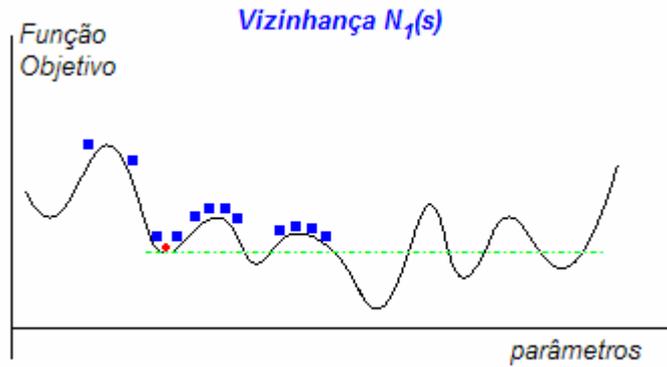


Figura 3.1: O círculo corresponde a melhor solução, de um conjunto de soluções atuais s (população atual). Os quadrados representam toda a vizinhança $N_1(S)$ da população atual. A linha tracejada representa o valor da função objetivo em s^* (melhor indivíduo), separando regiões do espaço de busca em melhores (inferior a reta) e piores. Não existem soluções definidas na vizinhança $N_1(S)$ que melhoram a solução já encontrada.

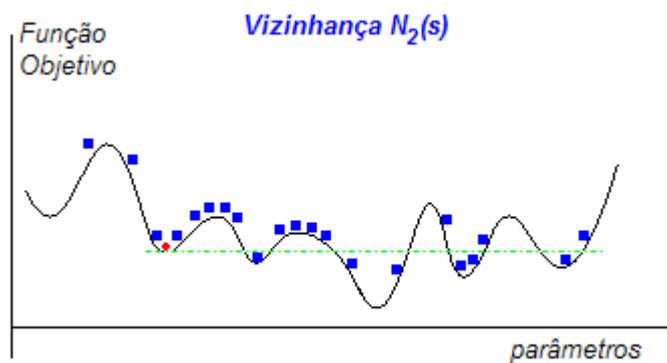


Figura 3.2: Os quadrados representam uma vizinhança $N_2(S)$ mais abrangente que $N_1(S)$ da Figura 3.1. Ao contrário, agora existem soluções definidas na vizinhança $N_2(S)$ que melhoram a solução já encontrada, embora nenhuma delas ainda seja o mínimo global.

As Figuras 3.1 e 3.2 representam duas possibilidades distintas de geração de novas soluções através de diferentes operadores de mutação, a partir de uma mesma população do algoritmo genético. Aqui, para efeito de simplificação só é mostrada a melhor solução (representada pelo ponto em círculo), ou seja, o indivíduo com a melhor adaptabilidade $s^* \in S$, sendo S uma geração de indivíduos soluções qualquer. O conjunto $N_1(S)$ representa todas as soluções possíveis de serem geradas a partir de S . Na Figura 3.1 não há soluções vizinhas, do conjunto vizinhança disponível, que podem ser geradas a partir da população atual (representadas por quadrados) com redução de custo. Isso significa que o conjunto de operadores utilizados para definir $N_1(S)$ é pobre. Para que a evolução representada pelo AG

da Figura 3.1 escape do mínimo local atual, várias gerações poderão ser necessárias, até que um indivíduo qualquer, diferente do melhor (já que este já foi considerado anteriormente) possa mapear alguma das regiões abaixo da linha tracejada. Já o conjunto de operadores que definem $N_2(S)$ é melhor, pois indivíduos que superam a solução atual podem ser gerados. Eles não necessariamente serão gerados na próxima geração, uma vez que nem todos os pontos possíveis em $N_2(S)$ são utilizados. Porém, o conjunto de operadores que definem $N_2(S)$ não irá impedir que isso aconteça, aumentando a possibilidade de escapar de mínimos locais. Embora não se possa garantir que no primeiro caso haverá estagnação e também não há garantia que no segundo o mínimo local será evitado, a probabilidade de melhores soluções no segundo é evidente.

Entretanto, a questão é como projetar ou mesmo assegurar que um conjunto de operadores é suficientemente abrangente a ponto de permitir que qualquer região do espaço de busca seja pertencente ao conjunto possível de soluções $N(S)$. Por outro lado, deverá haver o cuidado para não gerar um conjunto de operadores abrangente, porém ineficiente, aproveitando pouco as características já alcançadas pelo indivíduo solução atual. Um exemplo extremo seria um operador que refaz todas as rotas. Embora qualquer região possa ser atingida a possibilidade desta estratégia ser proveitosa é pequena. Tem sido comum diversas propostas na literatura de operadores específicos para problemas de roteamento. Muitas vezes se tratam de operadores heurísticos que testam de forma exaustiva uma região considerável na vizinhança da solução atual. Neste trabalho são testados alguns destes operadores, porém outros novos são propostos, uma vez que os objetivos aqui são diferentes, tanto da função objetivo, que é a minimização de distância e não o número de veículos, como também a necessidade de que o algoritmo genético desenvolvido encontre um mínimo local de qualidade em um tempo suficientemente curto para que possa ser executado diversas vezes.

2) *Função adaptabilidade adequada facilitando a seleção dos indivíduos*

A função objetivo utilizada no modelo, conforme equação (3.1), tem como único objetivo a redução da distância total percorrida no PRVJT. No entanto, muitas vezes existem outras características em uma solução, representadas no indivíduo do AG, que devam ser valorizadas na função adaptabilidade. Um exemplo é o número de veículos utilizados numa solução. Duas soluções com valores de distância percorrida próximos, mas com

número de veículos bem diferentes, podem apresentar perspectivas de redução da distância percorrida diferentes. Nestes casos, o número de veículos, que nada tem a ver com a função objetivo do problema, pode ser utilizado na função adaptabilidade do problema, indicando uma região de busca apropriada. Complexas funções de adaptabilidade têm sido propostas na literatura, principalmente quando o objetivo é o número de veículos. Quanto ao número de veículos e a distância percorrida, verifica-se que são objetivos colaborativos na fase inicial da busca, enquanto as soluções estão mais distantes do ótimo, e se tornam altamente concorrentes nas fases finais, onde geralmente a redução de NV provoca elevação de DT e vice-versa. Conseqüentemente, a escolha adequada da função adaptabilidade pode ter impacto decisivo no desempenho do algoritmo.

3) *Cruzamento heurístico*

Muitos trabalhos na classe dos algoritmos evolutivos têm suprimido a etapa de recombinação ou cruzamento dos indivíduos (se a recombinação é eliminada não se atribui normalmente o nome de algoritmo genético, mas sim de programação evolucionária ou algoritmos evolutivos). Por exemplo, [HG99] em sua estratégia evolutiva ES1, que trata da minimização da distância percorrida, não utiliza recombinação, gerando os descendentes simplesmente pela aplicação do conjunto de operadores de mutação na população atual. Isso ocorre porque existe uma grande dificuldade de realizar uma combinação de dois ou mais indivíduos gerando uma nova solução viável que mantenha ou melhore o nível da adaptabilidade dos descendentes. A recombinação de indivíduos, ou de soluções candidatas ao ótimo da função, deve representar uma busca na direção representada por estes indivíduos, cada qual contribuindo um pouco com suas características. Para o PRVJT é natural imaginar um descendente pela composição de rotas dos pais. O problema está no fato que após algumas rotas diferentes terem sido extraídas de cada um é comum restarem rotas incompatíveis. Nestas rotas estão alguns consumidores que já foram incluídos através das rotas do outro indivíduo pai. Se o número de consumidores restantes é elevado, cria-se uma situação difícil, de como incluir os consumidores restantes no descendente. Este processo de inserção dos consumidores restantes nas rotas já existentes ou a criação de novas rotas pode prejudicar a qualidade da solução ali representada. É difícil um procedimento eficiente para esta tarefa, sendo comum a elevação do valor da adaptabilidade e, conseqüentemente da função objetivo. Entretanto, o cruzamento

representa uma vantagem se eficiente, pois a combinação de duas soluções de qualidade para o PRVJT é uma região muito promissora de busca, como o próprio modelo PPC indica, considerando R' o conjunto de rotas de dois indivíduos pais. Neste trabalho um cruzamento eficiente foi proposto.

4) Soluções rápidas

Para a estratégia aqui proposta, o algoritmo genético é executado diversas vezes, gerando vários mínimos locais, cada qual gerando um conjunto de rotas para o modelo de geração de colunas proposto. Por um lado, se o algoritmo genético permanecer na busca de um mínimo local melhor por muito tempo, poucas soluções são geradas no intervalo de tempo disponível, reduz-se com isso a probabilidade de inserir todas as rotas (colunas) necessárias no conjunto R' do PPC. Por outro lado, muitas rotas de qualidade ruim resultarão em uma combinação final também ruim. É fato que se a heurística for suficientemente boa, uma única solução produzida poderá ter as poucas rotas importantes, mas geralmente as heurísticas têm dificuldade de manter esta qualidade para diferentes problemas e diferentes execuções. É justamente esta robustez buscada pelo esquema proposto. Entretanto, não é simples gerar muitas soluções suficientemente razoáveis em curto intervalo de tempo. O algoritmo desenvolvido deve estar em um ponto intermediário dos dois extremos, produzindo uma rápida convergência com qualidade, tentando chegar o mais próximo do ótimo possível em alguns poucos minutos ou até mesmo segundos.

Representação dos Indivíduos

Os indivíduos e a população são representados por estruturas hierárquicas compostas. A estrutura básica é o veículo, que contem um conjunto ordenado de números inteiros, representando os consumidores. O depósito central é omitido no veículo, pois está implícito que toda rota tem sua origem e término no mesmo. Uma variável atualizada com o número de consumidores presentes em cada veículo é usada para aumentar o desempenho nos cálculos e na identificação dos veículos vazios.

```

TVeículo = registro           { definição do tipo TVeículo }
    NConsumidores = inteiro;    { número de consumidores atendidos pelo veículo }
    VConsumidores = vetor de inteiro; { consumidores atendidos }
fim registro;

```

Um indivíduo do AG_DT é formado por um vetor de veículos, que representam um conjunto de rotas, proposta de solução para o PRVJT. Por isso algumas vezes o indivíduo é chamado de *indivíduo solução* do problema. Para garantir uma solução sempre viável, uma heurística de construção é utilizada para gerar a população inicial, conforme será mostrado. O número de veículos é considerado ilimitado inicialmente. Como foi dito, mesmo encontrar uma solução viável para um número fixo de veículos para o PRVJT é um problema *NP-difícil*. Abaixo é mostrada a estrutura descrita para representação dos indivíduos do AG_DT.

```
TIndivíduo = registro           { definição do tipo TIndivíduo }
    VVeículos = vetor de TVeículo; { vetor de veículos que compõe uma solução }
fim registro;
```

Por fim, uma população é um vetor de indivíduos soluções, conforme estrutura abaixo.

```
TPop = registro           { definição do tipo TPop (população) }
    NIndivíduos = inteiro;    { número de indivíduos na população }
    VIndivíduos = vetor de TIndivíduo; { vetor de indivíduos da população }
fim registro;
```

Para maior entendimento, a representação do indivíduo pode ser mostrada em forma de matriz, conforme a Figura 3.3. Como pode ser visto, os indivíduos são representações diretas de soluções viáveis para o PRVJT, sem nenhuma necessidade de codificação binária ou outra qualquer.

5	1	5	10	12	15
2	14	11			
3	9	2	4		
5	6	3	8	7	13

Figura 3.3: Representação de um indivíduo. Cada linha representa um veículo, a primeira coluna indica quantos consumidores são atendidos pelo veículo. As demais colunas contêm os consumidores em ordem de atendimento. O depósito central é omitido.

Geração da População Inicial

A geração de um conjunto de soluções iniciais, que representam os indivíduos da primeira geração do AG_DT desenvolvido é feita observando os critérios de diversidade e qualidade inicial, a saber:

- *Diversidade*: o processo deve ser suficientemente aleatório para que a cada execução, diferentes populações iniciais sejam produzidas, evitando que a busca inicie sempre dos mesmos pontos do espaço de busca, uma vez que são desejados diferentes mínimos locais (diferentes rotas de qualidade) para compor o conjunto R' do modelo de particionamento de conjuntos utilizado como problema principal.
- *Qualidade Inicial*: um processo totalmente aleatório produz soluções iniciais muito ruins, onde a distância percorrida e o número de veículos são exagerados. Consumidores nos extremos da região podem ser colocados juntos na mesma rota, forçando rotas com poucos consumidores. Uma heurística inicial básica e rápida deve ser utilizada para que o AG_DT possa partir de soluções razoáveis, reduzindo o tempo necessário de convergência do algoritmo.

Muitos autores têm utilizado a heurística proposta inicialmente por Solomon [SM87], denominada *PFIH (Push Forward Insertion Heuristic)* para gerar uma solução inicial de qualidade. A *PFIH* possui um mecanismo eficiente de calcular o custo de se inserir um novo consumidor em uma rota, conforme será mostrado.

Heurística Push Forward Insertion Heuristic (PFIH) original

Considere uma rota qualquer com m consumidores, $r_q=(m,\dots,n)$, onde m é o primeiro consumidor atendido e n é o último. O depósito central não é considerado na representação uma vez que sempre deve ser a origem e o destino final em qualquer rota. Considere também, $[a_m, b_m]$ e $[a_n, b_n]$ os intervalos que definem a janela de tempo do primeiro e último consumidor da rota r_q , respectivamente. Lembrando que o intervalo corresponde aos horários inicial e final dentre os quais o veículo deve chegar ao consumidor. A chegada pode acontecer antes, porém será necessário esperar para o início do serviço. O *PFIH* avalia a viabilidade de inserir um novo consumidor na rota r_q inserindo-o em todas as posições possíveis, da posição inicial antes de m , até a última posição depois de n , sendo

selecionada aquela de menor aumento na distância percorrida na rota r_q . Se o consumidor é inserido entre os consumidores i e j qualquer, o tempo de chegada t_j ao consumidor j irá sofrer um deslocamento para frente, podendo causar o mesmo para todos os consumidores seguintes (*Push Forward*). Nada precisa ser avaliado do consumidor m ao i . Caso a chegada em algum consumidor posterior exceda o limite superior da janela de tempo do mesmo, a inserção de tal consumidor é inviável naquela posição da rota r_q . Por outro lado, uma espera pode estar ocorrendo em algum consumidor da rota r_q , sendo então o valor do deslocamento de tempo neste consumidor igual a zero. A partir deste consumidor, os tempos de chegada do veículo nos consumidores seguintes permanecem também inalterados. No pior caso, todos os consumidores serão avaliados.

A heurística *PFIH* inicia uma rota com um único consumidor, escolhido conforme o custo de inicialização c_i , de um consumidor i , dado pela equação (3.2), abaixo.

$$c_i = -\alpha d_{0i} + \beta b_i + \gamma((p_i / 360)d_{0i}) \quad (3.2)$$

Onde:

α = 0.7 (calculado empiricamente por Solomon [SM87]);

β = 0.1 (calculado empiricamente por Solomon [SM87]);

γ = 0.2 (calculado empiricamente por Solomon [SM87]);

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

b_i = limite superior da janela de tempo de chegada ao consumidor i ;

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

A partir do primeiro consumidor escolhido, conforme equação (3.2), os demais são testados um a um, entre cada posição possível na rota em construção. A posição e o consumidor que resultarem no menor acréscimo da distância total percorrida, sem violação de capacidade e janela de tempo é escolhido. Após não mais haver consumidores possíveis para inserção (ainda não utilizados) na rota em construção, esta é encerrada e inicia-se

novamente o mesmo processo com uma nova rota vazia, sendo o primeiro consumidor aquele de menor custo em (3.2), dentre os que ainda não foram roteados.

PFIH estocástico

O algoritmo proposto neste trabalho para o PRVJT, considerando minimização de distância, foi um *PFIH* simplificado e estocástico. Verifica-se que *PIFH* original proposto por Solomon é determinístico. É possível observar que a escolha do consumidor inicial tem um efeito muito grande na heurística proposta, pois o aumento na distância percorrida de cada consumidor vai depender dos consumidores já inseridos. Depois de definido o primeiro consumidor de uma rota, o consumidor seguinte é determinístico para uma determinada instância. Como a proposta foi gerar diferentes mínimos locais, diferentes populações iniciais são necessárias. Adicionalmente, para uma única população inicial será necessário um conjunto de indivíduos diferentes. Por isso, optou-se por escolher aleatoriamente os consumidores, tanto o primeiro, na criação de uma nova rota, quanto os demais, não fazendo uso da equação (3.2). O custo de inserção de cada consumidor foi exclusivamente o aumento da distância total percorrida da rota em questão. Isso possibilita soluções iniciais de boa qualidade e diferentes entre si, condição fundamental para compor os diversos indivíduos da geração inicial do algoritmo genético utilizado. O algoritmo proposto pode ser visto a seguir, Algoritmo 3.1

Procedimento *Gera_Individuo_População_Inicial (Indiv);*

Início

NCons := 0;

NRota := 0;

enquanto *NCons < Nconsumidores* **faça**

início

i = Escolhe_Consumidor_Aleatoriamente;

Incrementa(NRota);

Incrementa(NCons);

Indiv.Rota[NRota] = Inicia_Rota(i);

FimRota = falso;

enquanto não *FimRota* **faça**

início

i = Escolhe_Consumidor_Aleatoriamente;

pos = Posição_Menor_Custo_DT(Indiv.Rota[NRota],i);

se pos = -1 então *FimRota = Verdadeiro*

senão início

Insero_Consumidor(i,pos, Indiv.Rota[NRota]);

Incrementa(NCons);

fim senão;

fim enquanto;

fim enquanto;

fim procedimento.;

Algoritmo 3.1: PFIH Estocástico gera soluções iniciais baseando-se no PFIH original de Solomon [SM87] para a população inicial do AG_DT.

Adaptabilidade e Seleção de Indivíduos

Muitas vezes a adaptabilidade utilizada no algoritmo genético precisa conter ingredientes, além do valor da função objetivo, para ser possível uma seleção mais eficiente e identificar os indivíduos mais promissores. Nestes casos, dentre os indivíduos que já apresentam o melhor valor da função objetivo, aqueles que também correspondem a uma maior perspectiva de melhora (baseado em algum critério, geralmente heurístico) deverão ser valorizados. Isto fica muito evidente no capítulo seguinte, onde é descrita a adaptabilidade para a minimização do número de veículos, seguido hierarquicamente da distancia percorrida. Já para a minimização da distância total percorrida, uma função de adaptabilidade dos indivíduos bem simples foi suficiente. A adaptabilidade representa o inverso da distância total percorrida por todos os veículos no indivíduo solução (R_s), ou seja, a soma das distâncias de toda rota $r \in R_s$

$$Adaptabilidade_s = \frac{1}{\sum_{r \in R_s} c_r} \quad (3.3)$$

Uma vez que os indivíduos tenham sido avaliados, o AG_DT implementado realiza uma seleção dos pares de indivíduos que através dos cruzamentos irão gerar descendentes para a geração seguinte.

Duas formas de seleção de indivíduos foram avaliadas. A primeira é conhecida por roleta giratória [GD89], conforme descrição a seguir.

Roleta Giratória

A roleta giratória é um processo de seleção de indivíduos probabilístico, influenciado pela adaptabilidade relativa dos mesmos. A adaptabilidade relativa dos indivíduos é calculada dividindo-se a adaptabilidade absoluta, dada pela (3.3), pela soma das adaptabilidades absolutas de toda a população. Os valores maiores da adaptabilidade relativa irão resultar em chances diretamente proporcionais de escolha dos indivíduos para seleção. O processo pode ser representado por uma roleta giratória de área igual a 1, conforme Figura 3.4. Cada indivíduo tem uma fatia de área da roleta igual a sua adaptabilidade relativa. Um processo randômico deve simular um giro na roleta. Um apontador irá indicar o indivíduo escolhido quando a roleta parar. A probabilidade de indicação de cada indivíduo é diretamente proporcional a sua adaptabilidade.

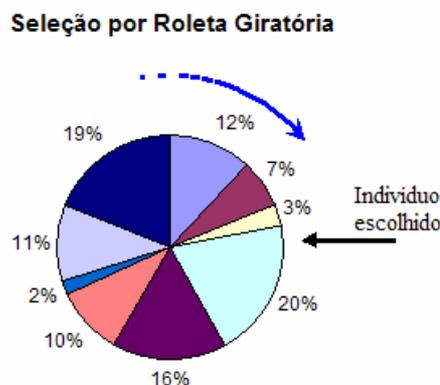


Figura 3.4: Visualização do processo de seleção por roleta giratória

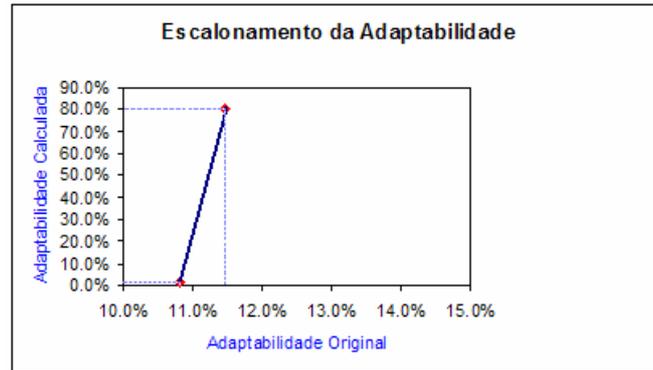


Figura 3.5: Processo de escalonamento das adaptabilidades para aumentar ou diminuir a pressão seletiva durante o processo de escolha de indivíduos.

Um processo seletivo pode ser alterado, intensificando ou suavizando a pressão seletiva sobre os indivíduos. Entende-se por pressão seletiva a chance maior ou menor de um indivíduo mediano ser selecionado para cruzamento. Quanto maior a pressão seletiva menor a chance de um indivíduo de baixa adaptabilidade ser escolhido no processo seletivo. O processo de roleta giratória apresenta uma variação natural na pressão seletiva durante a evolução do algoritmo genético. Inicialmente os indivíduos são bem diferentes e suas adaptabilidades relativas distantes, havendo maior seleção dos indivíduos com maior adaptabilidade. No entanto, a medida que as diferenças são reduzidas, o que é muito comum com o avanço do algoritmo, a adaptabilidade relativa de todos os indivíduos se tornam muito próximas, diferenciando-se por um pequeno percentual. Nesta situação, durante o processo seletivo da roleta giratória descrito, os indivíduos terão probabilidades muito próximas de serem selecionados, resultando em uma quase ausência de pressão seletiva, e conseqüente estagnação do algoritmo prematuramente. Para contornar este problema, mantendo a pressão seletiva sobre controle de forma estável ou variando conforme a necessidade, uma opção comumente utilizada é o escalonamento das adaptabilidades dos indivíduos. O escalonamento utilizado foi do tipo linear. Definiu-se os valores máximo ($MaxAdap = 0.8$) e mínimo ($MinAdap = 0.02$) desejados para a nova adaptabilidade do melhor e pior indivíduo, respectivamente. A partir destes valores e dos extremos atuais ($MaxAdapAtual$ e $MinAdapAtual$) encontra-se a equação da reta correspondente, equações (3.4) a (3.6). Aplica-se então a função linear encontrada, equação (3.6), sobre cada solução s , $AdapAtual_s$, encontrado uma nova adaptabilidade $NovaAdap_s$, que irá resultar na distribuição dos indivíduos em todo o novo intervalo,

conforme mostra o gráfico da Figura 3.5. Após a transformação existirá uma maior diferença entre as adaptabilidades dos indivíduos e haverá uma maior pressão seletiva. Essa pressão pode ser controlada, se necessário, variando os valores de *MaxAdap* e *MinAdap*. Para o algoritmo implementado estes valores foram mantidos constantes

$$A = \frac{(MaxAdap - MinAdap)}{(MaxAdapAtual - MinAdapAtual)} \quad (3.4)$$

$$B = MaxAdap - A.MaxAdapAtual \quad (3.5)$$

$$NovaAdap_s = A.AdapAtual_s + B \quad (3.6)$$

Torneio Múltiplo

A segunda técnica de seleção avaliada é conhecida como torneio múltiplo [OFC91]. Embora muito simples, tem sido largamente utilizado apresentando bons resultados. Um grupo de *M* indivíduos ($M > 1$) é selecionado aleatoriamente na população. As adaptabilidades são comparadas, vencendo o indivíduo que apresentar o melhor valor. Geralmente, a cada torneio de *M* indivíduos, um único vencedor é escolhido para recombinação. No algoritmo implementado $2P$ torneios são realizados, e conseqüentemente $2P$ indivíduos são selecionados para cruzamento, onde *P* é um percentual do tamanho da população. Após a seleção, os indivíduos escolhidos são todos recombinados dois a dois. Cada recombinação gera um descendente, que irá fazer parte da nova geração de indivíduos. O restante da população é composta de indivíduos da geração anterior copiados integralmente, após terem sido selecionados pelo mesmo método de torneio múltiplo descrito.

Elitismo

Mesmo parte dos indivíduos de uma nova geração sendo composta dos indivíduos selecionados da geração anterior, existe a possibilidade de se perder a melhor solução já encontrada. Uma opção é salvar continuamente a melhor solução encontrada fora do contexto do algoritmo, em uma variável independente, que será a solução final encontrada. Entretanto, o melhor indivíduo também corresponde a um conjunto de rotas importantes

nas recombinações e mutações seguintes. Por isso, optou-se pelo processo de cópia integral da melhor solução na geração sucessora, processo conhecido como *Elitismo*. Na realidade, duas cópias foram geradas, nas duas últimas posições do vetor de soluções que contem a nova geração. Como nesta etapa do processo global de evolução do AG_DT as mutações ainda não foram aplicadas, a melhor solução ainda poderia ser perdida por alterações realizadas nos indivíduos copiados. Para evitar que isso aconteça, duas cópias do melhor indivíduo são mantidas, uma cópia pode sofrer mutações, o que é altamente desejável, pois este representa um importante ponto do espaço de busca onde deve se intensificar as avaliações. A outra cópia é isolada e não sofre mutações, garantindo que não haverá retrocesso. A Figura 3.6 mostra o processo descrito.

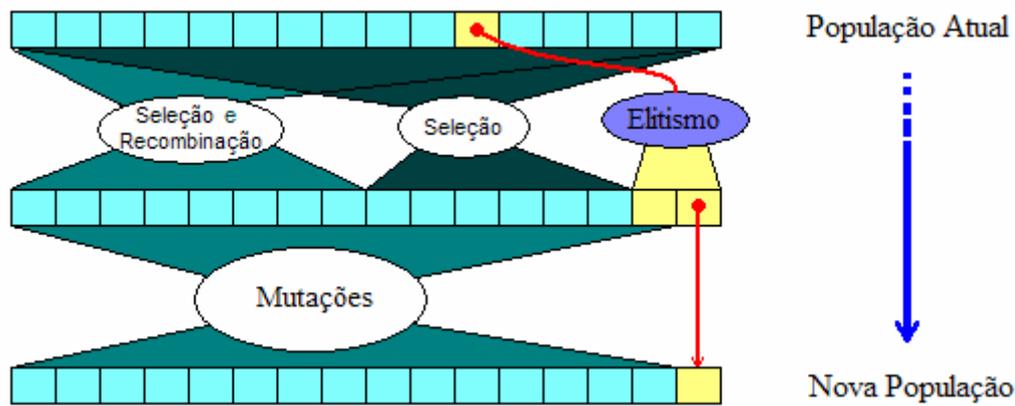


Figura 3.6: Etapas para gerar uma nova população, enfatizando o elitismo.

Recombinação

Conforme discutido previamente, um dos processos importantes do algoritmo genético é a recombinação dos indivíduos. O ponto crítico na recombinação de indivíduos soluções para o PRVJT é que normalmente é necessário algum tipo de heurística em parte do processo, uma vez que a simples troca de partes das seqüências (material genético) conduz a soluções inviáveis. Blanton em 1991 [BW91] já mostrava que a recombinação do tipo fusão ou *merge* apresentava resultados pobres para o PRVJT. Homberger [HG99] optou por não fazer qualquer tipo de recombinação na sua estratégia evolutiva ES1, que trata da redução da distância percorrida. Já na sua estratégia ES2, realiza um cruzamento baseado no código de mutação, e não no indivíduo solução em si. Ou seja, duas soluções distintas são recombinadas pela fusão do seu código de mutação, sendo a solução em si apenas

copiada integralmente de um terceiro indivíduo. Como o código de mutação representa uma seqüência de operações de remoção e inclusão de consumidores que foram aplicadas previamente aos pais, suas características são transmitidas de forma indireta.

A idéia central que motivou o procedimento de recombinação implementado neste trabalho foi o de gerar um descendente com as seguintes características:

- 1) O maior número de rotas originais possíveis dos indivíduos pais;
- 2) Equilibrar o número de rotas utilizadas de cada indivíduo doador;
- 3) Uma vez inevitável, a criação de novas rotas com os consumidores restantes deve aproximar ao máximo da seqüência original de atendimento àqueles consumidores como nos indivíduos doadores (pais).

Estes pontos são fundamentais para o máximo aproveitamento da informação de busca contida nos indivíduos pais, ou seja, para que os descendentes realmente sejam uma combinação dos pais e não o resultado de mais um operador qualquer de busca local.

O processo de recombinação tem início com a escolha de dois indivíduos soluções do conjunto gerado pelo processo de seleção utilizado. Como o processo de seleção gerou $2N$ indivíduos de forma aleatória, dois a dois são selecionados para cruzamento na própria ordem que aparecem no vetor de indivíduos selecionados $\{[1,2],[3,4],\dots, [2N-1,2N]\}$, produzindo N descendentes. Para cada par de indivíduos selecionados, Ind_1 e Ind_2 , a recombinação é feita conforme o Algoritmo 3.2 abaixo, gerando um único descendente, Ind_{des} .

1. Inicia-se dois vetores com os índices das rotas presentes em cada indivíduo: $Ind_1 = \{1,2,3,\dots,Nrotas\}$ e $Ind_2 = \{1',2',3',\dots,Nrotas'\}$;
2. Inicializa-se o indivíduo descendente vazio, sem rotas: $Ind_{des}=\{\emptyset\}$;
3. $Nenhuma_rota_possível = \text{Falso}$; // Somente para entrar no loop a 1ª vez
4. Enquanto (houver rotas em Ind_1 e Ind_2) e ($Nenhuma_rota_possível = \text{Falso}$) faça:
 - a. $Nenhuma_rota_possível = \text{Verdadeiro}$; //Inicialmente, nenhuma é possível ser inserida no descendente;

- b. Sorteia uma rota de forma aleatória de Ind_1 , em R_{sel} ;
 - c. Verifica a compatibilidade de inserir toda a rota R_{sel} em Ind_{des} . Se possível realiza a inserção, retira o índice R_{sel} de Ind_1 e atribui $Nenhuma_rota_possível = Falso$;
 - d. Sorteia uma rota de forma aleatória de Ind_2 , em R_{sel} ;
 - e. Verifica a compatibilidade de inserir toda a rota R_{sel} em Ind_{des} . Se possível realiza a inserção, retira o índice R_{sel} de Ind_2 e atribui $Nenhuma_rota_possível = Falso$;
- // Nenhuma rota inteira é mais possível no indivíduo sendo formado (Ind_{des})
5. Elimina os consumidores já atendidos em Ind_{des} de todas as rotas em Ind_1 , gerando rotas completamente vazias e algumas não vazias;
 6. Insere as rotas possíveis não vazias de Ind_1 em Ind_{des} .
 7. Enquanto houver consumidor não atendido, faça:
 - a. Insere o de menor custo, na posição de menor custo e viável das rotas já existentes de Ind_{des} ;
 - b. Se não existir possibilidade de inserção viável, cria uma nova rota, com o consumidor mais próximo do depósito central (rota de menor custo);

Algoritmo 3.2: Algoritmo de recombinação utilizado no AG_DT.

A validade do processo de cruzamento será avaliada no capítulo 6, através da alteração do valor de N , aumentando e reduzindo o percentual da população que será formada pelo processo de cruzamento. Se $N=0$, a etapa de recombinação é eliminada do AG_DT implementado, gerando a nova população diretamente a partir das operações de seleção e mutação, assim como na estratégia ES1 de Homberger [HG99]. Note que sem o processo de recombinação o algoritmo trabalha exatamente como uma busca local simples, onde a vizinhança $N(S)$ é definida pelos operadores de mutação (onde S representa todos os indivíduos da população, e não uma solução única como nos processos de busca local tradicional).

Mutações

Um conjunto de mutações foi criado, buscando tanto movimentos aleatórios no espaço de busca, imprescindível para a diversificação no algoritmo genético, como também intensificação nas proximidades das melhores soluções. Alguns operadores foram inspirados na literatura, tais como *exchange*, *swap* e λ -interchange, conforme será apresentado. Outros novos operadores também são propostos. Os operadores e sua motivação serão descritos nesta seção, enquanto que a utilização de cada um deles no resultado final do algoritmo será apresentada no capítulo 6.

Mutação de Eliminação de Rota (*M_ER*)

A mutação de eliminação de rota escolhe aleatoriamente uma rota qualquer e tenta retirar todos os seus consumidores, inserindo-os em outras rotas. Iniciando do primeiro consumidor, percorre todas as demais rotas para verificar se existe uma posição viável para o mesmo. O consumidor é inserido na primeira posição viável encontrada, independente do custo resultante. Realiza esta operação com todos os consumidores da rota selecionada, na tentativa de eliminá-la completamente. Se um consumidor não puder ser eliminado (não existe rotas que o suportem sem violação de restrições de capacidade e janela de tempo) este é mantido e inicia-se o processo para próximo consumidor. O algoritmo termina quando a rota é eliminada ou quando todos os consumidores já foram testados.

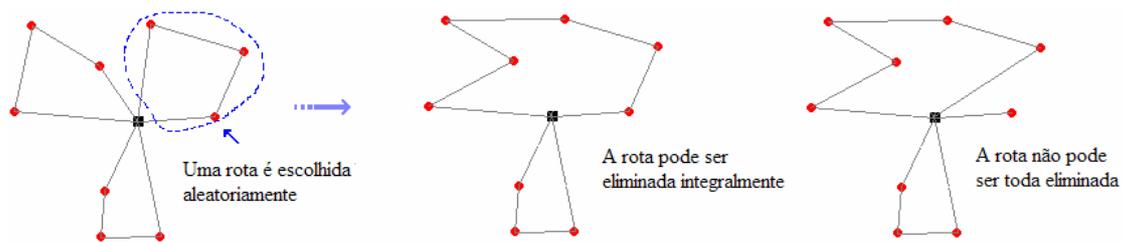


Figura 3.7: Diagrama esquemático da Mutação de Eliminação de Rota (*M_ER*)

Mutação de Retirada de Consumidor (M_{RC})

Esta mutação tem como objetivo maior aumentar a diversificação, pois é totalmente aleatória. Um consumidor é aleatoriamente retirado de um veículo na tentativa de encontrar uma posição em outro veículo para o mesmo. O veículo destino também escolhido aleatoriamente. A posição de inserção no novo veículo é definida pelo menor custo adicional obtido naquele veículo, embora a solução como um todo possa ter piorado. Se o veículo sorteado para receber o consumidor não suportá-lo, outro veículo será sorteado até esgotar todos os veículos possíveis. Somente se nenhum veículo for capaz de receber o consumidor o operador falha. A Figura 3.8 mostra esquematicamente a operação realizada por M_{RC} .

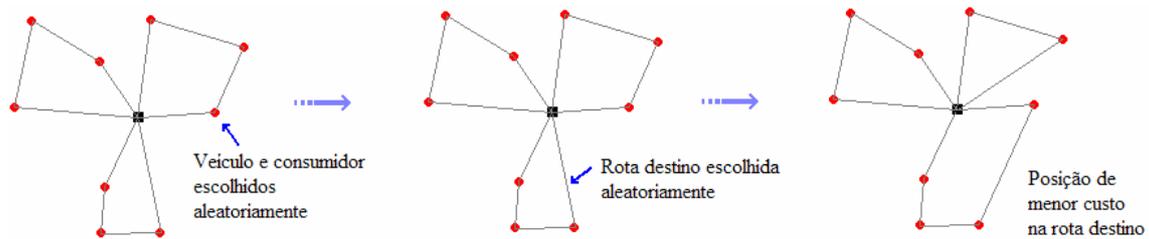


Figura 3.8: Diagrama esquemático da mutação M_{RC} .

Mutação de Inserção de Consumidor (M_{IC})

O operador de mutação de inserção de consumidor M_{IC} é semelhante ao M_{RC} , porém a rota escolhida aleatoriamente R_e irá receber um consumidor e não ceder. Definida a rota receptora aleatoriamente, todos os consumidores de outras rotas são testados um a um, em todas as posições possíveis de R_e . O consumidor que resultar na menor alteração da distância percorrida na rota R_e será retirado de sua rota origem e inserido na mesma, na posição de melhor custo. Ver na Figura 3.9 como opera M_{IC} . Não se tem conhecimento de uma utilização anterior deste tipo de operador.

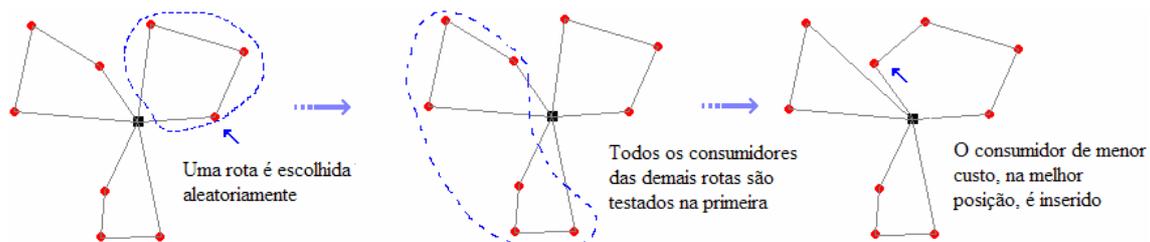


Figura 3.9: Diagrama esquemático da mutação M_{IC} .

Mutação por Troca Simples de Consumidores em Rotas Distintas (M_{TSC})

Este operador faz uma tentativa de troca aleatória de dois consumidores de duas rotas distintas. A posição do consumidor que entra será a mesma do consumidor que sai. A troca só é realizada se a solução resultante for viável.

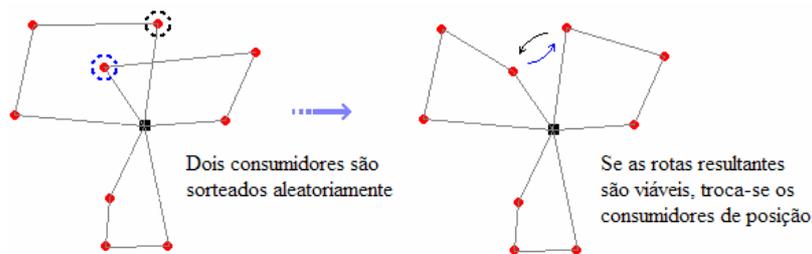


Figura 3.10 : Diagrama esquemático da mutação M_{TSC} .

Mutação por Troca de Consumidores em Rotas Distintas (M_{TC})

Este operador faz uma tentativa de troca aleatória de dois consumidores de duas rotas distintas. Ao contrário do que ocorre com a mutação M_{TSC} a posição do consumidor que entra não necessariamente será a mesma do consumidor que sai. Os consumidores irão assumir a melhor posição nas rotas destino. A troca só é realizada se houver viabilidade dos consumidores no destino em pelo menos uma ordem de atendimento.

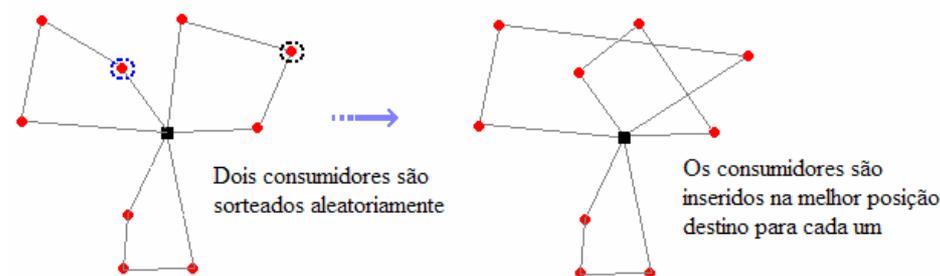


Figura 3.11: Diagrama esquemático da mutação M_{TC} .

Mutação λ -interchange (M_{INT})

O operador λ -interchange é amplamente utilizado na literatura para o PRVJT e outros problemas de roteamento. Foi introduzido por Osmar e Christofides [OC94].

Dada uma solução representada por $s = \{r_1, \dots, r_p, \dots, r_q, \dots, r_k\}$ onde r_p é um conjunto de consumidores servidos por um veículo p , uma operação λ -interchange ocorre entre um par de rotas. A operação λ -interchange entre r_p e r_q é uma troca do conjunto de consumidores $r'_p \subseteq r_p$ de dimensão $|r'_p| = \lambda$, por outro $r'_q \subseteq r_q$ de dimensão $|r'_q| = \lambda$, entre as rotas r_p e r_q . As rotas finais resultantes podem ser representadas por $r_p^* = (r_p - r'_p) \cup r'_q$ e $r_q^* = (r_q - r'_q) \cup r'_p$ e uma solução na vizinhança de s por $s^* = \{r_1, \dots, r_p^*, \dots, r_q^*, \dots, r_k\}$. O conjunto vizinhança de s , denominado $N_\lambda(s)$ é o conjunto de todas as possibilidades de s^* gerados a partir de λ -interchange.

Os valores de λ utilizados foram valores randômicos menores que 5. Dado um valor de $\lambda \leq 5$ e uma solução s , toda a vizinhança $N_\lambda(s)$ é avaliada e somente se encontrada uma solução $s^* \in N_\lambda(s)$ com redução de DT a operação é realizada.

Mutação por Inversão de Consumidores (M_INV)

Esta mutação realiza a inversão na ordem de atendimento de uma seqüência de consumidores de uma rota. Este operador também aparece com frequência na literatura do PRVJT, algumas vezes com o nome de *swap*.

Mutação por Troca Simples de Consumidores com Redução da Distância Total Percorrida (M_TSCDT)

Esta mutação é equivalente a M_{TC} , porém a troca só é realizada se houver um ganho na distância total percorrida (DT) global da solução.

Mutação por Troca de Consumidores com Redução da Distância Total Percorrida (M_TCDT)

Esta mutação retira dois consumidores aleatoriamente, C_1 de uma rota R_1 e C_2 de uma rota R_2 . Verifica-se se existe a possibilidade de R_1 receber C_2 e R_2 receber C_1 . Caso as duas possibilidades sejam viáveis e ainda houver um ganho global na distância total percorrida a troca dos consumidores é realizada. É equivalente a M_{TSMR} , porém não há ganho nos indicadores e sim redução de DT.

Mutação *PFIH* parcial (M_PFIHP)

Esta mutação também é utilizada para aumentar a diversificação, inserindo novas regiões diferentes do espaço de busca no conjunto de soluções do AG_DT. Conforme já foi dito, a heurística *PFIH estocástica* é responsável por posicionar as primeiras soluções, já com uma qualidade razoável, em determinadas posições do espaço de busca. A mutação M_PFIHP refaz o mesmo procedimento utilizado na geração inicial para um conjunto reduzido de rotas em um indivíduo. Um veículo é escolhido randomicamente. Obedecendo a ordem em que aparecem no vetor de veículos dos indivíduos (ver Figura 3.12, todos os posteriores são esvaziados). As encomendas contidas nos mesmos são então novamente roteadas utilizando o *PFIH estocástico* descrito anteriormente.

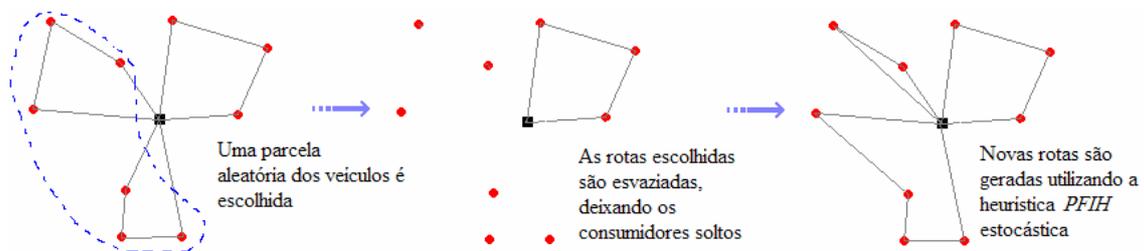


Figura 3.12: Diagrama esquemático da mutação M_PFIHP .

Mutação por Reinserção de Consumidores com ganho em DT (M_RDT)

Esta mutação retira um consumidor aleatório por vez de um veículo e testa todas as posições possíveis neste mesmo veículo. Se há alguma nova posição com menor valor de DT, sua posição é atualizada.

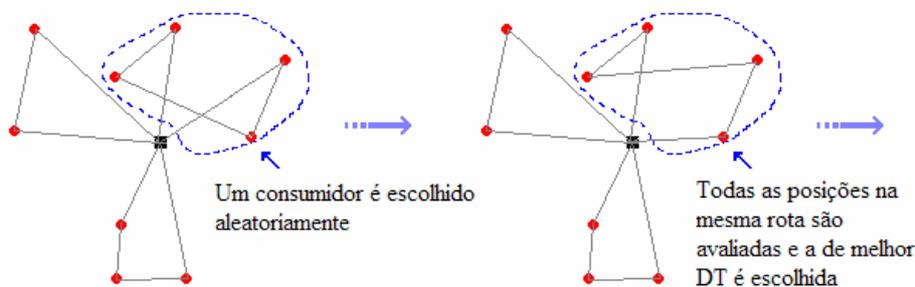


Figura 3.13: Diagrama esquemático da mutação M_RDT .

Mutação de Busca Exaustiva por Troca com Ganho de Custo (M_BETGC)

Esta mutação realiza a busca exaustiva por troca simples de consumidores entre duas rotas distintas. Duas rotas são escolhidas aleatoriamente, sendo n_{r_1} e n_{r_2} os números de consumidores das rotas r_1 e r_2 , respectivamente. Logo $n_{r_1} \times n_{r_2}$ trocas são avaliadas. A cada troca com ganho na distância total percorrida, a seqüência de avaliações é interrompida, reiniciando com rotas diferentes na próxima chamada da função.

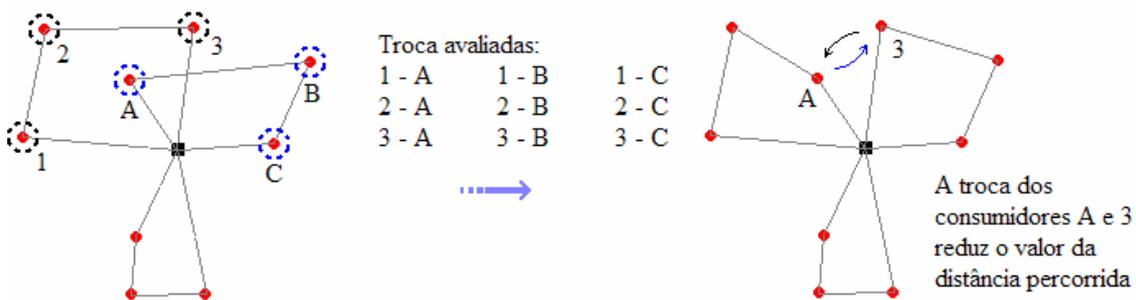


Figura 3.14: Diagrama esquemático da mutação M_TGC.

Mutação de Retirada e Troca de Consumidores seguida do PFIH (M_TCPFIH)

Esta mutação é uma combinação da busca exaustiva por troca de consumidores com ganho de DT (*M_BETGC*) e do *PFIH* parcial (*M_PFIHP*). Um veículo é escolhido aleatoriamente, uma quantidade randômica de consumidores consecutivos menor que 15 é retirada deste veículo, a partir de um consumidor também escolhido aleatoriamente. Em seguida, a solução é submetida à mutação *M_BETGC* descrita anteriormente, onde pela relaxação provocada pela retirada de consumidores possibilita trocas antes impossíveis. Posteriormente os consumidores restantes são reinseridos na solução utilizando o *PFIH*.

Unindo as partes no AG_DT

O algoritmo genético completo, parte da solução para o PRVJT é mostrado na Figura 3.15. Este algoritmo foi chamado de AG_DT, para se distinguir de sua variante que será mostrado posteriormente para atender a minimização do número de veículos. O algoritmo começa gerando uma população inicial de indivíduos soluções para o PRVJT utilizando o *PFIH* estocástico previamente descrito. O bloco de decisão a seguir, no fluxograma da Figura 3.15, apresenta a condição de parada do algoritmo, que foi definida como um tempo

limite arbitrário. Este tempo deve ser suficientemente curto para que o AG_DT possa ser executado várias vezes. O valor deste e outros parâmetros do algoritmo será discutido no Capítulo 6. A população é então avaliada e em seguida inicia-se o processo de formação da geração seguinte, esta composta de indivíduos oriundos de três estratégias diferentes, seleção direta, seleção seguida de recombinação dos indivíduos e, por fim, o elitismo, que garante a presente da melhor solução encontrada até então. Este algoritmo será utilizado diversas vezes como parte da heurística de geração de colunas proposta, conforme será mostrado.

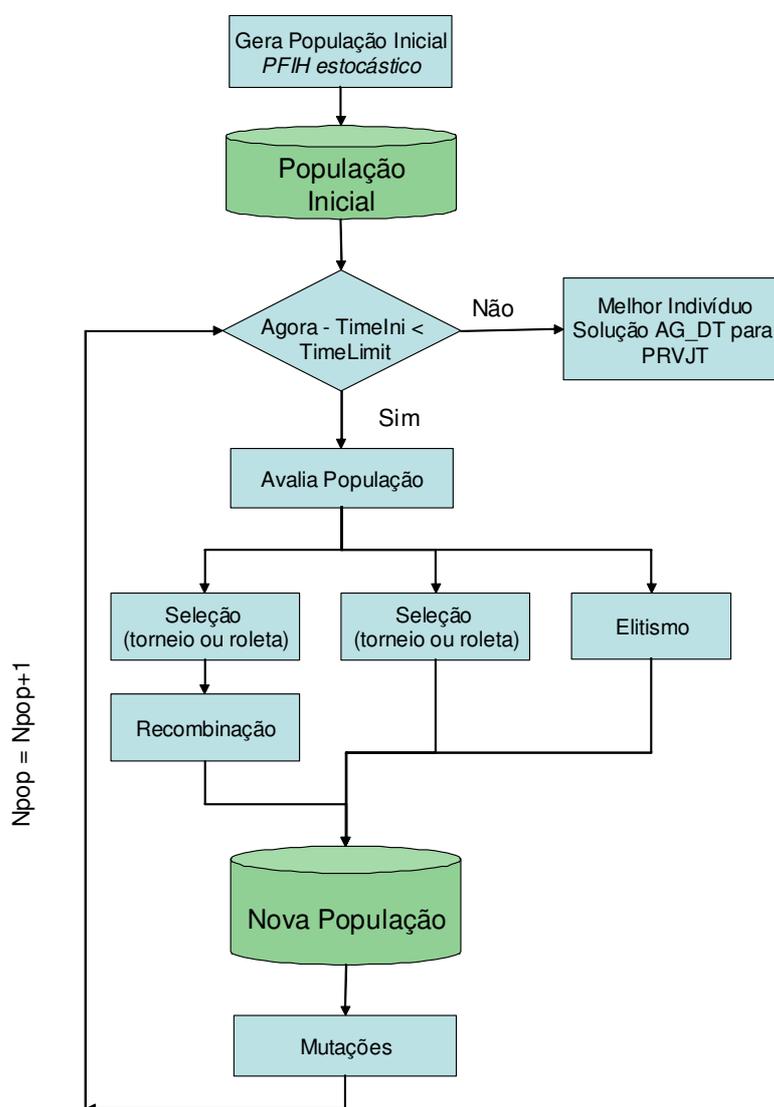


Figura 3.15 : Diagrama do Algoritmo Genético implementado (AG_DT), como parte da heurística de geração de colunas (HGC_DT) para solução do PRVJT em minimização de distância total percorrida.

3.3.3 Gerando os primeiros mínimos locais

O algoritmo genético desenvolvido é executado inicialmente várias vezes, gerando um conjunto inicial de mínimos locais e conseqüentemente rotas de qualidade para o problema principal. Um conjunto bem extenso de operadores de mutação foram propostos de forma a garantir que todo o espaço de busca possa ser atingido. Entretanto, não foi discutido como e quanto estes operadores serão aplicados, fator este decisivo para um bom desempenho do algoritmo. Estes valores e também outros parâmetros de calibração do AG_DT será discutido no Capítulo 6.

3.3.4 Melhorando os mínimos locais através do problema reduzido

Alguns trabalhos têm sido propostos utilizando a divisão do conjunto de consumidores do PRVJT em sub-regiões, podendo então resolver cada sub-região como um problema independente de menor ordem. Para o problema de roteamento de veículos sem janela de tempo está estratégia pode auxiliar, embora sempre será necessária uma etapa de melhoria das rotas, onde todo o conjunto de consumidores é novamente considerado, podendo haver troca de consumidores entre as rotas. No entanto, para o PRVJT, que possui janelas de tempo e restrições de capacidade, regiões físicas não são geralmente obedecidas, uma vez que um consumidor geograficamente próximo pode ter restrição que não permite o seu atendimento em uma ordem e rota natural. Isso resulta em rotas que se cruzam, resultando em perda na definição de regiões atendidas.

Entretanto, observa-se que a idéia básica de redução do número de consumidores é muito interessante, visto a natureza combinatória do problema. Essa estratégia também é bem adequada à heurística de geração de colunas, pois a redução do número de consumidores permite que os operadores trabalhem mais intensificadamente com um grupo de consumidores que são suficientes para compor possivelmente uma rota ótima procurada. Por outro lado, se a divisão for prejudicial, a rota não será aproveitada no problema principal sem maior prejuízo na solução final, já que não só o problema reduzido é usado na geração de rotas. Definiu-se então uma etapa do processo onde se elimina uma parcela dos consumidores, gerando um *problema reduzido*. Entretanto, para evitar o problema das regiões que não são bem definidas para o PRVJT, decidiu-se por não eliminar os

consumidores baseando-se proximidade geográfica, mas sim aqueles pertencentes a um subconjunto de rotas de uma solução parcial (mínimo local). Dado um mínimo local de uma etapa intermediária da heurística proposta, para cada rota um processo de sorteio foi realizado. Dada uma probabilidade de ROUTE_PERCENT, valor que foi ajustado na fase de calibração do algoritmo, uma rota pode ter seus consumidores incluídos no problema reduzido.

Definido então um conjunto reduzido de consumidores, o algoritmo AG_DT foi utilizado exatamente como descrito na seção 3.3.2, mantendo também os mesmos parâmetros reduzindo apenas o tempo disponível para cada execução. Como um pequeno conjunto de rotas é selecionado para compor o problema reduzido é necessário repetir o procedimento várias vezes, dando chance para que todos os consumidores sejam tratados. Por isso, o algoritmo é executado novamente diversas vezes, para cada execução um novo conjunto de consumidores é escolhido, repetindo o processo de sorteio de rotas.

Nenhuma rota foi aproveitada para compor um indivíduo da população inicial na solução do problema reduzido. A heurística *PFIH estocástica* foi utilizada novamente. As rotas são apenas usadas para definir um subconjunto de consumidores que têm mais chance de serem atendidos de forma conjunta na solução ótima desconhecida. A Figura 3.16 apresenta o diagrama do processo desenvolvido. No primeiro quadrado, está a solução gerada pelo AG_DT tratando o problema completo, com 100 consumidores. O segundo quadrado corresponde a um conjunto de rotas aleatoriamente selecionadas para compor o problema reduzido. O terceiro quadrado corresponde somente ao conjunto de consumidores utilizados no problema, sem qualquer informação das rotas anteriores. Se este conjunto corresponder integralmente a um conjunto de rotas da solução ótima desconhecida um problema reduzido terá sido produzido, independente das rotas originais. Ou seja, neste caso hipotético a solução mais simples do problema com número de consumidores menor será parte da solução ótima global.

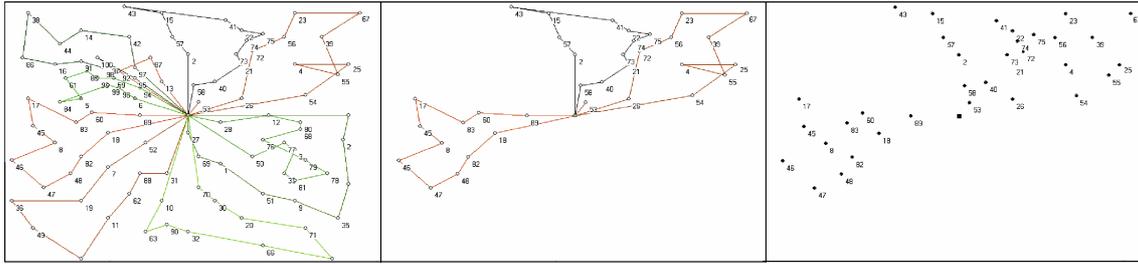


Figura 3.16: Os três quadros representam respectivamente, a solução do AG_DT para o problema completo, as rotas aleatoriamente selecionadas e o problema reduzido gerado.

Adicionalmente o fato de a divisão ser feita baseada em rotas de um mínimo local, haverá uma tendência do AG_DT encontrar soluções na mesma direção. Isso pode permitir pequenas correções, como consumidores em ordem inadequada ou em rotas trocadas, se em pequeno número de ocorrências, pois seriam mais facilmente identificadas com poucas rotas e consumidores no problema.

3.3.5 Encontrando soluções próximas ao ótimo global

Até aqui, foi mostrado o que foi utilizado para produzir uma solução inicial rápida, utilizando o algoritmo genético e seus operadores. Foi apresentado também como aperfeiçoar a solução escolhendo um conjunto de consumidores e resolvendo um problema de ordem reduzida (20% a 40% do original), utilizando o mesmo algoritmo AG_DT. Faz-se necessário lembrar que embora as qualidades das rotas produzidas estejam ligadas à qualidade da solução em si, produzida pelo AG_DT, a estratégia pretendida requer um número significativo de rotas de boa qualidade para levar ao modelo de particionamento de conjuntos, resultando em uma maior robustez para proposta global de solução do PRVJT. Utilizando-se poucas rotas a garantia de qualidade da solução não seria possível, pois não existe nenhuma garantia que os operadores desenvolvidos impedirão a estagnação do AG_DT em mínimos locais.

Um fluxo global de geração de rotas para o problema principal utilizando o AG_DT foi proposto, caracterizado por fases alternadas de diversificação e intensificação. A primeira fase, de diversificação, caracteriza-se pela geração de um número de mínimos locais utilizando o problema completo, através do AG_DT. Este processo está representado no diagrama da Figura 3.17. Inicia-se o algoritmo com os conjuntos R e R_{GLOBAL} vazios. O AG_DT é então executado por um tempo pré-determinado, utilizando o problema

completo, gerando uma primeira solução para o PRVJT, conforme bloco A do fluxograma da Figura 3.17. Todas as rotas que compõe a solução gerada pelo AG_DT serão incluídas no conjunto R , bloco B do fluxograma. Um número de Max_Evol execuções (bloco C), serão realizadas de forma completamente independente, gerando diversos mínimos locais diferentes, todos estes gerando rotas para o conjunto R . Após Max_Evol mínimos locais, o conjunto R conterà um número razoável de rotas, e um PPC será resolvido utilizando este conjunto, podendo já obter uma solução melhor que todas as anteriores, bloco D do fluxograma.

O resultado do PPC será um mínimo local com qualidade igual ou superior à melhor solução encontrada na fase I do processo. Esta solução é o ponto de partida para a fase II, fase esta de intensificação. As rotas da solução encontrada pelo PCC serão utilizadas como referência para dividir os consumidores, gerando diversos problemas reduzidos conforme foi descrito na seção 3.3.4. O mesmo AG_DT é utilizado para resolver o problema reduzido, alterando somente o tempo disponível para o algoritmo. Um número fixo e pré-determinado (Max_Evol2) de execuções é realizada, blocos E, F e G do fluxograma da Figura 3.17. Novamente todas as rotas das soluções são incluídas no mesmo conjunto R , que até o momento continua acumulando todas as rotas das soluções geradas pelo AG_DT. Conforme já foi dito, considera-se a fase II como intensificação, por dois motivos, a saber:

- Qualquer solução gerada pelo AG_DT tratando o problema reduzido, que seja melhor em distância percorrida que as rotas originais que deram origem ao conjunto de consumidores do problema, irão melhorar a solução atual através de uma composição natural com as demais rotas do PPC;
- As rotas geradas nas soluções do AG_DT para o problema reduzido, irão apresentar uma “tendência” na direção do mínimo local do PPC, pois a divisão dos consumidores conforme as rotas daquele mínimo local induzem a busca em determinada direção.

Termina assim o ciclo principal da heurística proposta, HGC_DT. Para iniciar mais um ciclo, idêntico ao descrito anteriormente, o conjunto R é esvaziado e todas as rotas são acumuladas em um grande conjunto R_{GLOBAL} , que receberá rotas até que o tempo total disponível para HGC_DT termine. Iniciando um novo ciclo pela fase I, novos mínimos

locais totalmente independentes são gerados pelo AG_DT. O conjunto R estará vazio, resultando em uma intensificação através do PPC e da solução dos problemas reduzidos em outra região do espaço de busca.

É importante observar que embora o objetivo principal da heurística HGC_DT seja gerar rotas de qualidade no grande conjunto R_{GLOBAL} (Bloco J e K do fluxograma), há uma estratégia adicional de aproveitar ao máximo o tempo disponível global balanceando o mais perfeitamente possível etapas de intensificação e diversificação, conforme descrito.

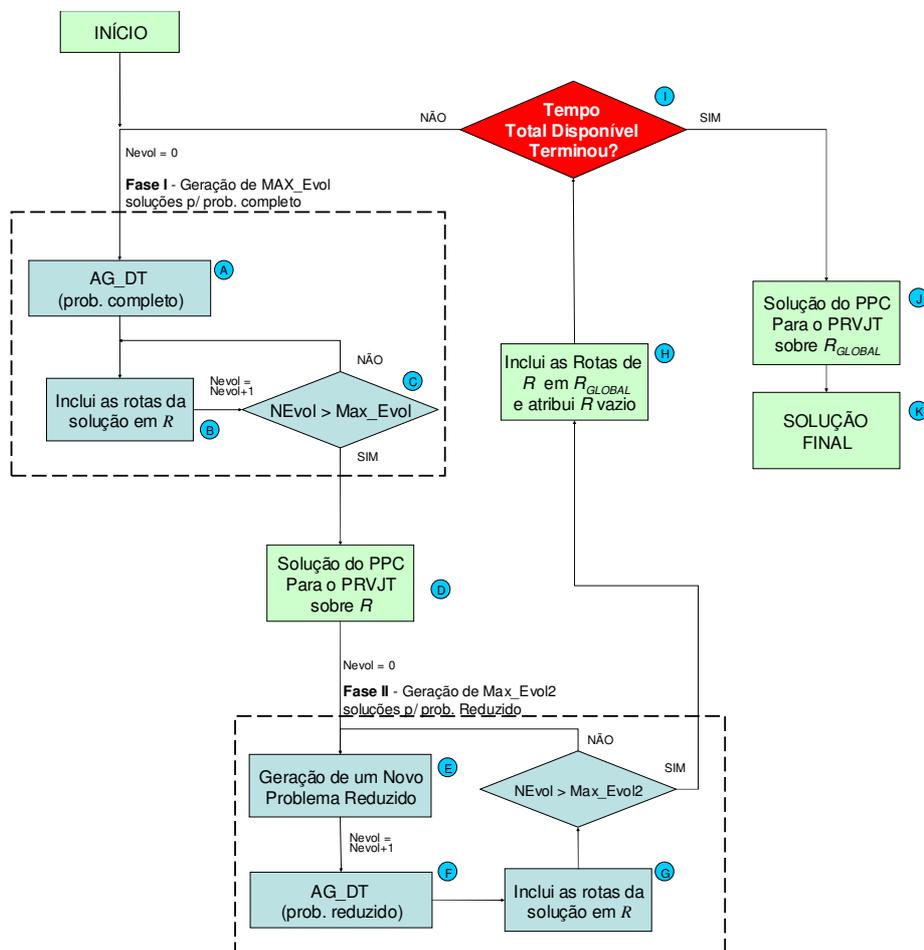


Figura 3.17: Diagrama completo da solução utilizada para resolver o PRVJT (HGC_DT).

Todos os parâmetros não mencionados neste capítulo, como número de indivíduos, taxa de mutação, tempo de execução do AG_DT para o problema completo e reduzido, entre outros, podem ser facilmente alterados para permitir encontrar as melhores configurações

através de testes. Os detalhes destes ajustes e os melhores parâmetros encontrados serão descritos no capítulo 6.

3.4 Conclusões

Este capítulo apresentou uma alternativa de solução para o PRVJT estático utilizando uma heurística baseada em geração de colunas e no algoritmo genético. O objetivo único foi a minimização da distância total percorrida pelos veículos. Para gerar colunas de qualidade, a serem utilizadas no problema principal de programação linear inteira, um algoritmo genético AG_DT foi proposto. Um processo cíclico composto de fases de diversificação e intensificação foi desenvolvido para evitar certos mínimos locais. A principal característica da fase de intensificação é a geração de problemas reduzidos, que é diversas vezes utilizado pelo mesmo AG_DT. Este processo é repetido até que o tempo global disponível se esgote, gerando sempre rotas para que um modelo de particionamento de conjuntos possa ser resolvido sobre um amplo conjunto com colunas de qualidade, geradas nas etapas anteriores. A estratégia é muito promissora, objetivando por um lado suprir a deficiência em robustez das heurísticas para o PRVJT, buscando soluções próximas ao ótimo global. Por outro lado, parece ser adequada ao tratamento do problema dinâmico, por possuir etapas independentes, mas que colaboram entre si na geração do conjunto de rotas do conjunto R_{GLOBAL} . A sua utilização para o problema dinâmico será descrita no capítulo 5, e os resultados para o PRVJT estático e dinâmico nos capítulos 6 e 7 respectivamente.

Capítulo 4

Minimizando o Número de Veículos no PRVJT Estático

Uma vez que os resultados de HGC_DT foram bem expressivos para minimização da distância total percorrida (DT), decidiu-se avaliar também seu desempenho para um outro objetivo também muito comum na literatura, o número de veículos (NV) necessário para o atendimento de todos os consumidores à frente de DT. Com este objetivo, este capítulo apresenta uma extensão da heurística proposta no capítulo anterior, também para o PRVJT estático. Como segundo objetivo, para um número de veículos já minimizado, permanece a distância total percorrida. A seção 4.1 apresenta contextualização do trabalho na literatura e as principais motivações para a proposta atual. A seção 4.2 apresenta as principais dificuldades na minimização do número de veículos, as diferenças encontradas comparando com a minimização de DT e as modificações realizadas, incluindo um novo algoritmo genético, AG_NV. A seção 4.3 apresenta a nova heurística como um todo e seu macrofluxo. Por fim, na seção 4.4 são apresentadas as conclusões.

4.1 Introdução

O número de veículos tem sido o principal objetivo adotado pela maior parte das heurísticas propostas para solução do PRVJT na literatura. Entretanto, o número NV minimizado alcançado por várias propostas de solução tem sido o mesmo em grande parte das instâncias utilizadas nos testes. Conseqüentemente, a minimização da distância total percorrida tratada no capítulo anterior, o segundo objetivo geralmente adotado, acaba por ser o critério de desempate. Como os resultados obtidos utilizando a heurística HGC_DT proposta para minimização da distância percorrida foram muito positivos, decidiu-se também avaliar o seu desempenho após a minimização do número de veículos. Isto é, avaliar o desempenho para minimização de distância com um número de veículos já minimizado. Adicionalmente, é objetivo deste trabalho desenvolver um novo algoritmo

genético capaz de minimizar o número de veículos (NV) para o PRVJT, uma vez que este pode ser o principal objetivo em alguns casos reais.

4.2 Reduzindo o número de veículos

A primeira tentativa para produzir um algoritmo genético capaz de produzir resultados positivos na minimização de NV foi a simples alteração nas funções objetivos do algoritmo genético anterior, descrito no capítulo 4 (AG_DT), e também no modelo do problema de particionamento de conjunto. Neste último uma constante K suficientemente grande foi somada ao custo de cada rota. Somando-se um valor bem maior que o custo esperado máximo c_r (que representa a distância total percorrida na rota r), se torna impossível qualquer redução do custo nas rotas já existentes serem viabilizadas com a adição de mais um veículo (rota).

O modelo do PPC desejado para minimização do número de veículos, seguido da distância total percorrida, é então obtido pela substituição da equação (3.1) pela equação (4.1) abaixo:

$$\text{Minimize } \sum_{r \in R} (K + c_r) x_r \quad (4.1)$$

Já a função adaptabilidade utilizada no AG_DT, foi alterada conforme equação (4.2). Como não existe rota com custo inferior a unidade que representa o atendimento ao consumidor mais próximo isoladamente (nas instâncias utilizadas), o primeiro termo da equação (4.2) estará sempre dentro intervalo $[0,1]$. Considerando K_1 uma constante positiva, bem maior que K_2 (Ex: $K_1 > 100 K_2$) também positiva maior que 1, qualquer redução de um veículo representará um aumento na adaptabilidade, independente de um possível aumento causado na distância total percorrida.

$$\text{Adaptabilidade}_s = K_1 - \frac{1}{\sum_{r \in R^s} c_r} - K_2 \cdot NV \quad (4.2)$$

Na equação 4.2, R^s representa as rotas de um indivíduo solução s . Entretanto a simples alteração da função de adaptabilidade, em direção à nova função objetivo, mantendo o

algoritmo AG_DT apresentado não gerou bons resultados. Observou-se que os objetivos, números de veículos e distância total percorrida são muito concorrentes. Como rapidamente, os indivíduos adquirem praticamente o mesmo número de veículos, ou seja, um a dois terços da população apresentará o mesmo número de veículos do melhor indivíduo, a pressão seletiva acaba por ser exercida no custo das rotas (DT). Como resultado, os indivíduos são selecionados pelo segundo critério e há uma tendência à redução de DT. Com distâncias cada vez menores, como são objetivos concorrentes, fica cada vez mais difícil diminuir o número de veículos, estabelecendo um mínimo local que segura o avanço dos indivíduos soluções. Berger [BBB01] enfatiza a necessidade de realizar o processo de busca em duas fases distintas, primeiro reduzindo o número de veículos e só então iniciar a busca por redução da distância percorrida, para um número fixo de veículos. Por isso, o AG_DT com a função adaptabilidade da equação (4.2) é utilizado somente depois do número de veículos já ter sido minimizado, conforme será descrito posteriormente.

Entretanto, a diferença do número de veículos não se apresenta adequada como critério de seleção dos indivíduos, pois um terço dos mesmos seriam avaliados com o mesmo resultado. Como a distância percorrida nem sempre atrai para direções no espaço de busca que significam redução de veículos, um outro critério de seleção deve ser utilizado. Um dos caminhos é encontrar características nos indivíduos soluções que, diferenciando-os uns dos outros, valoriza aqueles que têm maior possibilidade de, após sucessivas operações, reduzirem o número de veículos.

Algumas propostas são encontradas na literatura, como o tempo de retardo proposto por Homberger [HG99]. A soma dos quadrados do número de consumidores nas rotas e o número de consumidores na menor rota têm sido outros fatores que indicam a facilidade ou não de reduzir novas rotas nas diferentes soluções.

4.2.2 Atribuição de adaptabilidades hierárquicas aos indivíduos

Existem basicamente duas linhas distintas de realizar a busca por soluções para o PRVJT em termos de número de veículos. A primeira linha caminha pela região inviável do espaço de soluções, tentando minimizar a violação de tempo. Esta linha é bem representada pelo

trabalho de Berger [BBB01]. A outra linha caminha pela região viável do espaço de soluções. Esta opção requer a identificação das soluções com possibilidade de redução de veículos. Para isso, Homberger [HG99] propõe uma avaliação lexicográfica das soluções. O primeiro critério é o número de veículos já alcançado, o segundo é o número de consumidores da menor rota, e o terceiro o tempo mínimo de retardo ou “*minimal delay*”, conforme foi apresentado no capítulo 2. Nota-se que o tempo de retardo, que indica a violação prevista com a eliminação de um consumidor da menor rota, se assemelha um pouco com a idéia da busca pela região inviável, pois também indica o quanto há de violação das janelas de tempo nas soluções, mas neste caso, para uma hipotética eliminação dos consumidores da menor rota.

Um dos principais operadores de vizinhança, utilizado na busca de soluções com número reduzido de veículos, faz uma tentativa de eliminar todos os consumidores da menor rota. É natural então, avaliar os indivíduos tentando quantificar quanto difícil é eliminar consumidores da menor rota, como o fez Homberger [HG99].

A estratégia de minimização do número de veículos proposta neste capítulo busca expandir a idéia utilizada em [HG99] de quantificar a dificuldade de eliminar consumidores da menor rota, propondo novos indicadores de seleção dos indivíduos da população de forma hierárquica, ou lexicográfica conforme chamou Homberger. Para tal, foram utilizados nada menos que nove indicadores, conforme serão descritos a seguir. Um novo algoritmo genético foi desenvolvido, utilizando esta nova estratégia de seleção e novos operadores (mutações). Este novo algoritmo genético foi chamado de AG_NV, diferenciando-o do seu antecessor AG_DT. O algoritmo AG_DT original, cujos resultados motivaram a tentativa de redução do número de veículos será utilizado em uma segunda fase, unicamente para redução da distância total percorrida depois de NV ter sido minimizado.

A importância de cada um dos novos indicadores propostos para avaliar os indivíduos foi medida empiricamente e os resultados serão apresentados no capítulo 7. Estes indicadores foram chamados de indicadores de adaptabilidade. Durante o processo de avaliação dos indivíduos soluções, cada indicador de adaptabilidade será calculado de forma independente para cada indivíduo. Segue, a descrição dos indicadores propostos.

Número total de veículos (Adap_NV)

O indicador Adap_NV é definido diretamente como o número de veículos utilizados por um indivíduo solução para o PRVJT. Como NV é o primeiro objetivo de AG_NV, este deve existir como o principal critério de avaliação na seleção dos indivíduos.

Número de consumidores na menor rota (Adap_NCMR)

O indicador Adap_NCMR será atribuído como o número de consumidores da menor rota presente no indivíduo solução. Isto é, uma rota com um pequeno número de consumidores é um indicador importante utilizado para medir a dificuldade em se eliminar uma nova rota em uma solução. É razoável considerar que as rotas com menor número de consumidores terão maior probabilidade de serem mais fáceis de serem eliminadas, uma vez que as demais rotas deverão absorver um número menor de consumidores.

Dificuldade em eliminar um consumidor da menor rota (Adap_DEUCMR)

Este indicador é baseado no tempo de retardo ou *minimal delay* proposto por Homberger [HG99]. Quando um consumidor é retirado da menor rota e inserido em outra, três situações podem ocorrer:

- Há violação da capacidade de carga. Neste caso $\text{Adap_DEUCMR} = \infty$;
- Não há violação de capacidade, mas há violação em uma ou várias janelas de tempo. Neste caso, a quantidade de tempo total violada, em todos os consumidores da rota, é calculado e atribuído ao Adap_DEUCMR;
- O consumidor pode ser inserido na rota, mantendo a solução viável. Neste caso $\text{Adap_DEUCMR} = 0$.

A diferença de Adap_DEUCMR e do tempo de retardo de Homberger é que não é adicionado o tempo de retardo no consumidor seguinte, se este é atendido dentro da janela de tempo, ao contrário do que faz Homberger [HG99].

Cada consumidor da menor rota causará uma violação diferente. Que por sua vez será também diferente em cada posição a ser inserido na rota. Todos os consumidores da menor

rota são testados em todas as rotas e em todas as posições. Adap_DEUCMR receberá a menor violação de tempo encontrada. A rota onde este menor valor de violação de tempo foi calculado recebe o nome de *rota receptora*, pois será a rota com maior chance de receber o consumidor da menor rota. O conceito de *rota receptora* será importante na definição de Adap_DEUCRR.

Dificuldade em eliminar todos os consumidores da menor rota (Adap_DETCMR)

O cálculo da violação de tempo para a eliminação de todos os consumidores da menor rota é muito semelhante ao descrito para o indicador Adap_DEUCMR. Quando a menor rota tem apenas um consumidor eles são coincidentes. Para dois ou mais consumidores, Adap_DEUCMR receberá a soma das mínimas violações de tempo calculada sobre cada consumidor presente na menor rota. Este indicador é bastante significativo quando existem vários consumidores na menor rota, já que não há garantia que o consumidor aparentemente mais fácil será eliminado primeiro. Isso ocorre porque aquele consumidor que resultou no menor Adap_DEUCMR pode ser de difícil eliminação. Se existem vários consumidores causando pequena violação de tempo ao deixarem a menor rota, é maior a chance de um deles encontrar uma abertura de tempo para ser inserido em outras rotas, após algumas operações (mutações ou cruzamentos) com algum indivíduo solução.

Dificuldade de eliminar um consumidor da rota receptora (Adap_DEUCRR)

A *rota receptora*, conforme definido previamente, é a rota onde a menor violação de tempo foi obtida com a inclusão de um consumidor vindo da menor rota. Muitas vezes parece ser fácil eliminar um consumidor da menor rota, se os valores de Adap_DETCMR ou Adap_DEUCMR são pequenos. Porém embora a rota receptora tenha indicado uma “maior facilidade” para receber tal consumidor, pode ainda haver uma grande dificuldade de se eliminar a pequena violação de tempo atual. A forma mais provável de aumentar a janela de tempo na rota receptora e permitir a inserção do consumidor é retirando outros consumidores dela. Este indicador calcula o *tempo de retardo* (equivalente a Adap_DEUCMR), mas desta vez considerando a rota receptora e não a menor rota. Obviamente não se calcula a violação de tempo levando um consumidor para a menor rota, pois seria um retrocesso, mas sim para outra qualquer que viabilizaria a retirada de um

consumidor da rota receptora. Desta forma $Adap_DEUCRR$ é calculado sobre a rota receptora assim como $Adap_DEUCMR$ é calculado sobre a menor rota.

Dificuldade de eliminar todos os consumidores da rota receptora ($Adap_DETCRR$)

Ainda na tentativa de medir a dificuldade de abertura ou ampliação da janela de tempo disponível na rota receptora, a mínima violação de tempo causada para eliminar todos os consumidores da rota receptora é calculada em $Adap_DETCRR$. A forma de cálculo é a mesma de $Adap_DETCMR$, porém utilizando a rota receptora no lugar da menor rota.

Distância total percorrida ($Adap_DT$)

Se por um lado, a distância total percorrida utilizada como um dos indicadores de adaptabilidade dos indivíduos pode concorrer diretamente com a redução do número de veículos, por outro lado ela pode ajudar a criar maiores condições para a diminuição dos consumidores da menor rota. Por exemplo, se a rota receptora atende os mesmos consumidores com uma menor distância percorrida, o veículo irá ficar mais tempo disponível e, conseqüentemente, terá maior capacidade para receber outros consumidores. Por isso é interessante utilizar DT também como indicador de adaptabilidade na seleção dos indivíduos.

Número de consumidores na rota receptora ($Adap_NCRC$)

Este indicador é muito simples e intuitivo, porém nem sempre representa uma real tendência do indivíduo solução de perder uma rota. Considera que quanto menor for o número de consumidores na rota receptora, maior será a facilidade para receber outros consumidores da menor rota, que deverá ser eliminada.

Soma dos quadrados dos números de consumidores das rotas ($Adap_SQNC$)

A soma dos quadrados dos números de consumidores das rotas de uma solução, conforme equação 4.3, e tão maior quanto mais concentrado estiverem os consumidores em poucas rotas. Este indicador é interessante, pois diferentemente dos demais, é calculado sobre o

conjunto total de rotas. Este indicador foi utilizado por [] como principal indicador de adaptabilidade.

$$Adap_SQNC_s = \sum_{r \in R^s} N_r^2 \quad (4.3)$$

Onde

R^s = Conjunto de rotas de um indivíduo solução sendo avaliado;

r = uma rota qualquer pertencente a R^s ;

N_r = Número de consumidores da rota r .

Carga total da Rota Receptora (Adap_CTRR)

Uma rota com menor quantidade de carga na soma das demandas dos consumidores é uma rota com menor restrição para receber outros consumidores. Este indicador é útil para problemas onde a capacidade de carga é mais crítica que as restrições de tempo de entrega.

4.2.3 Seleção por Torneio Múltiplo

A seleção dos indivíduos no AG_NV objetiva estabelecer uma escolha baseada nos indicadores de adaptabilidade propostos de forma hierárquica. A seleção por torneio é ideal para este fim, pois permite a comparação sucessiva dos indicadores até que um deles apresente uma diferença, produzindo um indivíduo vencedor do torneio.

Inicialmente, define-se a ordem de prioridade dos indicadores de adaptabilidade propostos. Os primeiros serão sempre o número de veículos (Adap_NV) e o número de consumidores da menor rota (Adap_NCMR), nesta ordem, pois indicam muito diretamente a função objetivo do problema e o caminho de eliminação de rotas proposto, respectivamente. Quanto aos demais, foram avaliadas várias ordens de prioridade, mostradas no capítulo 6.

Define-se, primeiramente, uma hierarquia desejada dos indicadores de adaptabilidade, por exemplo:

- 1° - Adap_NV
- 2° - Adap_NCMR
-,
- 9° - Adap_CTRR

Em seguida, o procedimento de seleção por torneio pode ser descrito da seguinte forma:

1. Escolha aleatória de M indivíduos: $Ind_v = \{I_1, I_2, \dots, I_M\}$;
2. Retorna em $Ind_v = \text{Min}(\text{Adap_NV}, Ind_v)$;
3. Se $|Ind_v| \neq 1$ então Retorna em $Ind_v = \text{Min}(\text{Adap_NCMR}, Ind_v)$;
4. senão, se $|Ind_v| \neq 1$ então Retorna em $Ind_v = \text{Min}(\dots, Ind_v)$;
5. senão, se $|Ind_v| \neq 1$ então Retorna em $Ind_v = \text{Min}(\text{Adap_CTRR}, Ind_v)$;
6. Por fim, o individuo restante na primeira posição é selecionado:
Retorna($Ind_v[1]$);

Utilizando-se nove indicadores de adaptabilidade é improvável que fiquem dois diferentes indivíduos no conjunto Ind_v após a última seleção, se houver, o primeiro é selecionado. Entretanto, viu-se que é muito comum indivíduos com mesmo número de veículos, mesmo número de consumidores na menor rota e ainda mais alguns indicadores de adaptabilidade iguais. Por isso a importância de todos os indicadores propostos, para haver uma diferenciação e uma melhor seleção.

4.2.4 Operadores para minimização de veículos

Muitos dos operadores utilizados no AG_DT, aqueles responsáveis pela intensificação, direcionam a busca para a redução da distância percorrida. Por isso, novos operadores foram propostos. Os critérios utilizados para escolha e desenvolvimento de novos operadores foram os seguintes:

- Dois tipos básicos de operadores devem estar presentes. O primeiro tipo de operadores são aqueles que aumentam a diversificação durante a busca. Normalmente são baseados em trocas ou migração aleatória de consumidores. O segundo tipo intensifica a busca, utilizando heurísticas básicas para melhorar a qualidade das soluções atuais;
- Os operadores que intensificam a busca devem estar orientados na mesma direção dos indicadores de adaptabilidade utilizados para selecionar os indivíduos. Ou seja, os operadores de intensificação não devem estar focados em somente diminuir o número de veículos dos indivíduos solução, mas também melhorar os indicadores de adaptabilidade, tais como: eliminar consumidores da menor rota, aumentar a capacidade de receber consumidores na rota receptora, entre outros.

A seguir são descritos os operadores de mutação utilizados no AG_NV. A forma e quantidade utilizada será descrita no Capítulo 7.

Mutação por troca simples de consumidores em rotas distintas (M_{TSC})

Este operador é um operador de diversificação genérico, o mesmo utilizado para o AG_DT. Faz uma tentativa de troca aleatória de dois consumidores de duas rotas distintas. A posição do consumidor que entra será a mesma do consumidor que sai. A troca só é realizada se a solução resultante for viável.

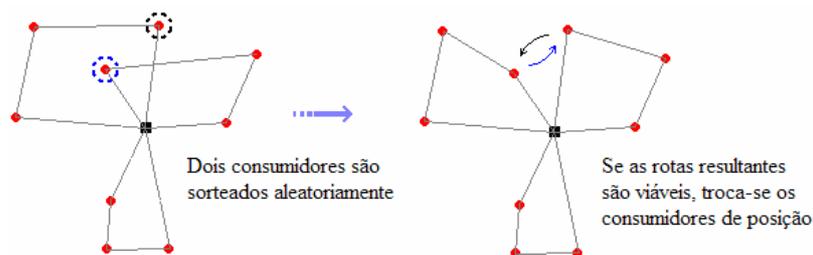


Figura 4.1: Diagrama esquemático da mutação M_{TSC} .

Mutação por troca simples de consumidores na mesma rota (M_{TSCMR})

Este operador é muito semelhante ao anterior, mas não há troca de rota, mas sim na ordem de atendimento de dois consumidores, dentro de uma mesma rota.

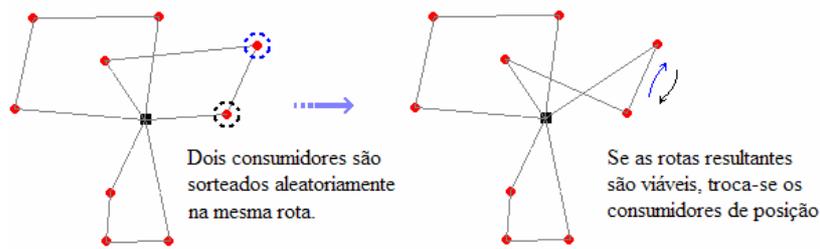


Figura 4.2: Diagrama esquemático da mutação M_{TSCMR} .

Mutação por troca de consumidores com redução de Adap_DETCMR, Adap_DEUCMR e Adap_DEUCRR ($M_{TCRA\text{adap}1}$)

Duas rotas são escolhidas aleatoriamente. Todos os pares de consumidores, um de cada rota, são candidatos à troca. A ordem de atendimento do consumidor que entra não necessariamente é a mesma do consumidor que sai, e vice-versa. A troca é feita se há ganho em termos de facilidade de eliminar a última rota, ou seja, ganho em pelo menos um dos indicadores Adap_DETCMR, Adap_DEUCMR ou Adap_DEUCRR.

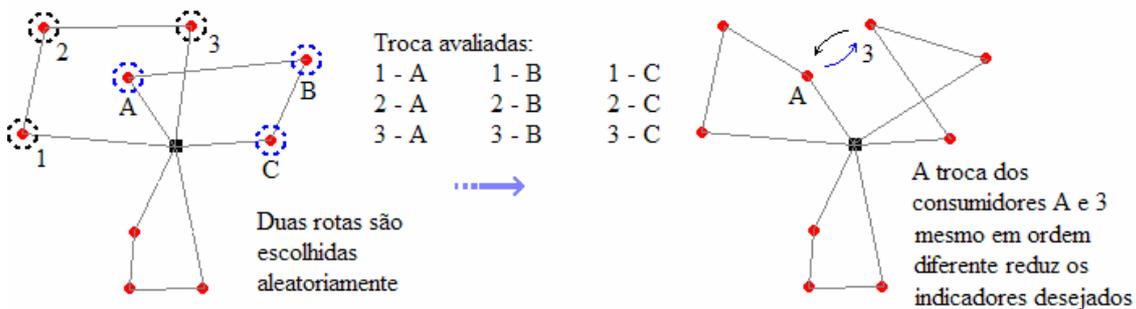


Figura 4.3: Diagrama esquemático da mutação $M_{TCRA\text{adap}1}$.

Mutação por troca simples de consumidores com redução de Adap_DETCMR, Adap_DEUCMR ($M_{TCRA\text{adap}2}$)

Duas rotas são escolhidas aleatoriamente. Todos os pares possíveis de consumidores são candidatos à troca. Diferentemente da mutação $M_{TCRA\text{adap}1}$, a troca é feita se o consumidor que entra puder preencher a mesma posição do que sai e vice-versa. A redução de dois indicadores da facilidade de eliminação da menor rota são condições para realizar a troca, Adap_DETCMR e Adap_DEUCMR.

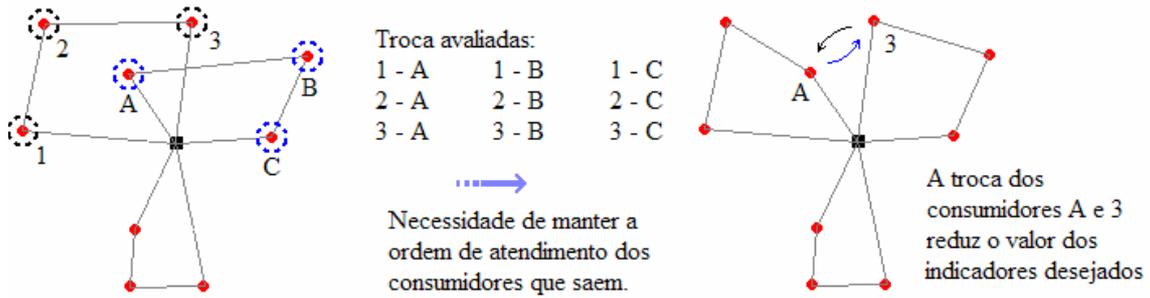


Figura 4.4: Diagrama esquemático da mutação $M_TCRAdap2$.

Mutação por troca de consumidores com redução da distância total percorrida (M_TCDT)

Este operador também é utilizado no AG_DT . Esta mutação retira dois consumidores aleatoriamente, C_1 de uma rota R_1 e C_2 de uma rota R_2 . Verifica-se se existe a possibilidade de R_1 receber C_2 e R_2 receber C_1 . Caso as duas possibilidades sejam viáveis e ainda houver um ganho global na distância total percorrida a troca dos consumidores é realizada.

Mutação por retirada de consumidor de uma rota qualquer (M_RC)

Este operador já foi também descrito no capítulo anterior, sendo também utilizado no AG_DT . Um consumidor é aleatoriamente retirado de um veículo na tentativa de encontrar uma posição em outro veículo para o mesmo. O veículo destino também escolhido aleatoriamente. A posição de inserção no novo veículo é definida pelo menor custo adicional obtido naquele veículo, embora a solução como um todo possa ter piorado. Se o veículo sorteado para receber o consumidor não suportá-lo, outro veículo será sorteado até esgotar todos os veículos possíveis.

Mutação por retirada de consumidor da rota receptora (M_RCRR)

Este operador é equivalente ao M_RC , porém a rota escolhida para retirada de um consumidor não é aleatória, mas sim a *rota receptora*, definida na seção anterior.

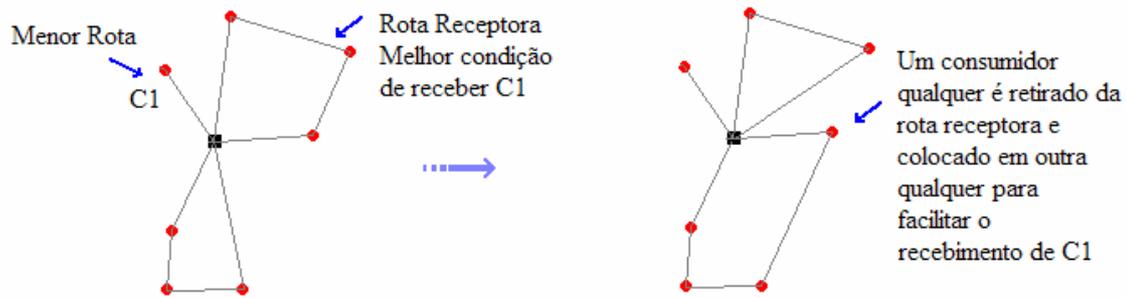


Figura 4.5: Diagrama esquemático da mutação M_RCRR .

Mutação por reinserção de consumidores com ganho em DT (M_RDT)

Este operador também foi utilizado em AG_DT . Esta mutação retira um consumidor aleatório por vez de um veículo e testa todas as posições possíveis neste mesmo veículo. Se há alguma nova posição com menor valor de DT , sua posição é atualizada.

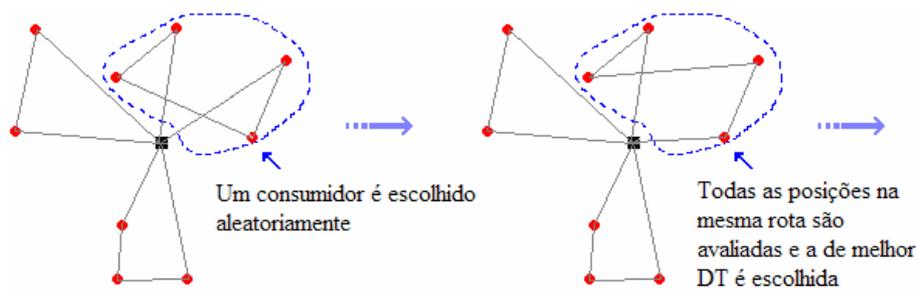


Figura 4.6: Diagrama esquemático da mutação M_RDT .

Mutação de eliminação de consumidores da rota receptora (M_ETCRR)

Este operador tenta retirar o maior número de consumidores possíveis da *rota receptora* (ver em M_RCRR qual é *rota receptora*). Todas as rotas e todas as posições são avaliadas como possíveis destinos para os consumidores retirados.

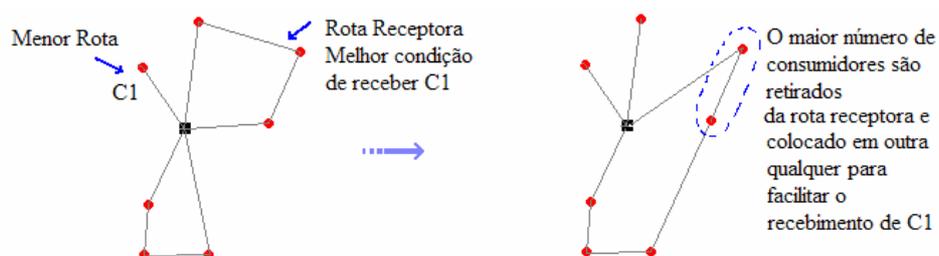


Figura 4.7: Diagrama esquemático da mutação M_ETCRR .

Mutação de eliminação do último consumidor da menor rota com troca de um consumidor em outra rota qualquer (M_EUTC1)

Este operador só está ativo quando a menor rota, em termos de número de consumidores, possui apenas um consumidor. Este operador retira este consumidor, tenta inseri-lo no lugar de algum outro. Todas as outras rotas e consumidores são avaliados. Caso seja possível, tenta encontrar uma posição para o consumidor retirado, uma vez que não há interesse em trazê-lo para a rota que está sendo eliminada. Se houver alguma rota onde este consumidor possa permanecer a menor rota terá sido eliminada.

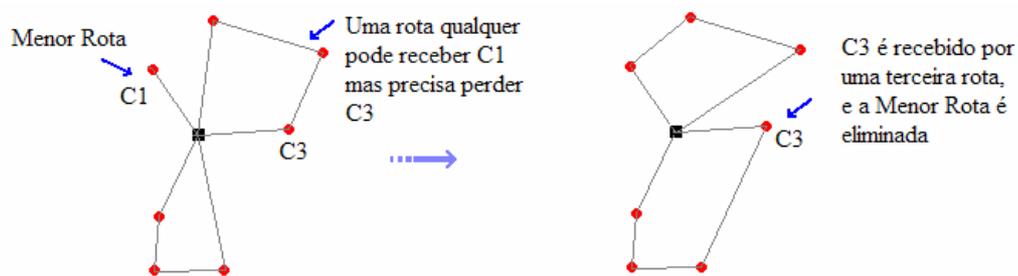


Figura 4.8: Diagrama esquemático da mutação M_EUTC1.

Mutação de eliminação do último consumidor da menor rota com troca de dois consumidores em outra rota qualquer (M_EUTC2)

Este operador também só está ativo quando resta apenas um consumidor na menor rota a ser eliminado. Este operador tenta inserir este último consumidor em alguma outra rota, mesmo que seja necessário retirar até dois consumidores da rota destino. Ele se baseia na idéia que as vezes é mais difícil encontrar uma posição para este último consumidor do que para outros dois. Se os dois consumidores consecutivos eliminados podem ser inseridos em algum outro local, a troca será feita. Se isso não for possível com os dois, mas com apenas um (após a tentativa com dois ter sido realizada), a operação é realizada e o consumidor restante é colocado na menor rota.

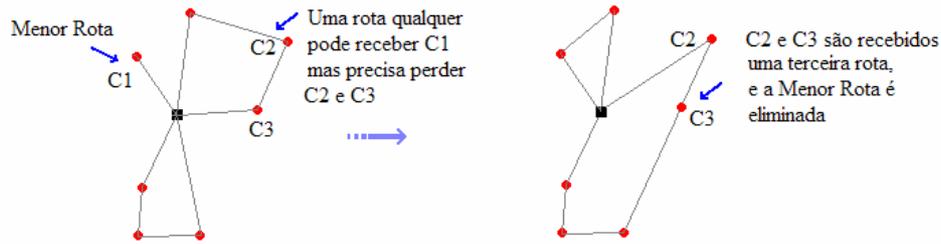


Figura 4.9 : Diagrama esquemático da mutação M_{EUTC2} .

Mutação de eliminação de toda a menor rota (M_{EUR})

Este operador, busca por uma posição viável para todos os consumidores da menor rota, em alguma outra rota, na tentativa de eliminá-la de uma só vez. Geralmente é um operador que têm bastante utilidade nas primeiras gerações, onde ainda existem muitas rotas que ainda podem receber novos consumidores.

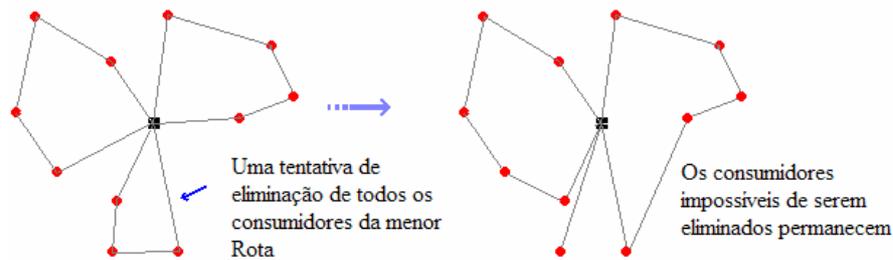


Figura 4.10 : Diagrama esquemático da mutação M_{EUR} .

4.3 Encontrando soluções em número de veículos para o PRVJT estático

Uma nova heurística de geração de colunas foi proposta para a minimização de NV no PRVJT, chamada HGC_NV. Esta heurística é composta de duas fases bem distintas. Na primeira fase, uma parcela do tempo disponível é dedicada exclusivamente à redução do número de veículos, utilizando o AG_NV descrito na seção anterior, em uma única execução. A última geração do AG_NV é então passada para o AG_DT com a função objetivo modificada, conforme foi descrito na seção 4.2. Como a primeira geração de AG_DT é obtida de AG_NV, a heurística construtiva de rotas PFIH estocástico não é utilizado. Os demais parâmetros do AG_DT foram mantidos, conforme descritos no capítulo anterior. A função objetivo do modelo do PPC também foi alterada conforme

equação (4.1). As demais características de HGC_DT são mantidas, repetindo-se também os ciclos de diversificação e intensificação, gerando rotas para o conjunto R_{GLOBAL} , que será tratado finalmente pelo problema principal.

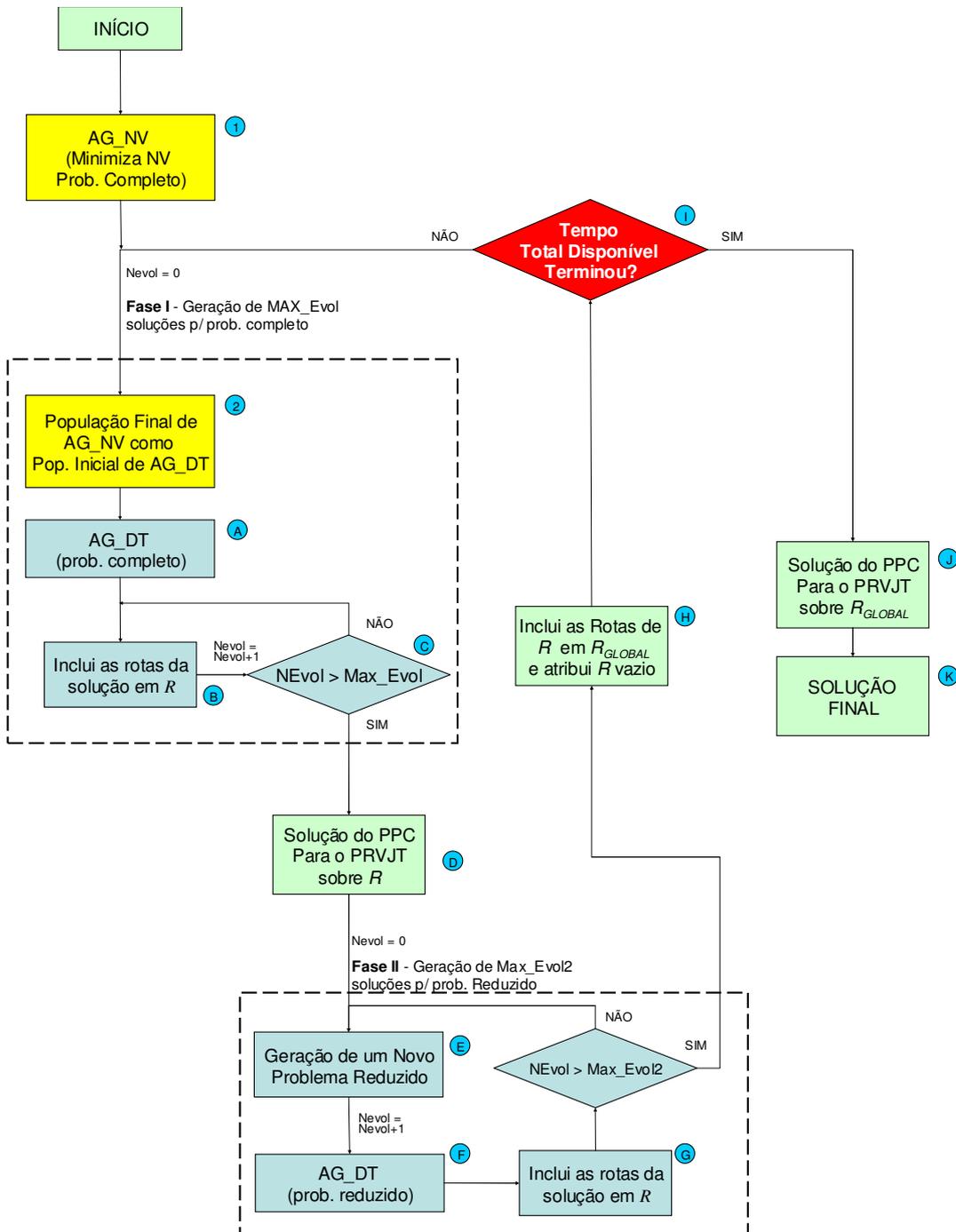


Figura 4.11 : Diagrama completo de HGC_NV utilizado para resolver o PRVJT estático com o objetivo (NV, DT), nesta ordem.

O diagrama da nova solução para o PRVJT pode ser visto na Figura 4.11. O fluxograma completo tem agora duas novas etapas, a minimização de NV como uma execução única de AG_NV, bloco 1, e a inserção de sua população final em AG_DT, bloco 2 do diagrama.

4.4 Conclusões

Neste capítulo foram apresentadas alterações na proposta de solução do capítulo 3, para contemplar a minimização do número de veículos no PRVJT estático. Um novo algoritmo genético AG_NV foi apresentado, utilizando uma seleção por torneio baseada em uma hierarquia de indicadores de adaptabilidade. Novos operadores também foram propostos, adequando-se a nova direção de busca pretendida. Um fluxo completo modificado foi proposto, funcionando em duas fases, onde a primeira corresponde à minimização do número de veículos pelo AG_NV, e a segunda minimizando a distância percorrida, mas mantendo o número de veículos previamente encontrado. A avaliação dos resultados será apresentada no capítulo 7.

Capítulo 5

Minimizando a Distância Percorrida no PRVJT Dinâmico

Neste capítulo será apresentada uma solução para o PRVDJT. Na seção 5.1 a proposta de solução para o problema dinâmico é situada dentre os diferentes tipos de abordagens na literatura. Na seção 5.2 são definidas as variáveis dinâmicas do problema e um cenário hipotético, com as alterações necessárias ao modelo estático apresentado nos capítulos anteriores. Na seção 5.3 a heurística HGC_DT do capítulo 3 é modificada para se adequar ao problema dinâmico proposto. Por fim, na seção 5.4 as conclusões são apresentadas.

5.1 Introdução

Embora tenha crescido o número de trabalhos inserindo algum tipo de dinamicidade no PRVJT, ainda não há uma linha comum para a modelagem do problema. Existe uma enorme diversidade de problemas de roteamento de veículos considerados dinâmicos. Também é grande a variedade de propostas de solução. Os principais tipos de abordagens quanto ao tratamento da dinamicidade foram apresentadas na seção 2.5. O primeiro tipo, amplamente utilizado, é o princípio da re-otimização. Assim que novas informações são conhecidas, um novo problema é resolvido, sem qualquer aproveitamento da etapa anterior. O segundo tipo discutido foi o tratamento *a priori*, onde o roteamento é feito considerando uma distribuição de probabilidades, cabendo a cada veículo atender um conjunto de consumidores fixos e aqueles que se confirmarem dentro de outro conjunto de consumidores duvidosos, não havendo troca de consumidores ao longo do processo. Por fim, os conhecidos algoritmos de otimização em tempo real, ou *real time optimization algorithms* ou ainda *online optimization*. A principal característica de um algoritmo de otimização em tempo real é que uma nova informação pode chegar em um instante e alguns segundos depois o resultado ser necessário. Para isso é muito desejado que a etapa anterior, onde houve tempo maior para a busca pela solução, seja aproveitada ao máximo.

As heurísticas apresentadas nos capítulos anteriores, HGC_DT para minimização da distância total percorrida e HGC_NV para minimização do número de veículos possuem características bem interessantes para tratamento do problema dinâmico, o PRVDJT. Especificamente a heurística HGC_DT será avaliada nesta tese, embora a proposta se aplique também para HGC_NV. Pequenas modificações serão propostas para a heurística de forma que possa tratar novas informações a qualquer instante, aproveitando sempre que possível ao máximo o trabalho de busca já realizado. Com isso, respostas rápidas são permitidas, características que torna a proposta atual uma opção competitiva entre os algoritmos de tempo real para o PRVJTD. No capítulo 8 a estratégia será avaliada através de problemas testes, conforme diferentes graus de dinamicidade e limites de tempo distintos.

5.2 Definindo as variáveis dinâmicas

Em vários cenários reais não há sequer minutos onde já se conhece 100% as variáveis do problema de forma estática. Conseqüentemente se faz necessário iniciar o processo de otimização com as variáveis que até então são conhecidas, dando um tratamento diferenciado às informações que chegam posteriormente. Este capítulo descreve a utilização da heurística HGC_DT como um método *online*, isto é, as variáveis até então desconhecidas não exercem nenhuma influência sobre a solução. Somente após uma nova variável ser conhecida, a solução deverá ser modificada para contemplar tais variáveis. Deseja-se, entretanto, que algum aproveitamento da busca já realizada seja obtido, permitindo uma vantagem em qualidade e tempo de resposta sobre a simples re-otimização.

Conforme a definição adotada neste trabalho [LA00], um problema dinâmico é caracterizado pelas seguintes características:

A. Assume-se que não são conhecidas todas as informações relevantes antes do início do processo de roteamento.

B. As informações relevantes depois de iniciado o processo de roteamento podem sofrer alterações.

Toda informação que faz parte da formulação do problema, como posição dos consumidores, janela de tempo, demanda de cada consumidor, capacidade de carga dos veículos, tempo de trajeto entre consumidores e outras mais, podem ser tratadas de forma dinâmica. Para isso, elas precisam ser modificadas ou inseridas após o processo de otimização ou roteamento ter sido iniciado.

Propostas para o tratamento dinâmico do PRVJT têm sido mais frequentes nos últimos anos, mas ainda persiste uma grande dificuldade em avaliar a qualidade dos algoritmos propostos. Seria de bastante utilidade, por exemplo, um limite inferior de distância percorrida para estes problemas de roteamento dinâmicos, permitindo se conhecer o *gap* produzido pelas diversas heurísticas propostas. Mas até mesmo o cálculo de um limite inferior para alguns tipos de PRVJT dinâmico parece não ser trivial, pois existem variáveis estocásticas na formulação do problema, dificultando bastante este caminho.

Lund [LMR96] avalia sua proposta de método *online* sobre uma variante dinâmica da instância bem conhecida de Solomon R101, com 100 consumidores. O autor considera que cada encomenda somente será conhecida em um instante randômico dentro do intervalo $[t_i - \Delta t, t_i + \Delta t]$, onde $t_i = \alpha a_i$, a_i é o início da janela de tempo do consumidor i , α uma constante que define o grau de dinamicidade do problema e Δt um valor randômico de aproximadamente 10 minutos. Quanto menor a constante α mais cedo se conhece os pedidos, diminuindo o grau de dinamicidade. Uma vez que os veículos já estão em trânsito quando podem chegar outros novos consumidores fica difícil avaliar qual deva ser o resultado desejado para a instância R101 modificada.

Allan Larsen [Lar00] por sua vez, realiza outras modificações também na instância original R101, para tornar o problema dinâmico, e mais uma vez fica difícil avaliar os resultados. As distâncias percorridas para R101 variam de 2100 a 3100 unidades de distância euclidianas, conforme o grau de dinamismo utilizado, enquanto a melhor distância conhecida para o problema estático é 1642,77. No entanto, essa grande diferença contém tanto as possíveis deficiências do algoritmo como também um acréscimo inerente a variação do problema, uma vez que no modelo proposto os veículos devem partir antecipadamente para atender alguns consumidores, sem o conhecimento do problema como um todo.

Em outras palavras, o algoritmo de otimização *online* ideal seria um algoritmo capaz de encontrar a solução ótima exata, consumindo um tempo nulo de processamento, a qualquer instante. Desta forma que, instantaneamente, a cada nova variável conhecida, executa-se novamente o algoritmo e se terá sempre a solução ótima naquele momento.

Observa-se que para os cenários propostos na literatura, mesmo utilizando um algoritmo ideal, se verifica um acréscimo no valor da função objetivo devido à dinamicidade, se comparado ao caso estático, uma vez que decisões foram tomadas (mesmo que dentro de uma solução ótima) sem o conhecimento do problema como um todo.

Na literatura atual, enquanto os melhores resultados para o PRVJT estático têm sido alcançados por metaheurísticas, as propostas de solução para o PRVJT dinâmico têm utilizado regras bem simples para contemplar informações conhecidas durante o processo de otimização. É comum, por exemplo, a simples inserção de novos consumidores na solução atual, geralmente depois de encerrado o processo de busca do algoritmo principal estático. Larsen menciona a dificuldade em se tratar o PRVDJT com metaheurísticas devido ao tempo geralmente necessário para gerar um resultado de qualidade por estes tipos de algoritmos.

Neste trabalho, a utilização de uma variação dinâmica alternativa das instâncias tradicionais de Solomon permitirá uma melhor comparação com os resultados estáticos já tão trabalhados na literatura. Desta forma, a principal característica desejada de um algoritmo *online*, que é a resposta rápida quando submetido ao conhecimento de alterações nas variáveis do problema, poderá ser isolada das alterações do problema em si, permitindo uma melhor avaliação dos resultados. Nas propostas anteriores, quando se altera o grau de dinamicidade do problema, altera-se também o valor ótimo da função objetivo almejado. Na proposta atual, o aumento do grau de dinamicidade do problema irá dificultar o trabalho do algoritmo *online*, exigindo respostas mais rápidas e o máximo aproveitamento do processo de busca já despendido, mas o valor da função objetivo almejado se mantém o mesmo. Isso significa que todo o aumento no resultado em DT para uma instância dinâmica qualquer proposta, se comparado com o valor desejado ou obtido para a instância correspondente estática, representa a dificuldade do algoritmo em si.

Para facilitar a visualização do cenário proposto, considera-se um tipo de serviço de coleta, como, por exemplo, coleta de lixo industrial, coleta de exames de um laboratório, entre outros. Todos os veículos saem vazios do depósito central e realizam coletas em um conjunto de clientes. Isso facilita o tratamento da dinamicidade uma vez que não é necessário considerar a situação de se carregar os veículos antes do despacho efetivo dos mesmos. São considerados dois cenários distintos, o Cenário A considera a chegada de novos consumidores após o processo de otimização já estar em um estágio avançado. O Cenário B considera o cancelamento de alguns consumidores, conforme está detalhado a seguir:

Cenário A: Chegada de novas coletas

1. Uma parcela dos pedidos de coletas (N_1 consumidores) é conhecida previamente;
2. Uma outra parcela dos pedidos de coletas (N_2 consumidores) será conhecida durante o processo de otimização;
3. Os pedidos novos podem surgir a qualquer momento, até o instante da saída dos veículos do depósito central;
4. No instante da partida dos veículos encerra-se o processo de otimização, definindo as rotas dos veículos com as coletas atuais;

Caso 2: Cancelamento de coletas previstas

1. Um conjunto de pedidos de coleta (N_1 consumidores) é conhecido previamente;
2. Parte desses pedidos de coleta (N_2 consumidores) serão cancelados durante o processo de otimização;
3. Cancelamentos podem surgir a qualquer momento, até o instante da saída dos veículos do depósito central;
4. No instante da partida dos veículos encerra-se o processo de otimização, definindo as rotas dos veículos;

A modelagem do problema dinâmico como descrito foi motivado principalmente por possibilitar uma comparação direta com a solução tradicional estática do PRVJT. Deseja-se que no estágio final do problema, seja após a inserção ou o cancelamento de consumidores, os dados sejam os mesmos de um problema estático já testado anteriormente, desta forma fica possível considerar os resultados desejados da solução do problema dinâmico.

Foram consideradas as características desejadas de um método *online* de solução para o problema dinâmico:

- Capacidade de aproveitar ao máximo a busca já realizada até o conhecimento das novas variáveis, permitindo uma vantagem à simples re-inicialização do algoritmo;
- Capacidade de aperfeiçoamento rápido da solução atual, condizente com a necessidade em tempo real do problema dinâmico;

5.3 Modificando o macrofluxo de HGC_DT

A heurística HGC_DT já foi concebida visualizando o tratamento do problema dinâmico, por isso poucas modificações serão necessárias no macrofluxo exposto no capítulo 3. A principal vantagem de HGC_DT para o tratamento dinâmico do problema é que *a capacidade de inclusão ou modificação dos dados do problema durante o processo é muito natural devido à independência entre as muitas execuções distintas do algoritmo genético AG_DT, e ao mesmo tempo, o reaproveitamento das informações já obtidas do espaço de busca, representado pelas diversas opções de rotas que podem continuar interessantes no grande conjunto R_{GLOBAL} , mesmo após a alteração de alguns consumidores.*

Existem quatro pontos do algoritmo onde foi considerada a possibilidade de alteração nas variáveis do problema, neste contexto de inclusão ou exclusão de consumidores. Conforme mostrado no fluxograma da Figura 5.1, o algoritmo é basicamente o mesmo, no entanto, nos pontos indicados pelo bloco hexagonal podem haver inclusão ou exclusão de consumidores. O algoritmo proposto não é preemptivo, ou seja, não há interrupção instantânea da tarefa sendo executada quando há um recebimento de uma nova informação. Ao contrário, há uma checagem por novas informações (*pooling*) toda vez que o algoritmo

passa por determinados pontos pré-estabelecidos. Embora um sistema preemptivo seja ideal neste caso para que a informação atualizada seja considerada o quanto antes, o *pooling* utilizado ocorre a cada intervalo entre gerações no AG_DT. O tempo máximo entre duas gerações não ultrapassa 2 segundos, sendo a média de tempo ainda menor. Por isso não houve necessidade de uma estrutura preemptiva mais elaborada no projeto de HGC_DT.

Inclusão e retirada de consumidores no AG_DT

Conforme a Figura 5.1, poderá haver inclusão ou retirada de consumidores no AG_DT, tanto no tratamento do problema completo, bloco A, quanto no tratamento do problema reduzido, bloco E.

Se houver um novo consumidor antes da geração da população inicial, nada precisa ser feito no AG_DT, apenas incluir o consumidor na estrutura do problema, uma vez que o PFIH estocástico irá rotear o mesmo naturalmente. O mesmo ocorre quando há uma retirada de consumidor nesta fase do algoritmo.

Entre duas gerações, a inclusão de um consumidor é tratada fazendo uma inserção do novo consumidor na melhor posição, ou seja, na ordem de atendimento que representar o menor acréscimo em DT, avaliando todos os veículos não vazios. Isso corresponde em aplicar o PFIH novamente, mas neste caso em uma população já existente.

Já a exclusão de um consumidor entre duas gerações é ainda mais direta, pois basta percorrer todos os indivíduos do AG_DT, eliminando aquele consumidor. Este também é retirado da estrutura de dados do problema para que não retorne na execução seguinte de AG_DT.

Quando o problema reduzido está sendo tratado, conforme foi discutido no capítulo 3, os consumidores que serão tratados pelo AG_DT são aqueles presentes em rotas escolhidas aleatoriamente. Neste caso, optou-se por incluir o novo consumidor, em todos os subproblemas, pois havendo alternância das rotas escolhidas, haverá maior probabilidade de rapidamente encontrar a melhor posição para o mesmo. O ponto negativo é que este irá atrapalhar um possível aperfeiçoamento que poderia ocorrer nas rotas que não devam recebê-lo, pois irá acarretar um acréscimo de DT, gerando rotas ruins para o conjunto

R_{GLOBAL} do PPC. Mas é um preço necessário, pois não se pode prever em qual conjunto de rotas este deverá ser inserido.

A exclusão de um consumidor no tratamento do problema reduzido é feita pela eliminação deste consumidor da solução parcial utilizada como referência para gerar os subconjuntos de consumidores.

Inclusão e retirada de consumidores antes da solução do PPC

Como os conjuntos R e R_{GLOBAL} estão sendo gerados com a inserção de rotas durante várias execuções do AG_DT, pode haver rotas com consumidores que já foram cancelados do problema. Por isso, antes da execução do algoritmo de programação linear, sobre o modelo do PPC, todo o conjunto de rotas deve ser revalidado. No caso de exclusão de consumidores, há uma busca em todo o conjunto de rotas e os consumidores excluídos do problema são simplesmente eliminados das rotas. Todas as rotas modificadas devem ter seu custo reavaliado, o que é feito também neste momento.

Já quanto aos novos consumidores, trabalhar as rotas dos conjuntos R e R_{GLOBAL} não é tão trivial, pois não se deve inserir os consumidores em todas elas, pois algumas não deverão atender tais consumidores. Por isso, optou-se por inserir estes em uma única rota, naquela que melhor receber tais consumidores com o mínimo acréscimo em DT. Note que os conjuntos R e R_{GLOBAL} possuem muitas opções de rotas diferentes para os demais consumidores, e poucas para os mais recentes. Se houvesse somente um consumidor a ser incluído, uma boa opção seria duplicar as rotas presentes nestes conjuntos, onde todas as rotas existentes seriam avaliadas com e sem o novo consumidor. Entretanto, como são vários consumidores, optou-se pela primeira alternativa.

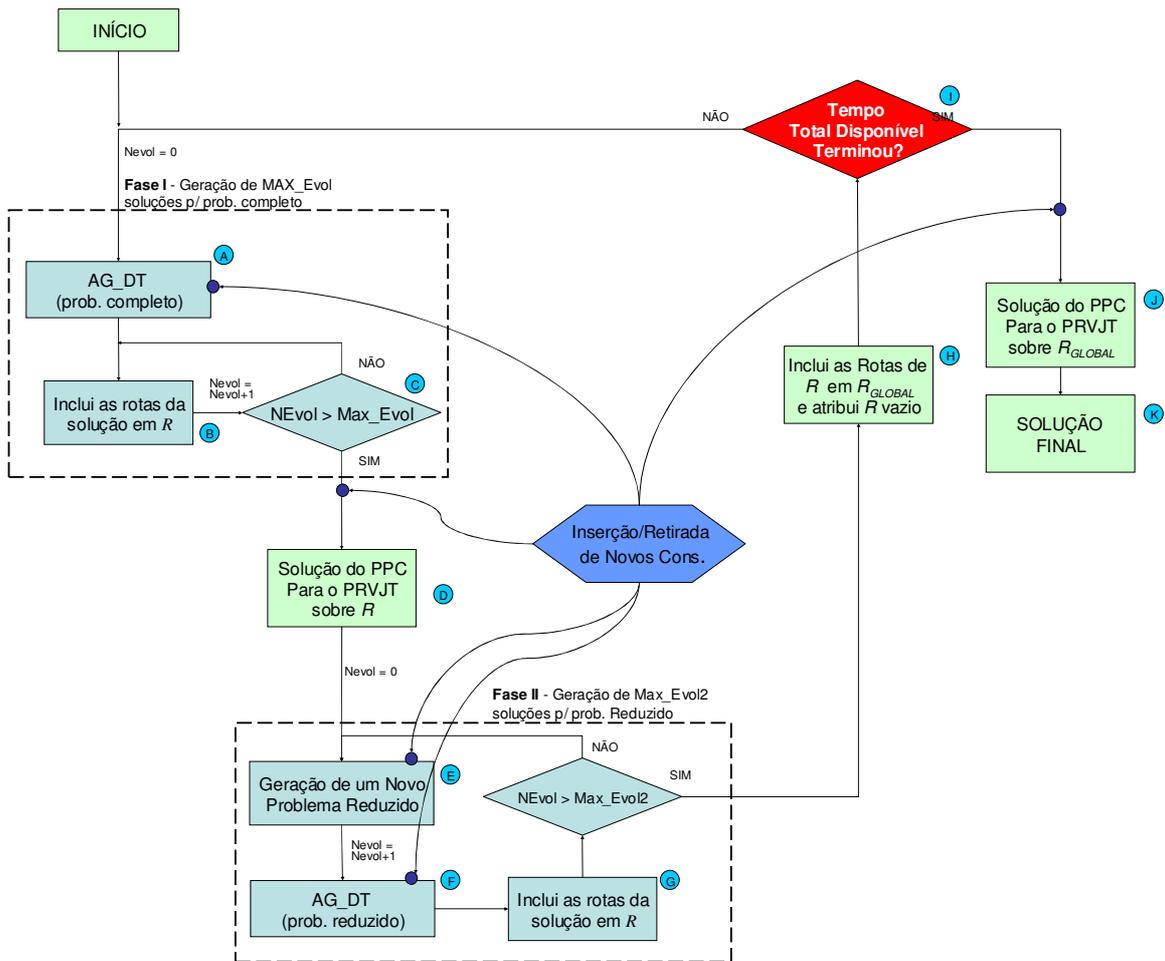


Figura 5.1: Diagrama de solução de HGC_DT. Os pontos circulares ligados ao bloco hexagonal correspondem às etapas de inclusão e retirada de novos consumidores.

5.4 Conclusões

Neste capítulo foi proposta uma variante dinâmica para as instâncias de Solomon que permitisse uma meta clara para um método de otimização *online*, onde os resultados esperados são os mesmos que aqueles do problema estático, permitindo uma melhor avaliação do algoritmo.

Foi também apresentado como a heurística de geração de colunas HGC_DT proposta no capítulo 3 pôde ser facilmente adaptada para o problema dinâmico. A variável dinâmica utilizada foi o conjunto de consumidores do problema, sendo possível a inserção e o cancelamento de consumidores. Os resultados de HGC_DT para o problema dinâmico são apresentados no capítulo 8.

Capítulo 6

Resultados Minimizando a Distância Percorrida no PRVJT Estático

Neste capítulo são apresentados os resultados de HGC_DT para a minimização da distância total percorrida no PRVJT estático. HGC_DT foi testada no conjunto de problemas de Solomon do PRVJT e outros 44 problemas do PRVC já bem testados por métodos exatos. Os resultados foram produzidos utilizando cálculos com dupla precisão e com apenas uma casa decimal para as instâncias de Solomon, possibilitando a comparação tanto com resultados produzidos por heurísticas como por métodos exatos. Já para o problema capacitado (PRVC) foi utilizado arredondamento para o inteiro mais próximo. Na seção 6.1 são apresentados os problema de Solomon [SM87] utilizados para avaliação de HGC_DT. Na seção 6.2 são apresentados os procedimentos de calibração dos parâmetros da heurísticas e os resultados obtidos. Por fim, na seção 6.3 são apresentadas as conclusões sobre os resultados de HGC_DT.

6.1 O conjunto de problemas testes de Solomon

O conjunto de problemas testes utilizados nesta tese foi proposto por Solomon em 1987 [SM87] para o PRVJT, baseado em alguns dados de problemas utilizados por Christofides [CMT79] para o PRV tradicional, ou *Vehicle Routing Problem (VRP)*. Trata-se de diferentes classes de instâncias, cada qual com características geográficas e de restrições características.

Os problemas foram propostos nas dimensões 25, 50 e 100 consumidores. São também divididos em seis grupos: R1, R2, C1, C2, RC1 e RC2. Em todos os casos, os consumidores estão geograficamente distribuídos em um quadrado no plano XY com dimensões 100x100.

Alterando alguns critérios de disposição geográfica e o conjunto de restrições é possível

uma variedade de soluções bem distintas, como por exemplo, grande variação no número de consumidores sendo atendidos pelo mesmo veículo. Outra variação importante é a diminuição ou aumento da janela de tempo dos consumidores. Se por um lado, pequenos intervalos como janelas de tempo diminuem significativamente o espaço de busca, o que pode ser desejável por um lado, requer complexos operadores de obtenção de vizinhança nas heurísticas de busca local. As diferentes classes de problemas de Solomon têm como objetivo estabelecer cenários distintos para que as propostas de solução possam ser avaliadas de forma mais abrangente possível.

Nos grupos R1 e R2 os consumidores estão posicionados de forma completamente randômica, ver Figura 6.1. Nos grupos C1 e C2 os consumidores estão posicionados em agrupamentos, ver Figura 6.2. E os grupos RC1 e RC2 são tipos mistos, parte dos consumidores posicionados de forma aleatória e parte em agrupamentos, ver Figura 6.3. Considera-se a distância euclidiana entre os consumidores como os custos de cada trecho. Para percorrer cada unidade de distância os veículos consomem uma unidade de tempo. As classes cujo nome tem o segundo dígito em 1 (R1, C1 e RC1) distinguem-se das classes com segundo dígito em 2 (R2, C2 e RC2) pela dimensão do intervalo das janelas de tempo. Nas primeiras as janelas de tempo e também o horizonte total são bem mais curtos. Isso resulta em um número de rotas mínimas necessárias para atender todos os consumidores muito maior para as classes do primeiro tipo. O número de problemas existentes nas classes R1, R2, C1, C2, RC1 e RC2 são 12, 11, 9, 8, 8 e 8, respectivamente. Para cada classe, o posicionamento dos consumidores é mantido constante, bem como suas demandas e o tempo de serviço, variando somente as janelas de tempo. A janela de tempo é entendida como a restrição imposta ao tempo de chegada do veículo no consumidor, que deve estar no intervalo $[a_i, b_i]$, exatamente como no modelo da equação (2.1).

Por representarem relativamente bem os cenários mais comuns do PRVJT têm sido amplamente empregadas para avaliar as diversas propostas de solução. Porém, a falta de definição de alguns pontos resultou em várias abordagens diferentes, dificultando a comparação direta dos resultados. A primeira diz respeito ao objetivo do problema. Os objetivos mais comuns encontrados na literatura são minimização da distância total percorrida, do número de veículos, tempo total de espera e combinação destes.

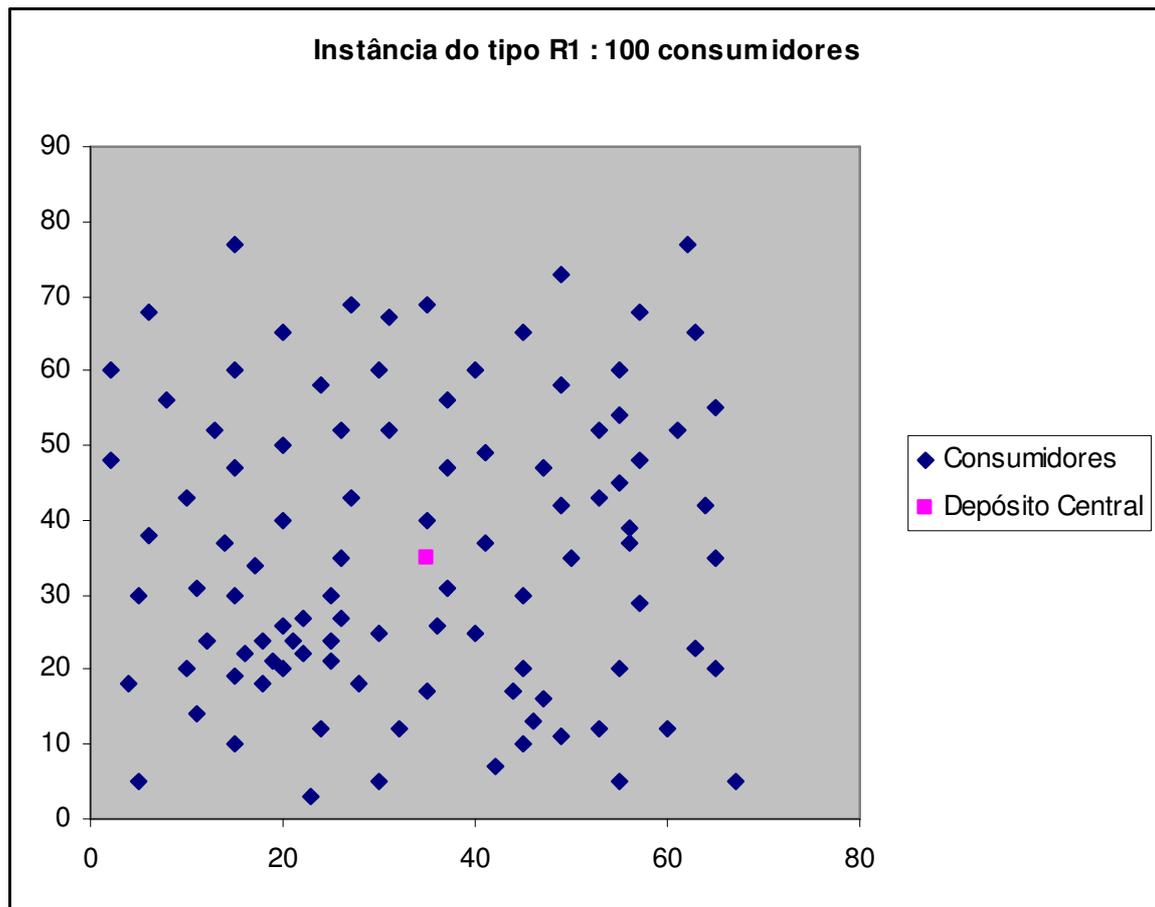


Figura 6.1: Disposição espacial dos consumidores nos problemas das classes R1 e R2.

Quanto à janela de tempo, a maioria dos autores tem considerado esta restrição do tipo forte, ou seja, qualquer violação é proibida, gerando soluções inviáveis. Outra opção, menos comum é penalizar a função objetivo pelas violações de janelas de tempo, conhecido como *soft time window*. E, finalmente, quanto ao critério de arredondamento e os tipos de variáveis utilizadas. Os trabalhos com métodos exatos têm optado pela utilização de variáveis inteiras. Para isso, os dados de entrada são multiplicados por dez e truncados. A argumentação dos autores é que isso permite uma melhor comparação, pois o tempo para cálculos em ponto flutuante é muito mais dependente do hardware utilizado. Já os trabalhos utilizando métodos heurísticos têm optado por tratar dos dados em ponto flutuante de dupla precisão (64 bits).

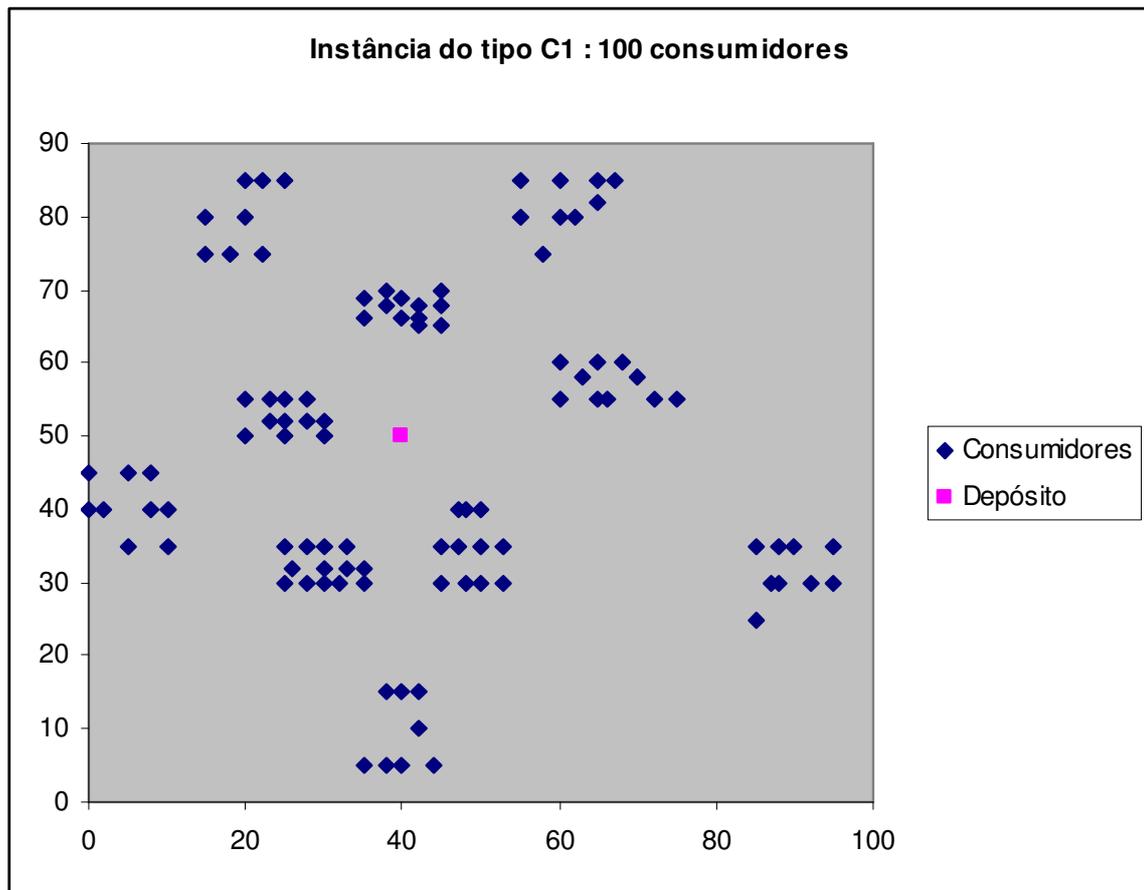


Figura 6.2: Disposição espacial dos consumidores nos problemas das classes C1 e C2.

Embora existam estas desvantagens, as instâncias de Solomon ainda são a melhor forma de avaliar propostas de solução para o PRVJT, devido à grande adoção na literatura. Adicionalmente, outros conjuntos têm sido propostos como expansão das instâncias iniciais, aumentando o número de consumidores.

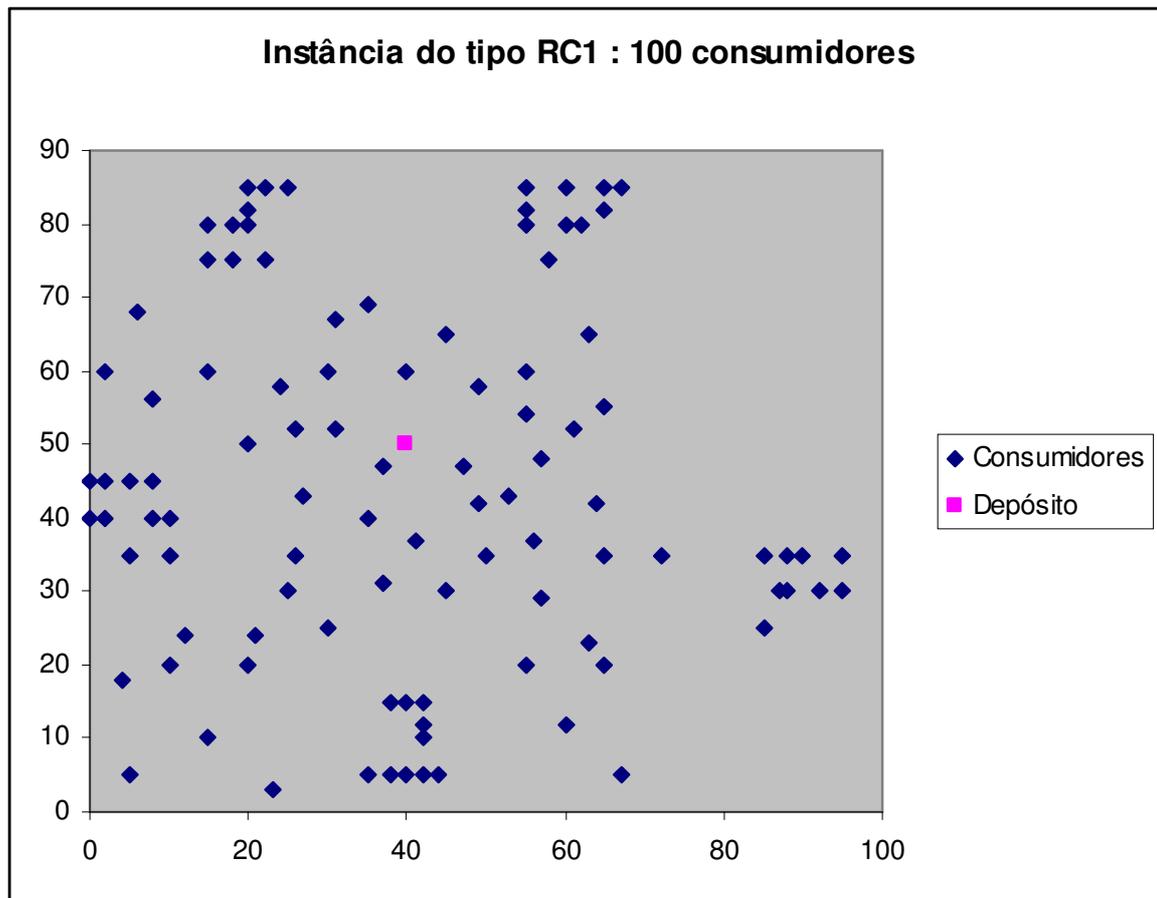


Figura 6.3: Disposição espacial dos consumidores nos problemas das classes RC1 e RC2.

6.2 Avaliando a heurística HGC_DT para minimização de distância total percorrida

A heurística de geração de colunas HGC_DT proposta no Capítulo 3 foi avaliada primeiramente com as instâncias de Solomon de 100 consumidores já descritas. Embora vários trabalhos têm sido propostos tratando este conjunto de problemas, ainda não se tem conhecimento de uma heurística minimizando unicamente a distância total percorrida que faça uso de truncamento na primeira casa decimal, possibilitando a comparação dos resultados com os trabalhos de métodos exatos, pois este critério pode alterar significativamente os resultados.

Adicionalmente, testes utilizando aritmética real (dupla precisão) também foram realizados, desta vez comparando os resultados com aqueles produzidos por heurísticas,

que também utilizam o mesmo procedimento. Poucos trabalhos com métodos heurísticos têm tratado distância percorrida como primeiro objetivo, fazendo dos resultados obtidos na grande maioria das instâncias as atuais metas (*benchmark*) da literatura, neste critério.

Por fim, HGC_DT é testada em outro conjunto de problemas de roteamento da literatura, agora sem restrição de janela de tempo (PRVC), uma vez que este conjunto também já foi amplamente utilizado em diversos trabalhos envolvendo métodos exatos. Se trata de vários problemas de diversos autores, disponíveis em www.branchandcut.org/VRP. Os testes foram feitos vários problemas das séries A, B, E, M e P.

6.2.1 Configuração dos parâmetros de HGC_DT

Configurando alguns parâmetros do Algoritmo Genético AG_DT

Conforme diagrama da Figura 3.16 no capítulo 3, o processo de solução é iniciado gerando vários mínimos locais, que são resultados individuais de soluções produzidas pelo AG_DT. A qualidade da solução final, embora não dependa de uma única evolução do AG_DT, é altamente dependente das qualidades das rotas geradas por este. Conseqüentemente, o bom desempenho de HGC_DT será melhor quanto melhor for os mínimos locais gerados por AG_DT.

Alguns dos parâmetros de configuração do AG_DT foram configurados empiricamente. Para os parâmetros mais críticos, ou seja, aqueles que têm uma maior influência na qualidade dos resultados, utilizou-se um terceiro algoritmo genético que realizasse uma busca por parâmetros os mais adequados possíveis, tendo como função adaptabilidade os resultados destes parâmetros no desempenho de AG_DT. Estes parâmetros são basicamente o número de mutações realizadas sobre os indivíduos solução para diversos problemas. Este algoritmo, denominado AG_PARAM será descrito posteriormente.

O primeiro parâmetro ajustado no AG_DT foi o número de indivíduos da população, sendo este fixo ao longo de todo o algoritmo. É sabido que os parâmetros são interdependentes, logo o melhor tamanho de população é dependente do conjunto de operadores de mutação e também do tipo de cruzamento utilizado. Por isso, um primeiro ajuste muito rudimentar foi feito, avaliando a diversificação dos indivíduos e avaliando os resultados para algumas

poucas instâncias escolhidas aleatoriamente. Para uma população inicial de 50 indivíduos, ajustou-se a quantidade de mutações e número de indivíduos do torneio (seleção) de forma que não houvesse mais que 5 indivíduos repetidos ao longo das populações nas diferentes gerações. Para tal foi adicionado um contador de indivíduos com adaptabilidades (*fitness*) iguais ao do melhor indivíduo, e acompanhado visualmente. As mutações foram então ajustadas para permitir uma diversificação suficiente, mas que não gerasse adaptabilidades muito distantes em uma mesma população. Esse procedimento bem rudimentar foi inicial, para permitir condições de ajuste do número de indivíduos da população de maneira mais criteriosa, embora ainda empiricamente. A taxa de mutação foi fixada linearmente proporcional ao número de indivíduos, para não ser ponto decisivo quando na alteração do número de indivíduos da população.

Para o ajuste final do número de indivíduos, as instâncias de Solomon R101, R201, RC101, RC201, C101 e C201 foram executadas dez vezes cada, utilizando 20, 30, 50, 75 e 100 indivíduos. O tempo de execução foi de 1 minuto. Este valor do tempo de execução foi definido em função da necessidade de executar o AG_DT diversas vezes, como parte da heurística global HGC_DT, não podendo por isso ser muito longo. No entanto, este tempo foi fixado posteriormente em 2 minutos para os testes finais, resultado de outros testes posteriores realizados. Os resultados são mostrados na Tabela 6.1.

Tabela 6.1: Avaliação dos resultados de AG_DT com a variação do número de indivíduos da população.

Resultados para 20 indivíduos na população de AG DT												
Instâncias	R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	21	1673,3	6	1253,9	10	828,9	3	591,6	16	1699,6	7	1343,6
2	20	1664,4	5	1293,7	10	828,9	3	591,6	12	1698,4	7	1401,7
3	20	1679,9	5	1267,7	10	828,9	3	591,6	17	1709,1	6	1462,9
4	20	1673,4	7	1208,5	10	828,9	3	591,6	17	1674,0	7	1340,6
5	20	1671,9	5	1254,1	10	828,9	3	591,6	17	1692,6	6	1398,2
Melhor DT	20	1664,4	7	1208,5	10	828,9	3	591,6	17	1674,0	7	1340,6
Média	20	1672,6	5,6	1255,6	10	828,9	3	591,6	16	1694,8	6,6	1389,4
Desvio P.	0,4	5,5	0,9	30,9	0	0,0	0	0,0	2,2	13,0	0,5	50,3
Continua na próxima pagina												

Continuação da página anterior												
Resultados para 30 indivíduos na população de AG_DT												
Instâncias	R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	20	1673,5	6	1246,5	10	828,9	3	591,6	18	1713,5	6	1419,9
2	20	1670,0	6	1253,9	10	828,9	3	591,6	17	1683,2	5	1475,9
3	20	1670,1	6	1218,8	10	828,9	3	591,6	17	1706,3	8	1330,9
4	20	1673,5	6	1210,4	10	828,9	3	591,6	17	1689,2	6	1446,9
5	20	1646,8	6	1200,8	10	828,9	3	591,6	17	1692,6	7	1351,2
Melhor DT	20	1646,8	7	1200,8	10	828,9	3	591,6	17	1683,2	7	1330,9
Média	20	1666,8	6	1226,1	10	828,9	3	591,6	17	1696,9	6,4	1405,0
Desvio P.	0	11,3	0	23,1	0	0,0	0	0,0	0,4	12,6	1,1	62,0
Resultados para 50 indivíduos na população de AG_DT												
Instâncias	R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	20	1670,1	7	1251,6	10	828,9	3	591,6	17	1679,1	5	1426,0
2	20	1673,0	7	1261,3	10	828,9	3	591,6	16	1683,0	7	1428,8
3	20	1673,1	6	1294,8	10	828,9	3	591,6	18	1715,2	7	1413,3
4	20	1669,8	8	1232,7	10	828,9	3	591,6	18	1697,8	6	1413,3
5	20	1650,2	5	1298,3	10	828,9	3	591,6	16	1710,1	7	1387,0
Melhor DT	20	1650,2	7	1232,7	10	828,9	3	591,6	17	1679,1	7	1387,0
Média	20	1667,3	6,6	1267,7	10	828,9	3	591,6	17	1697,0	6,4	1413,7
Desvio P.	0	9,6	1,1	28,3	0	0,0	0	0,0	1	16,0	0,9	16,5
Resultados para 75 indivíduos na população de AG_DT												
Instâncias	R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	20	1672,7	6	1243,5	10	828,9	3	591,6	17	1714,0	6	1347,1
2	20	1666,5	7	1220,7	10	828,9	3	591,6	17	1670,5	6	1416,0
3	20	1672,1	6	1262,6	10	828,9	3	591,6	16	1651,2	6	1440,0
4	20	1670,0	6	1218,6	10	828,9	3	591,6	17	1677,5	7	1390,6
5	20	1657,7	5	1309,5	10	828,9	3	591,6	17	1688,5	7	1344,4
Melhor DT	20	1657,7	7	1218,6	10	828,9	3	591,6	17	1651,2	7	1344,4
Média	20	1667,8	6	1251,0	10	828,9	3	591,6	17	1680,3	6,4	1387,6
Desvio P.	0	6,2	0,7	37,4	0	0,0	0	0,0	0,4	23,2	0,5	42,1
Resultados para 100 indivíduos na população de AG_DT												
Instâncias	R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	20	1656,0	5	1294,5	10	828,9	3	591,6	17	1687,4	8	1387,0
2	20	1697,0	6	1265,9	10	828,9	3	591,6	16	1665,1	6	1438,9
3	20	1664,9	7	1265,8	10	828,9	3	591,6	17	1685,8	7	1424,5
4	19	1656,2	6	1234,0	10	828,9	3	591,6	17	1664,6	7	1433,1
5	19	1678,9	6	1252,7	10	828,9	3	591,6	17	1691,8	7	1344,5
Melhor DT	20	1656,0	7	1234,0	10	828,9	3	591,6	17	1664,6	7	1344,5
Média	20	1670,6	6	1262,6	10	828,9	3	591,6	17	1679,0	7	1405,6
Desvio P.	0,5	17,5	0,7	22,1	0	0,0	0	0,0	0,4	13,0	0,7	39,7

Os melhores resultados para cada instância, e as melhores médias estão em negrito. O desvio padrão foi calculado somente para informação e como critério de desempate. Os resultados da Tabela 6.1 mostram um melhor desempenho de AG_DT com 30 indivíduos para as instâncias R101 e R201. Para estas instâncias, uma população de 30 indivíduos resultou na melhor média e nos melhores resultados alcançados. Para as instâncias C101 e C201 todos as execuções chegaram ao mesmo valor. Para as instâncias RC101 e RC201 um número maior de indivíduos na população, 75, resultaram nos melhores resultados. A partir destes testes decidiu-se a utilização de 30 indivíduos para as classes R e C, e 75 indivíduos para a classe RC. Diferentes parâmetros para diferentes classes de problemas se justificam pela razão de que classes semelhantes podem ser encontradas no mundo real, dependendo do tipo de serviço de entrega e coleta utilizado. Uma avaliação de HGC_DT utilizando um conjunto de parâmetros únicos para todas as classes de Solomon pode ser encontrada em [AM2004].

Uma vez definido o número de indivíduos da população no melhor resultado da Tabela 6.1, o efeito da variação no número de indivíduos em cada torneio no processo de seleção do AG_DT foi avaliado. Foram testados 2, 3 e 4 indivíduos. Valores acima de 4 indivíduos, para uma população com as dimensões definidas previamente foram descartados, pois representam uma pressão seletiva muito forte, onde somente os melhores indivíduos teriam chance de gerar descendentes, ocasionando certamente uma convergência prematura do algoritmo. A Tabela 6.1 foi resultado da utilização de um torneio de 3 indivíduos, restando avaliar o algoritmo com 2 e 4 indivíduos. Os resultados dessa avaliação são apresentados na.

Tabela 6.2: Resultados de AG_DT com a variação do número de indivíduos participando no torneio.

Torneio M=3	30 Indivíduos								75 Indivíduos					
	Instâncias		R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	19	1656,5	6	1246,5	10	828,9	3	591,6	17	1690,7	6	1347,1		
2	20	1670,0	6	1253,9	10	828,9	3	591,6	17	1685,5	9	1313,8		
3	20	1670,1	6	1218,8	10	828,9	3	591,6	16	1651,2	9	1348,6		
4	20	1648,7	6	1210,4	10	828,9	3	591,6	17	1677,5	6	1410,5		
5	20	1646,8	6	1200,8	10	828,9	3	591,6	17	1688,5	7	1344,4		
Melhor DT	20	1646,8	7	1200,8	10	828,9	3	591,6	17	1651,2	7	1313,8		
Média	20	1658,4	6	1226,1	10	828,9	3	591,6	17	1678,7	7,4	1352,9		
Desvio Padrão	0,4	11,2	0	23,1	0	0,0	0	0,0	0,4	16,2	1,5	35,3		

Continua na próxima página

Continuação da página anterior												
Torneio M=2	30 Indivíduos								75 Indivíduos			
Instâncias	R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	20	1650,5	6	1305,5	10	828,9	3	591,6	17	1680,5	5	1407,3
2	20	1648,2	6	1321,3	10	828,9	3	591,6	16	1678,0	7	1400,8
3	20	1657,4	7	1265,4	10	828,9	3	591,6	16	1669,3	9	1349,6
4	20	1660,5	7	1269,8	10	828,9	3	591,6	17	1674,8	6	1442,4
5	20	1657,2	5	1272,7	10	828,9	3	591,6	17	1684,2	9	1367,7
Melhor DT	20	1648,2	7	1265,4	10	828,9	3	591,6	17	1669,3	7	1349,6
Média	20	1654,8	6,2	1287,0	10	828,9	3	591,6	17	1677,3	7,2	1393,6
Desvio Padrão	0	5,2	0,8	24,9	0	0,0	0	0,0	0,5	5,6	1,8	36,1
Torneio M=4	30 Indivíduos								75 Indivíduos			
Instâncias	R101		R201		C101		C201		RC101		RC201	
Execuções	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
1	21	1662,9	6	1256,1	10	828,9	3	591,6	17	1703,5	6	1416,7
2	20	1652,4	5	1296,9	10	828,9	3	591,6	17	1692,8	6	1416,0
3	20	1652,8	5	1303,4	10	828,9	3	591,6	17	1714,0	6	1440,0
4	20	1651,2	6	1287,9	10	828,9	3	591,6	17	1670,5	7	1390,6
5	19	1670,6	6	1215,4	10	828,9	3	591,6	17	1700,5	7	1437,6
Melhor DT	20	1651,2	7	1215,4	10	828,9	3	591,6	17	1670,5	7	1390,6
Média	20	1658,0	5,6	1272,0	10	828,9	3	591,6	17	1696,2	6,4	1420,2
Desvio Padrão	0,7	8,5	0,5	36,5	0	0,0	0	0,0	0	16,3	0,5	20,0

Otimizando as Taxas de Mutação do AG_DT

Conforme foi descrito no capítulo 3, um número relativamente grande de diferentes operadores, que atuarão como mutações nos indivíduos solução do AG_DT, foram propostos a fim de possibilitar uma ampla movimentação no espaço de busca de soluções viáveis do PRVJT. Entretanto, a correta calibração da taxa de atuação de cada operador não é uma tarefa simples. Um ajuste empírico de tais parâmetros geralmente é realizado pelos trabalhos utilizando algoritmos genéticos na literatura. Para o AG_DT que utiliza 13 diferentes operadores, fica muito difícil encontrar manualmente as taxas ideais de utilização de cada um. Por isso, um novo AG foi proposto para encontrar valores otimizados para tais parâmetros, chamado AG_PARAM. Uma vez definido o número de indivíduos, a execução dos operadores sobre a população foi feita da forma apresentada no Algoritmo 6.1 abaixo:

Procedimento Mutação;**Início**

```

// Mutação de Eliminação de Rota (M_ER)
para n:= 1 até Param_M_ER   faça Operador_M_ER(random(NumIndividuos));

// Mutação de Retirada de Consumidor (M_RC)
para n:= 1 até Param_M_RC   faça Operador_M_RC (random(NumIndividuos));

// Mutação de Inserção de Consumidor (M_IC)
para n:= 1 até Param_M_IC   faça Operador_M_IC (random(NumIndividuos));

// Mutação por Troca Simples de Consumidores em Rotas Distintas (M_TSC)
para n:= 1 até Param_M_TSC  faça Operador_M_TSC (random(NumIndividuos));

// Mutação por Troca de Consumidores em Rotas Distintas (M_TC)
para n:= 1 até Param_M_TC   faça Operador_M_TC (random(NumIndividuos));

//Mutação  $\lambda$ -interchange2 (M_INT2) – Exige ganho em DT p/ concluir operação
para n:= 1 até Param_M_INT2  faça Operador_M_INT2 (random(NumIndividuos));

// Mutação por Inversão de Consumidores (M_INVC)
para n:= 1 até Param_M_INVC  faça Operador_M_INVC (random(NumIndividuos));

// Mutação PFIH parcial (M_PFIHP)
para n:= 1 até ParamM_PFIHP  faça Operador_M_PFIHP (random(NumIndividuos));

// Mutação  $\lambda$ -interchange (M_INT) – Realiza o operador independente de ganho em DT
para n:= 1 até Param_M_INT   faça Operador_M_INT (random(NumIndividuos));

// Mutação por Reinserção de 1(um) Consumidor com ganho em DT (M_RDT)
para n:= 1 até Param_M_RDT   faça Operador_M_RDT (random(NumIndividuos));

//Mutação de Retirada e Troca de Consumidores seguida do PFIH (M_TCPFIH)
para n:= 1 até Param_M_TCPFIH  faça Operador_M_TCPFIH (random(NumIndividuos));

// Mutação por Reinserção de N Consumidores com ganho em DT (M_RDT2)
para n:= 1 até Param_M_RDT2  faça Operador_M_RDT2 (random(NumIndividuos));

//Mutação por Troca Simples de Consumidores com Redução de DT (M_TSCDT)
para n:= 1 até Param_M_TSCDT  faça Operador_M_TSCDT (random(NumIndividuos));

//Mutação por Troca de Consumidores com Redução da Distância Total Percorrida (M_TCDT)
para n:= 1 até Param_M_TCDT   faça Operador_M_TCDT (random(NumIndividuos));

//Mutação de Busca Exaustiva por Troca com Ganho de Custo (M_BETGC)
para n:= 1 até Param_M_BETGC  faça Operador_M_BETGC (random(NumIndividuos));

```

Fim.

Algoritmo 6.1: Procedimento que realiza as mutações sobre os indivíduos de uma população de NumIndividuos no AG_DT.

No Algoritmo 6.1 é possível observar que os operadores são executados em uma ordem sequencial, porém os indivíduos são sorteados aleatoriamente. A ordem definida diz respeito ao tipo de operador, inicia-se com aqueles operadores randômicos, que diversificam a busca, podendo piorar a solução, terminando com os operadores de intensificação.

O número de execuções de cada operador sobre uma população de 30 ou 75 indivíduos, conforme a classe de problema, será otimizado por AG_PARAM. Os indivíduos em AG_PARAM são uma seqüência de 15 números inteiros que podem variar de 0 a 255 (1 byte), cada qual correspondendo ao número de execução de cada mutação (parâmetros mostrados no Algoritmo 6.1). A avaliação dos indivíduos de GA_PARAM é feita otimizando uma determinada instância do PRVJT utilizando AG_DT, mas fazendo uso dos parâmetros de mutação do indivíduo de AG_PARAM. Antes de atribuir uma adaptabilidade ao indivíduo, uma determinada instância do PRVJT, sempre a mesma, é executada 5 vezes. A adaptabilidade do indivíduo será dada pela equação (6.2), onde DT_1 , DT_2 , ... , DT_5 correspondem aos 5 diferentes resultados de AG_DT em cada execução. DT_{min} e DT_{max} representam o melhor e o pior valor respectivamente em distância percorrida. Desta forma a adaptabilidade será a soma dos resultados, retirando o pior deles e somando-se duas vezes o melhor, dando mais chance na seleção aos indivíduos que produziram os melhores resultados, mesmo que uma dessas execuções não tenha produzido um bom resultado.

$$Adap = DT_{min} + DT_1 + DT_2 + DT_3 + DT_4 + DT_5 - DT_{max} \quad (6.2)$$

Foram encontrados os parâmetros otimizados utilizando AG_PARAM para cada classe específica R, C e RC. Somente uma instância escolhida aleatoriamente de cada classe foi usada em AG_PARAM. Para detalhes do cruzamento, mutação, seleção e outros do algoritmo AG_PARAM, ver Anexo B.

A abaixo mostra os valores encontrados pelo AG_PARAM para cada classe de problema de Solomon.

Tabela 6.3 Utilização dos operadores de mutação sobre uma população de 30 (R,C) e 75 (RC) indivíduos. Valores otimizados encontrados pelo algoritmo AG_PARAM.

Operadores de mutação	Classes de Solomon			PRVC
	R	C	RC	A, B, E, M e P
<i>M_ER</i>	2	1	1	2
<i>M_RC</i>	3	0	1	1
<i>M_IC</i>	1	4	4	1
<i>M_TSC</i>	7	5	4	2
<i>M_TC</i>	3	3	11	3
<i>M_INT2</i>	0	0	5	0
<i>M_INVC</i>	11	5	0	3
<i>M_PFIHP</i>	4	4	3	1
<i>M_INT</i>	1	2	2	2
<i>M_RDT</i>	6	6	7	4
<i>M_TCPFIH</i>	0	4	3	0
<i>M_RDT2</i>	1	3	4	3
<i>M_TSCDT</i>	4	4	2	3
<i>M_TCDT</i>	0	3	1	3
<i>M_BETGC</i>	1	3	2	3

Os resultados da Tabela 6.3 não são muito conclusivos quanto a importância de cada tipo de operador na qualidade dos resultados. O resultado mais expressivo foi a pouca importância atribuída ao operador λ -interchange (parâmetro *M_INT2*) onde houve exigência de ganho em DT para que a troca de consumidores seja realizada. Embora este operador seja um dos mais utilizados na literatura, mostrou-se vantajoso somente para as instâncias das classes mistas RC1 e RC2.

Por fim AG_DT foi avaliado utilizando-se o processo de seleção por roleta, descrito no capítulo 3. Os resultados por torneio foram expressivamente superiores, talvez pelo fato de todos os ajustes anteriores terem utilizado este procedimento. Devido aos primeiros resultados tão distantes não foram realizados testes mais abrangentes.

6.2.2 Resultados

Utilizando as instâncias de Solomon, os primeiros testes foram realizados nas mesmas condições utilizadas na literatura pelos métodos exatos no tratamento do PRVJT, ou seja, os valores das coordenadas XY, dos limites da janela de tempo e do tempo de serviço são

multiplicados por 10 antes do início dos cálculos. A partir de então todas as operações são realizadas seguidas de truncamento. As operações de truncamento são limitadas ao cálculo da matriz de adjacência, já que durante a busca em si, são realizadas somente operações de soma e comparação entre valores já inteiros. Nada é necessário fazer com a demanda dos consumidores ou a limitação de capacidade dos veículos, pois estes valores já são inteiros. A função objetivo foi somente a distância total percorrida (DT), sem considerar o número de veículos (NV).

Os testes foram realizados executando HGC_DT durante 60 minutos, 2 minutos para cada evolução de AG_DT tratando o problema completo e 1 minuto e 12 segundos (60% do intervalo anterior) tratando o problema reduzido. Foram realizadas 3 execuções para cada instância de Solomon em um computador Pentium IV, 2.4 Mhz e 512 Mbytes de RAM, produzindo os resultados da Tabela 6.4. Os parâmetros foram configurados conforme os melhores resultados dos testes descritos e a otimização das taxas de mutação realizada por AG_PARAM. Foram usados diferentes parâmetros para cada uma das classes R, C, RC, conforme Tabela 6.3.

A Tabela 6.5 mostra todos os problemas de Solomon com 100 consumidores e os resultados ótimos já encontrados para minimização da distância total percorrida. Os valores de número de veículos utilizados na solução também são apresentados como referência, não sendo, porém objetivo da otimização. Os melhores resultados obtidos com HGC_DT das 3 execuções são também apresentados nesta mesma tabela, possibilitando uma comparação inédita dos resultados dos métodos exatos com uma heurística. Embora todos os métodos utilizados nos resultados da Tabela 6.5 tenham trabalhado com inteiros, estes são apresentados novamente divididos por 10 (por isso uma casa decimal) para corresponder à distância original dos consumidores. Os valores em **negrito** correspondem àqueles onde o ótimo foi alcançado. As células em branco indicam as instâncias onde o problema ainda não foi resolvido pelos métodos exatos, portanto o valor mínimo global ainda é desconhecido.

Tabela 6.4: Resultados de 3 execuções de HGC_DT utilizando inteiros durante o processamento. Os valores foram divididos por 10 posteriormente para refletir novamente as distâncias reais. O tempo de execução foi de 60 minutos, utilizando um computador utilizado Pentium IV, 2.4 MHz, 512 de RAM.

Instância	1a Execução		2a Execução		3a Execução		Média		Desvio Padrão	
	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
C101	10	827,3	10	827,3	10	827,3	10	827,3	0	0,0
C102	10	827,3	10	827,3	10	827,3	10	827,3	0	0,0
C103	10	826,3	10	826,3	10	826,3	10	826,3	0	0,0
C104	10	822,9	10	822,9	10	822,9	10	822,9	0	0,0
C105	10	827,3	10	827,3	10	827,3	10	827,3	0	0,0
C106	10	827,3	10	827,3	10	827,3	10	827,3	0	0,0
C107	10	827,3	10	827,3	10	827,3	10	827,3	0	0,0
C108	10	827,3	10	827,3	10	827,3	10	827,3	0	0,0
C109	10	827,3	10	827,3	10	827,3	10	827,3	0	0,0
C201	3	589,1	3	589,1	3	589,1	3	589,1	0	0,0
C202	3	589,1	3	589,1	3	589,1	3	589,1	0	0,0
C203	3	588,7	3	588,7	3	588,7	3	588,7	0	0,0
C204	3	617,0	3	590,6	3	590,6	3	599,4	0	15,2
C205	3	586,4	3	586,4	3	586,4	3	586,4	0	0,0
C206	3	586,0	3	586,0	3	586,0	3	586,0	0	0,0
C207	3	585,8	3	585,8	3	585,8	3	585,8	0	0,0
C208	3	585,8	3	585,8	3	585,8	3	585,8	0	0,0
R101	20	1637,7	20	1637,7	20	1637,7	20	1637,7	0	0,0
R102	18	1466,6	18	1466,6	18	1466,6	18	1466,6	0	0,0
R103	14	1212,2	14	1210,2	14	1211,2	14	1211,2	0	1,0
R104	11	979,3	11	989,2	11	980,8	11	983,1	0	5,3
R105	15	1355,8	15	1360,1	13	1355,3	14	1357,1	1	2,6
R106	13	1235,7	13	1234,6	13	1235,8	13	1235,4	0	0,7
R107	11	1068,7	11	1071,0	11	1067,3	11	1069,0	0	1,9
R108	10	947,2	10	948,4	10	946,2	10	947,3	0	1,1
R109	13	1152,4	13	1150,9	13	1146,9	13	1150,1	0	2,8
R110	12	1079,4	12	1081,7	12	1083,5	12	1081,5	0	2,1
R111	12	1063,0	12	1048,7	12	1053,4	12	1055,0	0	7,3
R112	10	962,9	10	962,3	11	977,4	10	967,5	1	8,6
R201	7	1175,8	8	1165,3	7	1177,2	7	1172,8	1	6,5
R202	6	1053,9	6	1067,0	6	1072,4	6	1064,4	0	9,5
R203	5	910,1	5	924,4	6	924,3	5	919,6	1	8,2
R204	5	802,3	5	798,7	4	770,8	5	790,6	1	17,2
R205	5	972,5	6	986,8	5	1001,0	5	986,8	1	14,3
R206	5	934,4	5	945,5	5	952,2	5	944,0	0	9,0
R207	5	863,1	4	860,1	5	881,7	5	868,3	1	11,7
R208	4	744,1	4	774,1	4	750,1	4	756,1	0	15,9
R209	5	888,5	5	911,8	5	909,0	5	903,1	0	12,7
R210	5	952,2	5	960,0	5	954,7	5	955,6	0	4,0
R211	5	815,5	5	815,5	4	833,6	5	821,5	1	10,5
RC101	16	1635,7	15	1623,5	16	1626,1	16	1628,4	1	6,4
RC102	14	1458,2	14	1473,5	14	1473,5	14	1468,4	0	8,8
RC103	12	1300,9	12	1273,6	12	1272,5	12	1282,3	0	16,1
RC104	11	1145,0	10	1142,6	10	1137,6	10	1141,7	1	3,8
RC105	15	1513,7	15	1513,7	15	1528,9	15	1518,8	0	8,8
RC106	13	1373,9	13	1373,9	13	1374,8	13	1374,2	0	0,5
RC107	12	1230,8	12	1223,8	12	1209,3	12	1221,3	0	11,0
RC108	11	1131,5	11	1115,2	11	1126,2	11	1124,3	0	8,3
RC201	8	1274,3	8	1286,4	8	1282,4	8	1281,0	0	6,2
RC202	7	1119,5	8	1140,8	7	1159,2	7	1139,8	1	19,9
RC203	5	961,1	5	958,0	5	959,5	5	959,5	0	1,6
RC204	4	832,7	4	807,6	5	826,2	4	822,2	1	13,0
RC205	4	1160,0	7	1169,2	7	1154,0	6	1161,1	2	7,7
RC206	5	1081,3	6	1092,2	6	1080,4	6	1084,6	1	6,6
RC207	7	1005,6	6	1018,2	6	1023,9	6	1015,9	1	9,4
RC208	6	834,4	5	820,5	6	836,7	6	830,5	1	8,8

Tabela 6.5: Comparação dos resultados da HGC_DT com os ótimos conhecidos na literatura. Resultados em negrito significam que o ótimo global foi alcançado.

Instância	Ótimo Conhecido		Melhor HGC_DT		Média HGC_DT		Instância	Ótimo Conhecido		Melhor HGC_DT		Média HGC_DT	
	NV	DT	NV	DT	NV	DT		NV	DT	NV	DT	NV	DT
C101	10	827,3	10	827,3	10	827,3	R201	8	1143,2	8	1165,3	7	1172,8
C102	10	827,3	10	827,3	10	827,3	R202			6	1053,9	6	1064,4
C103	10	826,3	10	826,3	10	826,3	R203			5	910,1	5	919,6
C104	10	822,9	10	822,9	10	822,9	R204			4	770,8	5	790,6
C105	10	827,3	10	827,3	10	827,3	R205			5	972,5	5	986,8
C106	10	827,3	10	827,3	10	827,3	R206			5	934,4	5	944,0
C107	10	827,3	10	827,3	10	827,3	R207			4	860,1	5	868,3
C108	10	827,3	10	827,3	10	827,3	R208			4	744,1	4	756,1
C109	10	827,3	10	827,3	10	827,3	R209			5	888,5	5	903,1
C201	3	589,1	3	589,1	3	589,1	R210			5	952,2	5	955,6
C202	3	589,1	3	589,1	3	589,1	R211			5	815,5	5	821,5
C203	3	588,7	3	588,7	3	588,7	RC101	15	1619,8	15	1623,5	16	1628,4
C204			3	590,6	3	599,4	RC102	14	1457,4	14	1458,2	14	1468,4
C205	3	586,4	3	586,4	3	586,4	RC103	11	1258,0	12	1272,5	12	1282,3
C206	3	586,0	3	586,0	3	586,0	RC104			10	1137,6	10	1141,7
C207	3	585,8	3	585,8	3	585,8	RC105	15	1513,7	15	1513,7	15	1518,8
C208	3	585,8	3	585,8	3	585,8	RC106			13	1373,9	13	1374,2
R101	20	1637,7	20	1637,7	20	1637,7	RC107			12	1209,3	12	1221,3
R102	18	1466,6	18	1466,6	18	1466,6	RC108			11	1115,2	11	1124,3
R103	14	1208,7	14	1210,2	14	1211,2	RC201	9	1261,8	8	1274,3	8	1281,0
R104			11	979,3	11	983,1	RC202	8	1092,3	7	1119,5	7	1139,8
R105	15	1355,3	13	1355,3	14	1357,1	RC203			5	958,0	5	959,5
R106	13	1234,6	13	1234,6	13	1235,4	RC204			4	807,6	4	822,2
R107	11	1064,6	11	1067,3	11	1069,0	RC205	7	1154,0	7	1154,0	6	1161,1
R108			10	946,2	10	947,3	RC206			6	1080,4	6	1084,6
R109	13	1146,9	13	1146,9	13	1150,1	RC207			7	1005,6	6	1015,9
R110	12	1068,0	12	1079,4	12	1081,5	RC208			5	820,5	6	830,5
R111	12	1048,7	12	1048,7	12	1055,0							
R112			10	962,3	10	967,5							

Os resultados para as classes C1 são todos constantes para as três execuções, mostrando que estes problemas não são um grande desafio também para HGC_DT, como tem ocorrido com outros trabalhos na literatura. Para a classe C2, somente a instância C204 não tem o resultado ótimo conhecido. Nesta mesma instância, singularmente na classe C2 ocorreu uma diferença entre a média e o melhor resultado de HGC_DT. Na classe R1, os resultados também foram muito positivos, 6 ótimos dos 9 conhecidos foram alcançados (66.6%). Os melhores resultados onde o ótimo não foi alcançado na classe R1 ficaram somente 0.47% acima em média. Os melhores resultados e os resultados na média de HGC_DT em R1 ficaram 0.29% e 0.13% acima dos ótimos conhecidos. Mostrando que a estratégia de HGC_DT é muito competitiva para esta classe de problemas. Nas classes R2 somente um resultado ótimo é conhecido na literatura e não foi alcançado por HGC_DT, ficando 1.93% acima. Em RC2 três resultados ótimos são conhecidos, sendo que um foi alcançado pelo melhor resultado de HGC_DT (0.33%). Nos outros dois os resultados ficaram 1.69% acima. Para a classe RC, dos quatro ótimos conhecidos somente um foi alcançado, ficando os demais 0.44% acima. Os resultados ótimos conhecidos e a

referência bibliográfica respectiva são mantidos atualizados em <http://web.cba.neu.edu/~msolomon/problems.htm>.

Em termos globais, 72.7% (24/33) dos casos onde existe uma solução ótima conhecida estes ótimos foram alcançados em pelo menos uma das execuções de HGC_DT. Nos nove casos onde o ótimo não foi alcançado, os melhores resultados ficaram a 0.86% acima em DT. Os valores de DT na média geral foi 0.29% acima dos ótimos conhecidos. Os resultados apresentados e ainda o fato do tempo na execução de HGC_DT ser bem inferior àqueles utilizados pelos métodos exatos, comprovam a competitividade de HGC_DT.

Como um caso particular do PRVJT, o problema sem restrição de tempo, isto é, o Problema de Roteamento de Veículos Capacitado (PRVC) também tem sido muito tratado na literatura por métodos exatos. HGC_DT foi avaliada para um conjunto de instâncias, onde os resultados já estão muito bem trabalhados. A Tabela 6.6 apresenta o melhor resultado de HGC_DT após 3 execuções de 30 minutos. Novamente os resultados foram muito competitivos.

Tabela 6.6: Comparação dos resultados da HGC_DT com os ótimos conhecidos para 44 instâncias da literatura do PRVC. Foi utilizada operação com inteiros após o arredondamento das distâncias para o inteiro mais próximo.

Instância	Ótimo Conhecido		Melhor HGC DT		Média HGC DT		Instância	Ótimo Conhecido		Melhor HGC DT		Média HGC DT	
	NV	DT	NV	DT	NV	DT		NV	DT	NV	DT	NV	DT
A-n53-K7	7	1010	7	1010	7,0	1014,7	E-n51-k5	5	521	5	521	5,0	521,0
A-n54-k7	7	1167	7	1167	7,0	1167,0	E-n76-k7	7	682	7	688	7,0	690,3
A-n55-k9	9	1073	9	1073	9,0	1073,3	E-n76-k8	8	735	8	738	8,0	743,0
A-n60-k9	9	1354	9	1354	9,0	1356,7	E-n76-k10	10	830	11	832	11,0	838,7
A-n61-k9	9	1034	9	1035	9,7	1035,0	E-n76-k14	14	1021	14	1023	14,0	1023,5
A-n62-k8	8	1288	8	1291	8,0	1301,3	E-n101-k8	8	815	8	836	8,0	838,5
A-n63-k9	9	1616	9	1623	9,0	1626,3	E-n101-k14	14	1071	14	1102	14,0	1105,0
A-n63-k10	10	1314	10	1314	10,0	1316,5	M-n101-k10	10	820	10	821	10,0	822,0
A-n64-k9	9	1401	10	1411	9,3	1412,0	M-n121-k7	7	1034	8	1081	8,0	1084,3
A-n65-k9	9	1174	9	1177	9,0	1177,0	M-n151-k12	12	1053	12	1067	12,0	1074,0
A-n80-k10	10	1763	-	-	-	-	M-n200-k16			17	1390	17,0	1404,0
B-n50-K7	7	741	7	741	7,0	741,0	M-n200-k17	17	1373	17	1394	17,7	1411,0
B-n50-K8	8	1312	8	1313	8,0	1313,0	P-n50-k8	8	649	9	629	9,0	629,7
B-n51-K7	7	1032	8	1016	8,0	1016,0	P-n50-K10	10	696	10	697	10,0	697,0
B-n52-K7	7	747	7	747	7,0	747,0	P-n55-k15	15	993	16	941	16,0	943,0
B-n56-k7	7	707	7	707	7,0	707,0	P-n60-k10	10	744	10	744	10,0	744,5
B-n57-k7	7	1153	8	1140	8,0	1140,0	P-n60-k15	15	968	15	968	15,0	968,0
B-n57-k9	9	1598	9	1598	9,0	1598,0	P-n65-k10	10	792	10	792	10,0	794,0
B-n63-k10	10	1496	9	1606	9	1606,0	P-n70-k10	10	827	10	828	10,0	832,0
B-n64-k9	9	861	9	864	9	864,0							
B-n66-k9	9	1316	9	1322	9	1322,0							
B-n67-k10	10	1032	10	1035	10	1035,0							
B-n68-k9	9	1275	9	1288	9	1288,0							
B-n78-k10	10	1221	10	1226	10	1226,0							

Os resultados em negritos na Tabela 6.6 mostram que em 14 problemas o resultado ótimo foi encontrado, ou superado utilizando um número de veículos maior. Em algumas dessas instâncias os valores em DT são inferiores por utilizarem um número de veículos maior, pois HGC_DT não fixa o número de veículos. As colunas com “-” representam problemas onde não a heurística proposta não chegou a ser avaliada. No geral, os resultados ficaram 0.50% acima dos valores ótimos conhecidos. Se retirarmos as instâncias com diferentes número de veículos, onde HGC_DT obtém um valor de DT inferior, a diferença sobre para 0.56%. Considerando o tempo de 1 hora de execução da heurística, comparado ao tempo geralmente muito superior dos métodos exatos, HGC_DT pode ser uma opção competitiva. Adicionalmente, os valores médios estão próximos dos melhores obtidos, somente 0.19% acima, indicando uma boa robustez.

Tabela 6.7 Resultados de 3 execuções de HGC_DT utilizando dupla precisão. O tempo de execução foi de 60 minutos, utilizando um computador utilizado Pentium IV, 2.4 MHz, 512 de RAM.

Instância	1a Execução		2a Execução		3a Execução		Média		Desvio Padrão	
	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
C101	10	828,937	10	828,937	10	828,937	10	828,9	0	0,0
C102	10	828,937	10	828,937	10	828,937	10	828,9	0	0,0
C103	10	828,065	10	828,065	10	828,065	10	828,1	0	0,0
C104	10	824,770	10	824,770	10	824,770	10	824,8	0	0,0
C105	10	828,937	10	828,937	10	828,937	10	828,9	0	0,0
C106	10	828,937	10	828,937	10	828,937	10	828,9	0	0,0
C107	10	828,937	10	828,937	10	828,937	10	828,9	0	0,0
C108	10	828,937	10	828,937	10	828,937	10	828,9	0	0,0
C109	10	828,937	10	828,937	10	828,937	10	828,9	0	0,0
C201	3	591,557	3	591,557	3	591,557	3	591,6	0	0,0
C202	3	591,557	3	591,557	3	591,557	3	591,6	0	0,0
C203	3	591,173	3	591,173	3	591,173	3	591,2	0	0,0
C204	3	590,599	3	597,129	3	600,023	3	595,9	0	4,8
C205	3	588,876	3	588,876	3	588,876	3	588,9	0	0,0
C206	3	588,493	3	588,493	3	588,493	3	588,5	0	0,0
C207	3	588,286	3	588,286	3	588,286	3	588,3	0	0,0
C208	3	588,324	3	588,324	3	588,324	3	588,3	0	0,0
R101	20	1642,877	20	1642,870	20	1643,791	20	1643,2	0	0,5
R102	18	1472,815	18	1472,620	18	1472,968	18	1472,8	0	0,2
R103	14	1220,210	14	1214,701	14	1213,620	14	1216,2	0	3,5
R104	11	990,070	11	991,609	11	986,096	11	989,3	0	2,8
R105	15	1371,561	15	1360,783	15	1361,490	15	1364,6	0	6,0
R106	13	1243,293	13	1241,518	13	1247,341	13	1244,1	0	3,0
R107	11	1081,135	11	1076,125	11	1078,629	11	1078,6	0	2,5
R108	10	948,573	10	950,973	10	953,833	10	951,1	0	2,6
R109	13	1154,154	13	1151,839	13	1153,060	13	1153,0	0	1,2
R110	12	1093,320	12	1092,347	12	1094,476	12	1093,4	0	1,1
R111	12	1054,556	12	1053,496	12	1064,466	12	1057,5	0	6,1
R112	10	973,108	10	963,476	10	960,675	10	965,8	0	6,5
R201	8	1150,736	9	1148,483	8	1148,717	8	1149,3	1	1,2
R202	7	1049,737	8	1059,598	7	1055,997	7	1055,1	1	5,0
R203	5	900,080	7	906,474	5	939,061	6	915,2	1	20,9
R204	4	772,330	4	783,168	4	788,780	4	781,4	0	8,4
R205	8	989,869	6	970,886	5	972,420	6	977,7	2	10,5
R206	5	898,914	6	912,452	4	902,706	5	904,7	1	7,0
R207	3	853,700	4	834,930	3	851,428	3	846,7	1	10,2
R208	4	737,229	3	739,601	3	723,610	3	733,5	1	8,6

Continua na próxima página

Continuação da página anterior										
	1a Execução		2a Execução		3a Execução		Média		Desvio Padrão	
R209	6	880,809	6	879,531	6	895,829	6	885,4	0	9,1
R210	7	932,887	7	936,170	6	934,208	7	934,4	1	1,7
R211	6	809,574	5	787,511	5	810,952	5	802,7	1	13,2
Instância	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
RC101	16	1652,685	16	1639,968	16	1646,532	16	1646,4	0	6,4
RC102	14	1466,840	15	1481,100	14	1479,338	14	1475,8	1	7,8
RC103	12	1277,666	11	1264,707	12	1277,078	12	1273,2	1	7,3
RC104	10	1135,520	10	1153,630	11	1151,948	10	1147,0	1	10,0
RC105	16	1557,994	16	1518,600	16	1528,180	16	1534,9	0	20,5
RC106	14	1395,635	13	1378,454	13	1377,352	13	1383,8	1	10,3
RC107	12	1214,186	12	1212,830	12	1229,955	12	1219,0	0	9,5
RC108	11	1123,860	11	1134,000	11	1117,526	11	1125,1	0	8,3
RC201	10	1278,354	9	1280,956	9	1274,537	9	1277,9	1	3,2
RC202	9	1125,762	9	1119,271	8	1113,526	9	1119,5	1	6,1
RC203	5	945,960	5	972,296	6	970,042	5	962,8	1	14,6
RC204	4	799,670	4	809,078	4	817,855	4	808,9	0	9,1
RC205	7	1161,810	8	1162,331	8	1166,193	8	1163,4	1	2,4
RC206	8	1088,538	7	1081,074	7	1059,886	7	1076,5	1	14,9
RC207	6	991,230	7	976,396	6	994,281	6	987,3	1	9,6
RC208	5	795,391	5	801,821	6	847,550	5	814,9	1	28,4

Na segunda seqüência de testes, continuando como único objetivo a distância total percorrida, os cálculos foram feitos utilizando dupla precisão. Novamente foram HGC_DT foi executada 3 vezes para cada instância de Solomon. Os resultados podem ser vistos na Tabela 6.8.

Desta vez os resultados foram comparados com aqueles produzidos por heurísticas, que fazem uso dos mesmos critérios. Como o esforço na literatura para minimizar DT tem sido bem menor, alguns dos melhores resultados em DT foram produzidos por artigos onde NV foi o objetivo principal e DT apenas o segundo. Na Tabela 6.8 estão os melhores resultados anteriores em DT e os melhores de HGC_DT dentro de 3 execuções. Os melhores resultados anteriores foram encontrados em [RGP99] [TBG+97] e [TLZ+01] que utilizaram DT como primeiro objetivo e outros onde NV foi a prioridade, embora também tenham obtido os melhores resultados em DT. Os resultados destes últimos são mantidos atualizados em <http://www.sintef.no/static/am/opti/projects/top/vrp/benchmarks.html>.

Tabela 6.8 Comparação dos melhores resultados de HGC_DT com os melhores resultados na literatura, produzidos por heurísticas e cálculos com dupla precisão. Os valores em negrito correspondem a soluções atingidas ou superadas por HGC_DT.

Instância	Melhor Anterior		Melhor HGC_DT		Instância	Melhor Anterior		Melhor HGC_DT	
	NV	DT	NV	DT		NV	DT	NV	DT
C101	10	828,9	10	828,937	R201	4	1252,4	9	1148,483
C102	10	828,9	10	828,937	R202	3	1191,7	7	1049,737
C103	10	828,1	10	828,065	R203	3	939,5	5	900,080
C104	10	824,8	10	824,770	R204	2	825,5	4	772,330
C105	10	828,9	10	828,937	R205	3	994,4	6	970,886
C106	10	828,9	10	828,937	R206	3	906,1	5	898,914
C107	10	828,9	10	828,937	R207	2	893,3	4	834,930
C108	10	828,9	10	828,937	R208	2	726,8	3	723,610
C109	10	828,9	10	828,937	R209	3	909,2	6	879,531
C201	3	591,6	3	591,557	R210	3	939,3	7	932,887
C202	3	591,6	3	591,557	R211	2	892,7	5	787,511
C203	3	591,2	3	591,173	RC101	14	1696,9	16	1639,968
C204	3	590,6	3	590,599	RC102	12	1554,8	14	1466,840
C205	3	588,9	3	588,876	RC103	11	1261,7	11	1264,707
C206	3	588,5	3	588,493	RC104	10	1135,5	10	1135,520
C207	3	588,3	3	588,286	RC105	13	1629,4	16	1518,600
C208	3	588,3	3	588,324	RC106	11	1424,7	13	1377,352
R101	19	1645,8	20	1642,870	RC107	11	1230,5	12	1212,830
R102	17	1486,1	18	1472,620	RC108	10	1139,8	11	1117,526
R103	13	1292,7	14	1213,620	RC201	4	1406,9	9	1274,537
R104	9	1007,2	11	986,096	RC202	3	1365,7	8	1113,526
R105	14	1377,1	15	1360,783	RC203	3	1049,6	5	945,960
R106	12	1252,0	13	1241,518	RC204	3	798,4	4	799,670
R107	10	1104,7	11	1076,125	RC205	4	1297,2	7	1161,810
R108	9	960,9	10	948,573	RC206	3	1146,3	7	1059,886
R109	11	1194,7	13	1151,839	RC207	3	1061,1	7	976,396
R110	10	1118,6	12	1092,347	RC208	3	828,1	5	795,391
R111	10	1096,7	12	1053,496					
R112	9	982,1	10	960,675					

Tabela 6.9: Comparação dos resultados da HGC_DT com os melhores resultados na literatura, em termos da média por classe de problema. Valores negativos correspondem a uma redução percentual da distância percorrida.

Ganho médio em DT			
Instância	DT (%)	Instância	DT (%)
R1	-2,19	R2	-5,46
C1	0,00	C2	0,00
RC1	-3,07	RC2	-9,23
REDUÇÃO GLOBAL (DT)		-5,28%	

Os resultados produzidos pela HGC_DT superaram ou alcançaram 53 dos 56 problemas existentes, em termos de distância total percorrida. Os melhores resultados foram obtidos nos conjuntos R2 e RC2, reduzindo em 5.46% e 9.23% a distância total percorrida respectivamente. Tanto em C1 quanto em C2 os resultados foram exatamente os mesmos, indicando que podem ser os ótimos globais, visto que as rotas são as mesmas dos resultados ótimos encontrados com números inteiros.

A Tabela 6.10 apresenta os melhores resultados encontrados individualmente pelo AG_DT e os melhores encontrados no problema principal (PPC) de HGC_DT. É possível observar

que há uma significativa melhora dos resultados, indicando o grande mérito da heurística de geração de colunas. Conseqüentemente os resultados não são apenas consequência de várias tentativas de AG_DT, como em uma estratégia do tipo *restart*. Somente nas instâncias C1 e C2 não houve melhora nos resultados na execução do problema principal sobre o conjunto global de rotas R_{GLOBAL} , indicando que as execuções individuais de AG_DT já atingiram os resultados finais, que parecem ser os mínimos globais. As reduções em DT médias na solução do PPC foram de 1.2% para a classe R1, 3.18% para a classe R2, 2.41%, para RC1 e também 2.41% para RC2.

Tabela 6.10 : Comparação do melhor resultado de um AG_DT individual com o resultado do problema principal (PPC) na mesma execução de HGC_DT.

	Melhor AG_DT		Resultado Final PPC		Redução			Melhor AG_DT		Resultado Final PPC		Redução	
	NV	DT	NV	DT	NV	DT (%)		NV	DT	NV	DT	NV	DT (%)
C101	10	828,937	10	828,937	0	0,00%	R201	9	1179,790	9	1148,483	0	-2,65%
C102	10	828,937	10	828,937	0	0,00%	R202	8	1071,070	7	1049,737	-1	-1,99%
C103	10	828,065	10	828,065	0	0,00%	R203	7	932,760	5	900,080	-2	-3,50%
C104	10	824,770	10	824,770	0	0,00%	R204	4	807,380	4	772,330	0	-4,34%
C105	10	828,937	10	828,937	0	0,00%	R205	6	1005,340	6	970,886	0	-3,43%
C106	10	828,937	10	828,937	0	0,00%	R206	6	935,240	5	898,914	-1	-3,88%
C107	10	828,937	10	828,937	0	0,00%	R207	3	875,300	4	834,930	1	-4,61%
C108	10	828,937	10	828,937	0	0,00%	R208	3	748,350	3	723,610	0	-3,31%
C109	10	828,937	10	828,937	0	0,00%	R209	6	900,650	6	879,531	0	-2,34%
C201	3	591,557	3	591,557	0	0,00%	R210	7	955,160	7	932,887	0	-2,33%
C202	3	591,557	3	591,557	0	0,00%	R211	4	808,560	5	787,511	1	-2,60%
C203	3	591,173	3	591,173	0	0,00%	RC101	16	1660,100	16	1639,968	0	-1,21%
C204	3	590,599	3	590,599	0	0,00%	RC102	14	1482,910	14	1466,840	0	-1,08%
C205	3	588,876	3	588,876	0	0,00%	RC103	12	1297,580	11	1264,707	-1	-2,53%
C206	3	588,493	3	588,493	0	0,00%	RC104	11	1174,620	10	1135,520	-1	-3,33%
C207	3	588,286	3	588,286	0	0,00%	RC105	15	1572,400	16	1518,600	1	-3,42%
C208	3	588,324	3	588,324	0	0,00%	RC106	14	1417,460	13	1377,352	-1	-2,83%
R101	20	1646,840	20	1642,870	0	-0,24%	RC107	13	1261,100	12	1212,830	-1	-3,83%
R102	18	1482,740	18	1472,620	0	-0,68%	RC108	11	1129,310	11	1117,526	0	-1,04%
R103	15	1230,040	14	1213,620	-1	-1,33%	RC201	9	1286,830	9	1274,537	0	-0,96%
R104	11	1009,540	11	986,096	0	-2,32%	RC202	9	1148,840	8	1113,526	-1	-3,07%
R105	14	1380,870	15	1360,783	0	-1,45%	RC203	6	1001,250	5	945,960	-1	-5,52%
R106	13	1249,400	13	1241,518	0	-0,63%	RC204	5	826,150	4	799,670	-1	-3,21%
R107	11	1079,730	11	1076,125	0	-0,33%	RC205	8	1168,220	7	1161,810	-1	-0,55%
R108	10	965,580	10	948,573	0	-1,76%	RC206	8	1084,300	7	1059,886	-1	-2,25%
R109	13	1156,260	13	1151,839	0	-0,38%	RC207	6	999,260	7	976,396	1	-2,29%
R110	12	1106,990	12	1092,347	0	-1,32%	RC208	5	806,870	5	795,391	0	-1,42%
R111	12	1056,870	12	1053,496	0	-0,32%							
R112	11	996,790	10	960,675	-1	-3,62%							

É possível observar também que a estratégia se adapta muito bem para problemas com muitos veículos, pois há maior possibilidade de combinação de rotas dos diferentes mínimos locais produzidos.

Conforme fluxograma de HGC_DT apresentado no diagrama da Figura 3.16, no fim de cada fase de transição entre o tratamento do problema completo e do problema reduzido, um PPC é resolvido pelo algoritmo de programação linear inteira utilizado, o GLPK.

Conforme pode ser visto na Tabela 6.10, o problema que apresenta o maior número de rotas é o R101 com 20 rotas. Isso significa que após a primeira fase, o número de execuções de AG_DT, 5, multiplicadas por 20 rotas cada (em média), representam 100 rotas geradas. Conseqüentemente o PPC da primeira fase trabalha com 100 colunas em média. Por isso, a solução foi instantânea, consumindo menos de um segundo.

Na fase do problema reduzido, a redução do número de rotas acontece em 70% (ROUTE_PERCENT = 30%), ou seja, o número médio de rotas para R101 é de 6 rotas. Por outro lado, 15 mínimos locais são produzidos, resultando no total de 90 rotas para o PPC. Estas rotas são adicionadas ao conjunto R_{GLOBAL} para o PPC final. Após o fim do tempo total disponível para HGC_DT, que é de 60 minutos, são produzidas $(100+90) \times 3 = 570$ rotas, que correspondem a 3 ciclos completos da heurística. Restando 3 minutos que representam 3 evoluções de AG_DT com o problema completo, gerando mais 60 rotas, resultado em 630 rotas. Conseqüentemente, no caso mais crítico o GLPK deve resolver um PPC com 630 variáveis inteiras. Os resultados mostram que este número de colunas não é um obstáculo. Os testes mostraram que nenhuma execução consumiu mais que um segundo para a resposta exata.

6.3 Conclusões

Os resultados obtidos comprovaram a robustez e eficiência da estratégia proposta de geração de colunas utilizando AG_DT. Embora a comparação com heurísticas minimizando distância percorrida não seja tão relevante, devido ao pequeno número de artigos dedicados a este objetivo, estabelece-se um novo conjunto de metas (*benchmarks*), no critério distância percorrida, para o PRVJT estático. Adicionalmente, a comparação dos resultados obtidos com aqueles produzidos por métodos exatos foi bastante relevante, permitindo estabelecer quantitativamente a distância que estes estão dos ótimos conhecidos. Um percentual expressivo de 72.7% (24/33) ótimos globais foi alcançado, comprovando a qualidade da proposta. A estratégia híbrida também proporcionou uma robustez significativa no comportamento de HGC_DT, onde os resultados médios e mínimos estão muito próximos.

Capítulo 7

Resultados Minimizando Número de Veículos no PRVJT Estático

Neste capítulo, a heurística HGC_NV proposta no capítulo 4 é avaliada no conjunto de problemas de Solomon para o PRVJT com a finalidade de minimizar prioritariamente o número de veículos. O segundo objetivo é definido como sendo a distância total percorrida. Isso possibilitou avaliar a heurística proposta com vários resultados da literatura, uma vez que para minimização do número de veículos existe um amplo conjunto de trabalhos publicados utilizando heurísticas. Os resultados mostram que a heurística proposta é muito competitiva quanto ao primeiro objetivo, estando entre as poucas que obtiveram o melhor resultado para todas as instâncias de Solomon. Conforme foi descrito no capítulo 4, a heurística HGC_NV incorpora HGC_DT, sendo a última utilizada para minimizar a distância percorrida, uma vez minimizado o número de veículos pelo AG_NV, também apresentado no capítulo 4. Neste capítulo, a seção 7.1 apresenta como foram definidos os parâmetros de AG_NV e HGC_NV. A seção 7.2 apresenta os resultados e a seção 7.3 as conclusões obtidas.

7.1 Configurando os parâmetros em AG_NV e HGC_NV

Embora o algoritmo AG_PARAM pudesse ser adaptado para encontrar os melhores parâmetros para AG_NV, este trabalho é mais difícil, uma vez que o objetivo principal, número de veículos é discreto e varia pouco alterando as configurações dos parâmetros de AG_NV. Conseqüentemente, optou-se por ajustar todos os parâmetros de AG_NV empiricamente, executando o algoritmo para as instâncias que apresentavam os piores resultados comparados aos já conhecidos. A Tabela 7.1 abaixo mostra o número de vezes que cada mutação, definida do Capítulo 4 foi utilizada na obtenção dos resultados mostrados neste capítulo.

Faz-se necessário observar que os números da Tabela 7.1 parecem ser muito altos para uma população de 50 indivíduos, o que implicaria em uma diversificação muito acentuada, transformando o algoritmo genético em simples busca local, com pouco aproveitamento da memória contida nos indivíduos de maior adaptabilidade. Isso pode ocorrer se a quantidade de operações de troca de consumidores realizadas em um indivíduo ou seu descendente for muito alta antes de uma nova avaliação (na geração seguinte), acaba-se por realizar uma busca aleatória, que nada tem a ver com a região inicial que o indivíduo originalmente representava. Para evitar esta diversificação acentuada, foi utilizada uma taxa de probabilidade de se executar cada conjunto de mutações, `PROB_MUTATION`. O valor ideal encontrado também por testes empíricos foi de 30%.

Adicionalmente, existe uma grande chance da mutação falhar quando aplicada a um indivíduo solução. Isso ocorre uma vez que os operadores próprios de redução de veículos, quando aplicados a indivíduos representantes de mínimos locais (com número de rotas já reduzido) possuem restrições de carga e janela de tempo muito apertadas, apresentando pouca chance de permitir que tais operadores tenham sucesso.

Tabela 7.1: Número de vezes que cada tipo de mutação foi aplicada por geração, utilizado em AG_NV.

Operadores de Mutação	Abreviação	Execuções / geração
Mutação por Troca Simples de Consumidores em Rotas Distintas	M_TSC	40
Mutação por Troca Simples de Consumidores na Mesma Rota	M_TSCMR	10
Mutação por Troca de Consumidores com Redução de Adap_DETCMR, Adap_DEUCMR e Adap_DEUCRR	M_TCRAdap1	1
Mutação por Troca Simples de Consumidores com Redução de Adap_DETCMR, Adap_DEUCMR	M_TCRAdap2	5
Mutação por Troca de Consumidores com Redução da Distância Total Percorrida	M_TCDT	17
Mutação por Retirada de Consumidor de uma Rota Qualquer	M_RC	5
Mutação por Retirada de Consumidor da Rota Receptora	M_RCRR	100
Mutação por Reinserção de Um Consumidor com ganho em DT	M_RDT	10

Mutação de Eliminação de Consumidores da Rota Receptora	M_ETCRR	60
Mutação de Eliminação do Último Consumidor da Menor Rota com Troca de um Consumidor em Outra Rota Qualquer	M_EUTC1	10
Mutação de Eliminação do Último Consumidor da Menor Rota com Troca de dois Consumidor em outra rota qualquer	M_EUTC2	10
Mutação de Eliminação de Toda a Menor Rota	M_EUR	1

Foram utilizados 50 indivíduos em cada população. O algoritmo genético para minimização de número de veículos foi executado em uma primeira fase, reduzindo o número de veículos, conforme descrito no Capítulo 4. Foram utilizados 15 minutos de intervalo fixo, após o qual o algoritmo era interrompido, passando toda a última geração para o HGC_DT, conforme diagrama da Figura 3.16. Novos mínimos locais foram então produzidos, conforme o algoritmo padrão de geração de rotas já descrito, para a solução futura de um PPC utilizando o pacote de programação linear inteira, o GLPK. HGC_DT foi executada em um intervalo também fixo 60 minutos, após o qual era interrompido. Somente após a interrupção o PPC era iniciado. Novamente, como o número de rotas do conjunto R_{GLOBAL} do PPC é pequeno, este foi resolvido instantaneamente, consumindo sempre menos de 1 segundo.

O número de execuções de AG_DT, tratando o problema completo em HGC_DT, foi novamente fixado em 5 a cada ciclo. Já o número de execuções tratando o problema reduzido, pelo mesmo algoritmo AG_DT, foi fixado em 15. Ou seja, não houve alteração na configuração de HGC_DT para o tratamento do PRVJT minimizando NV, exceto na atribuição das adaptabilidades dos indivíduos, conforme foi descrito no capítulo 4.

O processo de seleção do AG_NV, responsável por minimizar NV na primeira fase, foi baseado no torneio múltiplo, utilizando 3 indivíduos aleatoriamente escolhidos. O indivíduo vencedor do torneio, que estará participando na etapa de recombinação, foi escolhido utilizando indicadores hierárquicos. Diferentes prioridades foram testadas quanto aos indicadores, sendo a ordem exibida na Tabela 7.2 a que resultou nos melhores resultados.

Tabela 7.2: Indicadores de Adaptabilidade dos indivíduos soluções em AG_NV, quantificando a dificuldade de eliminação de uma rota por diferentes aspectos.

Prioridade	Indicador de Adaptabilidade	Abreviação
1 ^a	Número de rotas (veículos)	<i>Adap_NV</i>
2 ^a	Número de Consumidores na Menor Rota	<i>Adap_NCMR</i>
3 ^a	Dificuldade de Eliminar Todos os Consumidores da Menor Rota	<i>Adap_DETCMR</i>
4 ^a	Dificuldade de Eliminar Um Consumidor da Menor Rota	<i>Adap_DEUCMR</i>
5 ^a	Dificuldade de Eliminar Um Consumidor da Rota Receptora	<i>Adap_DEUCRR</i>
6 ^a	Dificuldade de Eliminar Todos os Consumidores da Rota Receptora	<i>Adap_DETCRR</i>
7 ^a	Distância Total Percorrida	<i>Adap_DT</i>
8 ^a	Número de Consumidores na Rota Receptora	<i>Adap_NCRC</i>
9 ^a	Soma dos Quadrados dos Números de Consumidores das Rotas	<i>Adap_SQNC</i>
10 ^a	Carga Total da Rota Receptora	<i>Adap_CTRR</i>

Obviamente os indicadores *Adap_NV* e *Adap_NCMR* em 1^a e 2^a prioridade respectivamente, superaram qualquer outra tentativa, para qualquer opção dos demais. O primeiro, *Adap_NV*, representa a própria função objetivo, ou seja, o número de veículos presentes nos indivíduos solução do problema. O segundo critério é a forma mais natural de eliminar gradativamente uma rota, reduzindo gradativamente o seu número de consumidores. Da 3^a posição em diante, o melhor desempenho do algoritmo já não é tão facilmente avaliado. A escolha foi feita por tentativa, avaliando os resultados para algumas instâncias de diferentes classes. Observou-se que os *Adap_DEUCMR* e *Adap_DETCMR* são os critérios mais importantes após o número de veículos da menor rota (*Adap_NCMR*). O indicador *Adap_NCMR*, conforme descrito no Capítulo 4, indica a violação de tempo eliminando um único consumidor da menor rota atual, quase o mesmo tempo de retardo mínimo ou *minimal delay* D_R , equação 1.22, proposto por Homberger [HG99]. Entretanto, na comparação de desempenho do algoritmo trocando os dois na 3^a e 4^a posição, mostrou que *Adap_DETCMR* deve ser considerado prioritário. Este indicador representa a soma do tempo de retardo para todos os consumidores da menor rota. A utilização de *Adap_DEUCMR*, que considera apenas o consumidor de menor tempo de retardo, provoca

a eliminação rápida de vários consumidores de uma rota, porém resultou em maior ocorrência de convergência para mínimos locais.

A utilização do conceito de tempo de retardo mínimo também para os consumidores da *rota receptora*, que representa a rota com menor violação de tempo ao receber um consumidor da menor rota também propiciou um enorme aumento na capacidade de evitar mínimos locais em AG_NV.

Porém, observa-se que embora após 4 critérios utilizados no torneio, muitas vezes ainda ocorre um empate. A grande quantidade de indicadores possibilitou que uma simples troca de dois consumidores quaisquer em rotas *a priori* sem importância, o que poderia não ter efeito nos primeiros indicadores da Tabela 7.2, poderia resultar em alguma variação em alguns dos outros indicadores menos prioritários, melhorando a seleção dos indivíduos.

Em 7ª posição está *Adap_DT*, já não oferecendo risco para direcionar a busca em regiões exclusivamente de menor DT, se opondo a NV. Por fim, nas ultimas prioridades estão o número de consumidores da rota receptora *Adap_NCRC* (8ª), a soma do quadrado dos consumidores das rotas *Adap_SQNC* (9ª) e por fim, a carga total da rota receptora *Adap_CTRR* (10ª). Desta forma dificilmente haverá empate na seleção por torneio, havendo sempre uma solução que apresenta de alguma maneira uma melhor indicação de que haverá menor dificuldade de eliminação de novas rotas.

7.2 Resultados

Os resultados individuais de HGC_NV para os 56 problemas de Solomon de 100 consumidores são mostrados na Tabela 7.3. Na primeira e sexta colunas são apresentadas as instâncias de Solomon com 100 consumidores. As duas colunas seguintes correspondem ao número de veículos (NV) e a distância total percorrida (DT) correspondente à melhor solução conhecida na literatura. As colunas 4ª, 5ª, 9ª e 10ª correspondem aos resultados obtidos neste trabalho com HGC_NV. Os resultados apresentados são os melhores de 3 execuções, utilizando um computador Pentium IV, 2.4 Mhz e 512 Mbytes de RAM. Os resultados em termos de NV foram muito expressivos, atingindo o menor valor conhecido

em 100% dos problemas. A distância total percorrida (DT), que representa o segundo objetivo do problema, ficou em geral acima dos melhores resultados obtidos.

Tabela 7.3: Melhor resultados conhecido atualmente (dez/2004) e os resultados obtidos utilizando HGC_NV.

Instância	Melhor conhecido		HGC_NV		Instância	Melhor conhecido		HGC_NV	
	NV	DT	NV	DT		NV	DT	NV	DT
R101	19	1645.79	19	1650.8	R201	4	1252.37	4	1263.8
R102	17	1486.12	17	1486.64	R202	3	1191.7	3	1240.54
R103	13	1292.68	13	1298.16	R203	3	939.54	3	983.56
R104	9	1007.24	9	1012.59	R204	2	825.52	2	887.56
R105	14	1377.11	14	1400.00	R205	3	994.42	3	1057.63
R106	12	1251.98	12	1266.15	R206	3	906.14	3	952.93
R107	10	1104.66	10	1122.24	R207	2	893.33	2	970.54
R108	9	960.88	9	964.53	R208	2	726.75	2	808.66
R109	11	1194.73	11	1220.72	R209	3	909.16	3	981.06
R110	10	1118.59	10	1135.82	R210	3	939.34	3	1011.28
R111	10	1096.72	10	1112.83	R211	2	892.71	2	976.63
R112	9	982.14	9	1020.52					
C101	10	828.94	10	828.94	C201	3	591.56	3	591.56
C102	10	828.94	10	828.94	C202	3	591.56	3	591.56
C103	10	828.06	10	828.06	C203	3	591.17	3	591.17
C104	10	824.78	10	824.78	C204	3	590.6	3	590.60
C105	10	828.94	10	828.94	C205	3	588.88	3	588.88
C106	10	828.94	10	828.94	C206	3	588.49	3	588.49
C107	10	828.94	10	828.94	C207	3	588.29	3	588.29
C108	10	828.94	10	828.94	C208	3	588.32	3	588.32
C109	10	828.94	10	828.94					
RC101	14	1696.94	14	1698.82	RC201	4	1406.91	4	1462.21
RC102	12	1554.75	12	1623.23	RC202	3	1367.09	3	1514.07
RC103	11	1261.67	11	1277.97	RC203	3	1049.62	3	1081.33
RC104	10	1135.48	10	1163.31	RC204	3	798.41	3	815.62
RC105	13	1629.44	13	1681.43	RC205	4	1297.19	4	1356.34
RC106	11	1424.73	11	1468.49	RC206	3	1146.32	3	1294.6
RC107	11	1230.48	11	1246.35	RC207	3	1061.14	3	1149.17
RC108	10	1139.82	10	1178.96	RC208	3	828.14	3	891.19

A Tabela 7.4 apresenta as médias de NV e de DT para cada classe de Solomon. Os trabalhos foram ordenados conforme os objetivos em termos do resultado global apresentado na penúltima coluna. HGC_NV está em segundo lugar, pois obteve o melhor resultado global em NV juntamente com [Brä01], perdendo para este em DT. Isso mostra que o AG_NV é muito eficiente para a minimização do número de veículos, resultando nos melhores resultados (em termos de NV) já encontrado até o momento, considerando o fator

tempo, visto que [Brä01] utilizou 198 minutos. Por outro lado, AG_DT, principal responsável por gerar as colunas com DT reduzido em HGC_NV, não foi tão eficiente minimizando distância para um número de veículos já reduzido. Pelo menos não o foi tanto quanto na situação onde este número era livre, conforme mostra os resultados do capítulo anterior. Os resultados em termos médios de DT, segundo objetivo, ficaram razoavelmente superiores aos encontrados por [Brä01], 2,08% acima. A comparação com os demais trabalhos só é justa individualmente por classe, onde o número de veículos é o mesmo, pois como foi visto no Capítulo 6, distância percorrida e número de veículos são altamente concorrentes.

Tabela 7.4: Média do número de veículos e da distância percorrida dos principais resultados na literatura

Author	R1	R2	C1	C2	RC1	RC2	NV DT	Equipamento
[HO04]	11.92 1211.1	2.73 954.26	10 828.4	3 589.9	11.5 1384.2	3.25 1124.5	405 57273	Sun Ultra 440 MHz ? x 120 min
[Brä01]	11.92 1222.1	2.73 975.12	10 828.4	3 589.9	11.5 1389.6	3.25 1128.4	405 57710	Pentium 200 MHz, 1x (198) min.
HGC_NV	11.92 1224.3	2.73 1012.2	10 828.4	3 590.9	11.5 1417.3	3.25 1195.6	405 58911	Pentium IV 2,4 GHz 3x (75 min)
[GH01]	12 1217.6	2.73 961.29	10 828.6	3 590.3	11.5 1395.1	3.25 1139.4	406 57641	4xPentium 400 MHz, 5x (1458) min.
[HG99]	11.92 1228.1	2.73 969.95	10 828.4	3 589.9	11.63 1392.6	3.25 1144.4	406 57876	Pentium 200 MHz, 10x (312) min.
[BBB01]	12.17 1251.4	2.73 1056.6	10 828.5	3 590.1	11.88 1414.9	3.25 1258.2	411 60200	Pentium 400 MHz, 1x (162) min.
[RR95]	12.66 1317	2.91 1167	10 930	3 681	12.38 1523	3.38 1398	424 65827	PC 486/66 MHz, 3x (0.62) min.
[LS99]	12.17 1249.6	2.82 1016.6	10 830.1	3 591	11.88 1412.9	3.25 1204.9	412 59318	HP 9000/720, 3x (102) min.
[GH99]	12.42 1198	2.82 947	10 829	3 590	11.88 1356	3.25 1144	415 56946	4xPentium 200 MHz, 1x - (48) min.
[TBG+97]	12.33 1220.4	3 1013.4	10 828.5	3 590.9	11.9 1381.3	3.38 1198.6	417 58614	Sun Sparc 10, 1x - (248) min.
[GT99]	12.38 1210.8	3 960.31	10 828.4	3 591.9	11.92 1388.1	3.33 1149.3	418 57583	Sun Ultrasparc 1, 1x - (210) min.
[CL99]	12.42 1233.3	3.09 990.99	10 828.4	3 596.6	12 1403.7	3.38 1221	420 58927	Pentium 300 MHz, 1 x - (18) min.
[RT95]	12.58 1197.4	3.09 954.36	10 828.5	3 590.3	12.38 1369.5	3.62 1139.8	427 57120	Silicon 100MHz, 1 x - 92.2 (138) min.

Na classe R1, problemas randômicos somente [HG99], [Brä01] e [HO04] atingiram a marca de HGC_NV, sendo o resultado de HGC_NV médio melhor em termos de DT que [HG99]. Na classe R2 o número de veículos mínimo de 2.73 foi alcançado por vários autores. Vários deles com resultados em DT melhor que HGC_NV. Nas classes C1 e C2, os resultados também não variam muito. Em RC1 os resultados de HGC_NV estão mais próximos daqueles com o mesmo número de veículos. Novamente na classe RC2 o valor encontrado em DT médio por HGC_NV foi mais expressivo. Nota-se que a estratégia de geração de colunas utilizada em HGC_NV não apresenta grande vantagem quando se utilizam problemas com baixo número de rotas, como é o caso dos problemas de R2 e RC2.

Os resultados mostraram que a estratégia de avaliar os indivíduos soluções no AG_NV hierarquicamente é muito competitiva, produzindo uma heurística que está entre as melhores na redução do número de veículos. Por outro lado, os operadores e a configuração de busca utilizada em AG_DT com a simples alteração da função objetivo deixa um pouco a desejar quanto à minimização da distância percorrida. Mais uma vez, fica evidente a concorrência dos objetivos, número de veículos e distância total percorrida.

7.3 Conclusões

Neste capítulo os resultados de HGC_NV foram confrontados com os principais trabalhos na literatura para minimização de número de veículos, seguido da distância total percorrida. A heurística apresentou o melhor resultado quanto ao número de veículos e um resultado razoável no segundo critério (DT). Especificamente o algoritmo AG_NV proposto para a redução do número de veículos, executado somente durante 15 minutos (tempo bem inferior ao normalmente utilizado por outros autores), foi o responsável pelos bons resultados em termos de NV.

Os indicadores de adaptabilidade foram o principal motivo do excelente desempenho de AG_NV para minimização do número de veículos. Conforme já indicava o trabalho Homberger [HG99], que também apresenta um bom valor médio de NV, o tempo mínimo de retardo (D_R) é adequado quando se trabalha com soluções viáveis. Os quatro indicadores

subseqüentes ao número de consumidores da menor rota, representam na realidade uma extensão da idéia do D_R aplicada aos demais consumidores da menor rota e da rota receptora.

Quanto a minimização de DT para um dado valor minimizado de NV na fase anterior, a heurística HGC_NV utiliza o mesmo algoritmo AG_DT, que apresentou bons resultados no capítulo anterior, exceto quanto a função adaptabilidade. Porém os resultados de HGC_DT, avaliada muito positivamente para redução de distância como único objetivo no capítulo 6, quando empregada em HGC_NV não obteve tão bons resultados como esperado. Há indicações de que AG_NV não está adequado para uma busca por redução de DT com a nova função objetivo, que considera prioritariamente NV. As observações realizadas mostram que os indivíduos soluções de AG_DT em HGC_NV não apresentam uma diversidade adequada como ocorre em HGC_DT, pois os operadores utilizados apresentam um índice de falha alto. Este efeito pode ser resultado da dificuldade em gerar alterações nos indivíduos mantendo-os viáveis. Isso ocorre devido o aperto das restrições janela de tempo e carga quando se trabalha com o número de veículos reduzidos. Conclui-se que os resultados em DT poderiam ser melhorados com operadores no AG_DT mais adequados para um NV já reduzido. Isso possibilitará avaliar se o modelo de particionamento de conjuntos tão importante em HGC_DT ainda continua sendo uma boa estratégia para aumentar a robustez da heurística, quando minimizando NV como primeiro objetivo.

Capítulo 8

Resultados Minimizando Distância no PRVJT Dinâmico

8.1 Introdução

Se a avaliação de desempenho de algoritmos para o PRVJT estático já não é uma tarefa simples, para o problema dinâmico se torna ainda mais difícil. No problema estático, as variáveis relacionadas à dificuldade do problema eram a distribuição geográfica dos consumidores, o grau de aperto das restrições, como janela de tempo e capacidade dos veículos, e principalmente o número de consumidores. Diferente do problema estático, no problema dinâmico as variáveis não são todas conhecidas no momento inicial do processo de otimização. Conseqüentemente, não somente o conteúdo das informações é importante, mas quando elas são conhecidas. Por isso, um fator muito importante é a quantidade de informações recebidas após o início do algoritmo. Alguns autores têm avaliado o quanto estas informações conhecidas *a posteriori* influenciam no resultado. No entanto, ainda não há um critério claro de medição, até mesmo porque muitas podem ser as variáveis alteradas ou conhecidas ao longo do processo de otimização. A inexistência de instâncias para o problema dinâmico bem definidas, para que os resultados possam ser confrontados, também contribuem para que a comparação de algoritmos seja uma tarefa difícil.

Neste trabalho, a proposta de utilização das consagradas instâncias de Solomon também para avaliar o PRVDJT é uma tentativa de deixar mensuráveis os testes, e conseguinte, a avaliação de HGC_DT como proposta de solução para o problema dinâmico. A grande vantagem da utilização dessas instâncias é que se conhece muito bem os resultados desejados, pois já foram testadas na literatura por muitas propostas, ainda que para o problema estático. Para que estes resultados finais possam ser utilizados, o problema é inicialmente modificado, retirando informações ou realizando alterações. Entretanto, antes do processo de término da otimização a instância deve retornar ao conteúdo original, assim como ela é para o problema estático. Isso significa que a solução desejada, se possível

encontrá-la dentro do limite de tempo disponível, é exatamente aquela conhecida para o problema estático.

A solução heurística proposta para o PRVJT nos capítulos 3 e 4 parecem ser adequadas para o tratamento do problema dinâmico. Nesta tese somente será avaliada a proposta do capítulo 3, onde o objetivo único é a minimização de distância. Para avaliar HGC_DT, foram criadas diferentes condições típicas de um cenário real, aonde novas informações chegam a todo instante, mesmo após o início do algoritmo. Variações no tipo de informação, na quantidade de informações e no momento que tais informações são inseridas caracterizam diferentes graus e tipos de dinamicidade que serão avaliados.

8.2 Definindo os cenários dinâmicos a partir das instâncias de Solomon

Nesta seção são definidos os diferentes tipos de problemas dinâmicos utilizando como referência as instâncias de Solomon, observando também alguns problemas que ocorrem no mundo real.

No cenário hipotético do problema, durante o processo de otimização os veículos estão todos no depósito central. Evita-se também considerar alguns aspectos mais complexos relacionados à entrega de mercadorias, como o tempo necessário para colocá-las nos veículos. Inicia-se o algoritmo somente com uma parcela previamente conhecida dos dados. Novas informações são aceitas até o último instante antes da partida dos veículos. Conforme vai se tomando conhecimento das novas informações, estas vão sendo inseridas imediatamente no problema, forçando o algoritmo proposto a considerar um conjunto de informações novas.

Este cenário é condizente com alguns problemas dinâmicos do mundo real. Um exemplo é o transporte de valores. Uma frota de carros blindados é requisitada para coletarem quantias em dinheiro e outros valores em bancos e estabelecimentos comerciais a partir do término do horário comercial. Parte destes pedidos são rotineiros, sendo possível considerar tais pedidos com bastante antecedência. No entanto, outra parte destes pedidos chegam a qualquer momento, devendo ser inseridos sempre que possível de forma

otimizada no conjunto de veículos que deverá sair para fazer a coleta. Geralmente existe uma janela de tempo associada, bem como uma quantia máxima a ser transportada, geralmente relacionada ao contrato de seguro estabelecido pela empresa, tornando o problema muito próximo ao PRVDJT tratado nesta tese. Embora em alguns cenários pedidos possam ser inseridos nas rotas dos veículos já em trânsito, estes não são tratados aqui.

Um outro cenário típico e bem popular nos dias atuais é a entrega de comida expressa, como pizzas, sanduíches e outros. Os pedidos vão chegando a todo o momento e os veículos ou motocicletas devem realizar a entrega das encomendas reduzindo a distância percorrida e o tempo de atendimento. A cada período fixo, geralmente pequenos intervalos de tempo (Ex: 15 minutos), os novos pedidos são reunidos em grupos, sendo necessário definir o conjunto de veículos necessários para a entrega. Uma primeira opção simples é iniciar o algoritmo de roteamento somente após conhecer o último pedido dentro do intervalo. Entretanto isso pode gerar um tempo extra, necessário à otimização, prejudicial ao negócio. Um algoritmo ideal em tempo real ou *online*, como é muitas vezes chamado, deverá aproveitar todo o tempo disponível para realizar a busca, minimizando a necessidade de processamento posterior ao conhecimento do último consumidor ou pedido e, conseqüentemente, minimizando a necessidade de espera para despachar os veículos. Também deverá ser considerada a possibilidade de cancelamento de encomendas após o início do processo de otimização.

8.2.1 Chegada de novos consumidores

Com o intuito de abranger os tipos de cenários descritos e, adicionalmente, possibilitar uma comparação direta com os resultados estáticos da literatura, o primeiro tipo de problema dinâmico foi proposto. Este considera a chegada posterior ao início da otimização de novas encomendas ou consumidores. O horizonte total de 60 minutos será utilizado, sendo o mesmo adotado para os resultados de HGC_DT no problema estático do capítulo 6. Enquanto no problema estático o conjunto completo de consumidores é conhecido desde o início do processo, agora parte deste será disponibilizado ao longo do tempo, no decorrer do processo de otimização, dificultando sobremaneira o problema para a heurística desenvolvida. É fácil constatar que quanto maior o número de variáveis

conhecidas ou alteradas durante o processo de otimização, mais difícil será aproveitar os resultados da busca já realizada. Também é simples observar que o momento em que tais alterações ocorrem vai influenciar muito no grau de dificuldade para o algoritmo, principalmente se alterações são causadas próximas ao término da sua execução.

Nos dois primeiros cenários será considerado o período compreendido entre 45 e 55 minutos como o intervalo de chegada de novos consumidores. Dois graus de dinamismo serão avaliados, o primeiro com 30% dos consumidores sendo desconhecidos na primeira etapa da otimização, o segundo com 50%. Isso significa que para as instâncias de Solomon com 100 consumidores originais, 30 e 50, respectivamente, serão omitidos durante os primeiros 45 minutos de um horizonte total de 60 minutos disponível para HGC_DT. As encomendas desconhecidas serão informadas uma a uma, distribuídas uniformemente no período de chegada, sendo a última informada aos 55 minutos, restando apenas 5 minutos de processamento.

O objetivo principal é que HGC_DT tenha um desempenho melhor que um método do tipo re-otimização tratando o problema dinâmico, que necessitaria reiniciar o algoritmo completamente no momento da chegada da última encomenda, utilizando o tempo reduzido de 5 minutos para a busca. Para atingir este objetivo, HGC_DT apresenta duas características interessantes. A primeira é o curto intervalo de tempo utilizado para cada execução de AG_DT. Isso permite uma expectativa no mínimo igual a um método do tipo *restart*, pois haverá uma, duas, ou pelo menos grande parte de uma execução do AG_DT considerando todas as encomendas. Isso fará com que o conjunto R_{GLOBAL} do problema principal tenha um mínimo de rotas geradas por uma execução do AG_DT considerando todos os consumidores. A segunda característica que poderá deixar HGC_DT superior a outras metodologias, é que existe ainda a possibilidade de rotas geradas na fase inicial serem de grande valia na qualidade do resultado final, se estas de alguma forma não são afetadas pela inclusão de novos consumidores.

Três execuções são realizadas também utilizando dupla precisão, possibilitando comparar os resultados com aqueles do problema estático. As tabelas Tabela 8.1 e Tabela 8.2 apresentam respectivamente o resultado das três execuções para cada grau de dinamismo proposto, com 30 e 50 consumidores, conhecidos posteriormente.

Tabela 8.1: Resultados da execução de HGC_DT utilizando 30 consumidores informados no intervalo [45-55] minutos. O tempo de execução foi de 60 minutos, utilizando um computador utilizado Pentium IV, 2.4 MHz, 512 de RAM.

Instância	Melhor Estático		30 Consumidores informados no intervalo [45-55] min.						Distância do Estático	
			Execução 1		Execução 2		Execução 3			
	NV	DT	NV	DT	NV	DT	NV	DT	NV	DT
C101	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C102	10	828,937	10	831,897	10	828,937	10	837,599	0	0,00%
C103	10	828,065	10	831,199	10	834,160	10	833,311	0	0,38%
C104	10	824,770	10	828,066	10	831,270	10	824,777	0	0,00%
C105	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C106	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C107	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C108	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C109	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C201	3	591,557	3	591,557	3	591,557	3	591,557	0	0,00%
C202	3	591,557	3	591,557	3	591,557	3	591,557	0	0,00%
C203	3	591,173	3	603,367	3	591,173	3	610,038	0	0,00%
C204	3	590,599	3	590,599	5	684,273	3	610,682	0	0,00%
C205	3	588,876	3	588,876	3	588,876	3	588,876	0	0,00%
C206	3	588,493	3	588,493	3	588,493	3	588,493	0	0,00%
C207	3	588,286	3	588,286	3	588,286	3	588,286	0	0,00%
C208	3	588,324	3	588,324	3	588,324	3	588,324	0	0,00%
R101	20	1642,870	20	1644,246	20	1661,062	20	1644,815	0	0,08%
R102	18	1472,620	19	1486,269	18	1495,450	19	1488,026	1	0,93%
R103	14	1213,620	15	1233,401	14	1222,417	15	1235,921	0	0,72%
R104	11	986,096	11	996,073	11	1038,838	11	1014,156	0	1,01%
R105	15	1360,783	15	1363,627	16	1387,218	15	1381,152	0	0,21%
R106	13	1241,518	13	1251,050	14	1276,923	14	1272,697	0	0,77%
R107	11	1076,125	12	1093,900	12	1120,293	13	1116,515	1	1,65%
R108	10	948,573	10	965,757	11	969,336	11	982,143	0	1,81%
R109	13	1151,839	13	1181,720	14	1184,913	13	1185,450	0	2,59%
R110	12	1092,347	12	1106,599	12	1136,244	12	1125,573	0	1,30%
R111	12	1053,496	12	1071,050	13	1116,474	12	1079,340	0	1,67%
R112	10	960,675	11	1008,235	11	1015,999	11	993,788	1	3,45%
R201	9	1148,483	7	1206,245	6	1198,959	6	1229,048	-3	4,40%
R202	7	1049,737	11	1119,214	7	1082,222	5	1134,246	0	3,09%
R203	5	900,080	5	980,670	6	954,702	6	980,691	1	6,07%
R204	4	772,330	4	823,413	4	829,357	4	836,964	0	6,61%
R205	6	970,886	5	1057,341	5	1088,214	5	1008,207	-1	3,84%
R206	5	898,914	5	1015,218	5	1008,182	5	1025,325	0	12,16%
R207	4	834,930	4	936,373	3	920,767	4	907,948	0	8,75%
R208	3	723,610	4	829,529	3	784,957	5	790,101	0	8,48%
R209	6	879,531	5	951,142	5	953,066	6	967,382	-1	8,14%
R210	7	932,887	5	985,048	3	1021,569	5	952,454	-2	2,10%
R211	5	787,511	4	840,430	5	857,037	5	862,728	-1	6,72%
RC101	16	1639,968	15	1645,573	17	1667,432	16	1652,207	-1	0,34%
RC102	14	1466,840	15	1496,389	15	1490,408	15	1486,701	1	1,35%
RC103	11	1264,707	12	1304,592	12	1292,872	12	1317,009	1	2,23%
RC104	10	1135,520	11	1212,975	11	1176,632	10	1162,492	0	2,38%
RC105	16	1518,600	15	1583,565	16	1565,348	16	1568,373	0	3,08%
RC106	13	1377,352	14	1412,485	13	1402,037	14	1396,857	1	1,42%
RC107	12	1212,830	13	1246,148	13	1292,567	12	1254,848	1	2,75%
RC108	11	1117,526	11	1166,219	11	1160,984	12	1196,635	0	3,89%
RC201	9	1274,537	8	1347,624	6	1351,615	8	1335,921	-1	4,82%
RC202	8	1113,526	7	1171,845	7	1144,647	6	1140,944	-2	2,46%
RC203	5	945,960	5	977,774	5	976,305	5	968,641	0	2,40%
RC204	4	799,670	4	831,057	4	838,424	5	831,328	0	3,92%
RC205	7	1161,810	6	1243,960	7	1257,183	7	1203,915	0	3,62%
RC206	7	1059,886	5	1142,815	5	1156,687	6	1157,785	-2	7,82%
RC207	7	976,396	6	1028,182	6	1029,326	6	1015,002	-1	3,95%
RC208	5	795,391	5	894,765	4	901,746	4	927,103	0	12,49%

Tabela 8.2: Resultados da execução de HGC_DT utilizando 50 consumidores informados no intervalo [45-55] minutos. O tempo de execução foi de 60 minutos, utilizando um computador utilizado Pentium IV, 2.4 MHz, 512 de RAM.

Instância	Melhor Estático		50 Consumidores informados no intervalo [45-55] min.						Distância do Estático	
	NV	DT	Execução 1		Execução 2		Execução 3		NV	DT
			NV	DT	NV	DT	NV	DT		
C101	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C102	10	828,937	10	837,599	10	834,639	10	837,599	0	0,69%
C103	10	828,065	10	828,826	10	831,199	10	834,160	0	0,09%
C104	10	824,770	10	839,497	10	827,737	10	830,871	0	0,36%
C105	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C106	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C107	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C108	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C109	10	828,937	10	828,937	10	828,937	10	828,937	0	0,00%
C201	3	591,557	3	591,557	3	591,557	3	591,557	0	0,00%
C202	3	591,557	3	591,557	3	607,643	3	591,557	0	0,00%
C203	3	591,173	3	622,559	3	610,986	4	604,872	1	2,32%
C204	3	590,599	5	714,417	4	649,275	4	664,000	1	9,93%
C205	3	588,876	3	588,876	3	588,876	3	588,876	0	0,00%
C206	3	588,493	3	588,493	3	588,493	3	588,493	0	0,00%
C207	3	588,286	3	588,286	3	588,286	3	588,286	0	0,00%
C208	3	588,324	4	679,724	3	588,324	3	588,324	0	0,00%
R101	20	1642,870	20	1645,274	20	1645,274	20	1657,380	0	0,15%
R102	18	1472,620	18	1500,784	19	1489,329	18	1488,143	0	1,05%
R103	14	1213,620	14	1234,893	15	1236,599	14	1232,615	0	1,57%
R104	11	986,096	12	1013,315	12	1030,916	12	1006,054	1	2,02%
R105	15	1360,783	15	1368,085	16	1399,887	16	1380,534	0	0,54%
R106	13	1241,518	14	1271,246	14	1271,727	14	1273,879	1	2,39%
R107	11	1076,125	11	1112,485	11	1089,834	11	1103,176	0	1,27%
R108	10	948,573	11	990,055	11	986,910	11	987,830	1	4,04%
R109	13	1151,839	14	1214,556	14	1218,315	14	1195,275	1	3,77%
R110	12	1092,347	12	1128,162	12	1124,772	12	1111,886	0	1,79%
R111	12	1053,496	12	1088,520	12	1094,178	12	1088,533	0	3,32%
R112	10	960,675	11	987,389	11	995,033	11	1008,935	1	2,78%
R201	9	1148,483	7	1186,231	5	1244,333	6	1225,197	-2	3,29%
R202	7	1049,737	3	1116,868	6	1125,285	6	1147,342	-4	6,40%
R203	5	900,080	4	1030,667	5	979,766	5	1004,199	0	8,85%
R204	4	772,330	5	976,522	5	844,306	5	866,076	1	9,32%
R205	6	970,886	5	1093,715	4	1091,410	5	1008,207	-1	3,84%
R206	5	898,914	4	981,781	5	976,219	6	1025,325	0	8,60%
R207	4	834,930	4	915,019	4	917,627	5	921,984	0	9,59%
R208	3	723,610	4	800,309	3	799,977	5	802,714	0	10,55%
R209	6	879,531	5	976,522	3	919,148	6	972,994	-3	4,50%
R210	7	932,887	5	1031,069	5	984,762	5	979,050	-2	4,95%
R211	5	787,511	5	837,930	5	857,981	4	850,752	0	6,40%
RC101	16	1639,968	17	1675,611	16	1645,891	17	1661,508	0	0,36%
RC102	14	1466,840	15	1505,994	15	1498,979	15	1492,486	1	1,75%
RC103	11	1264,707	12	1305,064	12	1277,540	11	1277,631	1	1,01%
RC104	10	1135,520	11	1181,093	11	1172,184	11	1172,737	1	3,23%
RC105	16	1518,600	15	1585,993	15	1584,447	15	1575,636	-1	3,76%
RC106	13	1377,352	14	1411,591	14	1434,528	14	1413,540	1	2,49%
RC107	12	1212,830	12	1284,708	12	1217,054	13	1269,022	0	9,35%
RC108	11	1117,526	11	1171,054	12	1153,688	11	1175,313	1	3,24%
RC201	9	1274,537	7	1350,841	7	1348,270	9	1335,334	0	4,77%
RC202	8	1113,526	7	1124,914	8	1136,560	6	1180,126	-1	1,02%
RC203	5	945,960	5	967,160	5	978,780	5	1031,000	0	2,24%
RC204	4	799,670	5	836,468	4	805,428	4	826,672	0	0,72%
RC205	7	1161,810	5	1251,229	7	1240,674	7	1207,643	0	3,94%
RC206	7	1059,886	5	1130,051	5	1204,409	5	1208,000	-2	6,62%
RC207	7	976,396	5	1042,826	5	1048,389	5	1103,467	-2	6,80%
RC208	5	795,391	5	870,048	5	897,841	5	916,659	0	9,39%

É importante novamente ressaltar que os resultados desejados são os mesmos para o problema estático, uma vez que um algoritmo ideal, capaz de gerar soluções tão rápidas quanto se queira, poderia tratar o problema exatamente como um problema estático.

Os resultados mostram que o aumento do número de consumidores conhecidos *a posteriori* piora os resultados de HGC_DT. Uma vez que nos dois cenários há exatamente os mesmos 5 minutos finais conhecendo todos os consumidores, há menor aproveitamento do conjunto R_{GLOBAL} no PPC para o cenário com maior dinamismo. Os resultados para 30 consumidores conhecidos posteriormente foi 2,60% superior àqueles do problema estático. Já o resultado para 50 consumidores ficou 2,97%, ligeiramente superior. Estes resultados mostram também que o conjunto R_{GLOBAL} foi útil no primeiro caso, pois tanto num quanto no outro, o período de conhecimento de todos consumidores é o mesmo, de 5 minutos. A redução de 0,37% deve-se a melhor qualidade do conjunto de rotas geradas sem o conhecimento completo do problema.

A Tabela 8.3 apresenta o percentual de redução em DT obtido no resultado final (resolução do PPC utilizando R_{GLOBAL}) em relação à melhor execução de AG_DT conhecendo todas as rotas. Foi considerado o problema dinâmico com 30 consumidores conhecidos após o início da otimização. Para as instâncias do problema dinâmico utilizado, com os tempos de execução de AG_DT de 120 segundos, são realizadas 50% de uma execução e 2 execuções inteiras nos últimos 5 minutos de HGC_DT. Somente essas execuções de AG_DT dentro do processo global HGC_DT conhecem todos os consumidores. Adicionalmente é também apresentado o número de problemas onde houve aproveitamento do conjunto R_{GLOBAL} , ou seja, onde o valor final do PPC é menor que o melhor AG_DT. Para o problema dinâmico, esta avaliação é de extrema importância, pois também comprova que as rotas produzidas mesmo sem o conhecimento de todas as encomendas estão sendo utilizadas na composição da solução do PPC, validando a idéia central do algoritmo proposto para o problema dinâmico.

Tabela 8.3: Comparação dos melhores resultados individuais de AG_DT, interno a heurística HGC_DT e o resultado do PCC final. Estes resultados foram retirados da melhor execução de HGC_DT da Tabela 8.1.

Instância	30 Consumidores					
	Melhor AG_DT		Melhor PPC		Ganho	
	NV	DT	NV	DT	NV	DT
C101	10	828,937	10	828,937	0	0,00%
C102	10	837,6	10	828,937	0	1,03%
C103	10	875,36	10	831,199	0	5,04%
C104	10	835,26	10	824,777	0	1,26%
C105	10	828,937	10	828,937	0	0,00%
C106	10	828,937	10	828,937	0	0,00%
C107	10	828,937	10	828,937	0	0,00%
C108	10	828,937	10	828,937	0	0,00%
C109	10	828,937	10	828,937	0	0,00%
Total	90	7522	90	7459	0	0,84%
C201	3	591,557	3	591,557	0	0,00%
C202	3	611,080	3	591,557	0	3,19%
C203	3	603,370	3	591,173	0	2,02%
C204	3	674,880	3	590,599	0	12,49%
C205	3	588,876	3	588,876	0	0,00%
C206	3	588,493	3	588,493	0	0,00%
C207	3	600,610	3	588,286	0	2,05%
C208	3	588,324	3	588,324	0	0,00%
Total	24	4847	24	4719	0	2,65%
R101	21	1661,56	20	1644,246	1	1,04%
R102	19	1495,4	19	1486,269	0	0,61%
R103	14	1236,82	14	1222,417	0	1,16%
R104	12	1006,05	11	996,073	1	0,99%
R105	16	1379,84	15	1363,627	1	1,17%
R106	13	1258,83	13	1251,050	0	0,62%
R107	13	1096,43	12	1093,900	1	0,23%
R108	10	965,76	10	965,757	0	0,00%
R109	14	1235,87	13	1181,720	1	4,38%
R110	12	1143,89	12	1106,599	0	3,26%
R111	12	1071,05	12	1071,050	0	0,00%
R112	11	997,04	11	993,788	0	0,33%
Total	167	14548,54	162	14376,5	5	1,18%
R201	6	1219,91	6	1198,959	0	1,72%
R202	7	1082,22	7	1082,222	0	0,00%
R203	6	954,702	6	954,702	0	0,00%
R204	4	823,41	4	823,413	0	0,00%
R205	5	1008,207	5	1008,207	0	0,00%
R206	5	1008,182	5	1008,182	0	0,00%
R207	4	907,948	4	907,948	0	0,00%
R208	3	784,957	3	784,957	0	0,00%
R209	5	977,46	5	951,142	0	2,69%
R210	5	952,454	5	952,454	0	0,00%
R211	4	898,94	4	840,430	0	6,51%
Total	54	10618,39	54	10512,62	0	1,00%
RC101	15	1656,3	15	1645,573	0	0,65%
RC102	15	1487,3	15	1486,701	0	0,04%
RC103	12	1292,872	12	1292,872	0	0,00%
RC104	12	1197,31	10	1162,492	2	2,91%
RC105	16	1567,01	16	1565,348	0	0,11%
RC106	14	1427,95	14	1396,857	0	2,18%
RC107	13	1252,99	13	1246,148	0	0,55%
RC108	12	1261,57	11	1160,984	1	7,97%
Total	109	11143,3	106	10956,98	3	1,67%

Continua na próxima página

Continuação da página anterior						
30 Consumidores						
Instância	Melhor AG_DT		Melhor PPC		Ganho	
	NV	DT	NV	DT	NV	DT
RC201	7	1403,29	8	1335,921	-1	4,80%
RC202	6	1230,051	6	1140,944	0	7,24%
RC203	5	1043,636	5	968,641	0	7,19%
RC204	4	831,057	4	831,057	0	0,00%
RC205	6	1243,96	7	1203,915	-1	3,22%
RC206	5	1142,815	5	1142,815	0	0,00%
RC207	6	1015,002	6	1015,002	0	0,00%
RC208	4	927,103	5	894,765	-1	3,49%
Total Global	43	8836,914	46	8533,06	-3	3,44%

A Tabela 8.3 apresenta também os ganhos totais percentuais em distância para cada classe, bem como o ganho médio global, que foi de 3.44%. Este valor confirma a validade da proposta de HGC_DT para o tratamento do PRVDJT, mostrando que existe na grande maioria das vezes uma redução na distância percorrida com a solução do PPC. Considerando que duas execuções somente de AG_DT não seriam suficientes para um bom conjunto R_{GLOBAL} , é possível constatar que há rotas importantes presentes neste conjunto geradas durante a fase inicial, sem o conhecimento pleno dos consumidores.

A Figura 8.1 apresenta a evolução dos resultados parciais durante a busca em HGC_DT para o problema C201 à medida que novos consumidores vão sendo inseridos no problema. Os eixos verticais correspondem ao valor em distância total percorrida (DT) e o número de consumidores presentes no problema, da esquerda e direita, respectivamente. A curva inferior mostra que nos primeiros 45 minutos o número de consumidores foi mantido constante, igual a 70. Após os 45 minutos os consumidores vão sendo inseridos um a um, até o instante 55 minutos, onde toda a instância já está conhecida.

A curva superior representa os valores de DT ao longo do processo de busca de HGC_DT. Cada ponto corresponde a um resultado individual do algoritmo genético, AG_DT, utilizado internamente em HGC_DT. Valores bem inferiores estão presentes em etapas intermediárias por se tratarem de evoluções de AG_DT tratando o problema reduzido, que retira consumidores mesmo sendo estes conhecidos, para gerar rotas para o R_{GLOBAL} . Note que os resultados parciais (AG_DT) durante a fase de inserção de consumidores é bem irregular. Variações na qualidade de DT são resultados da inserção abrupta de novos consumidores, em alguns casos nas últimas gerações de indivíduos solução do AG_DT.

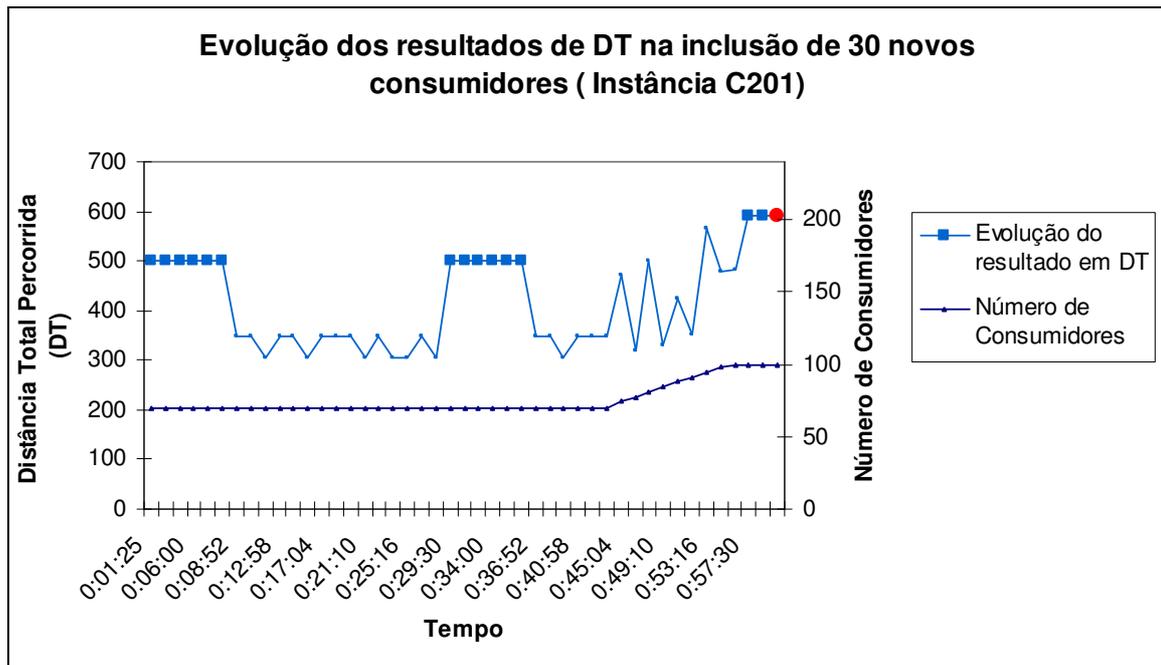


Figura 8.1 : Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema C201.

O último marcador (maior e circular) à direita corresponde ao resultado final do PPC. Para o problema C201, conforme pode ser visto no gráfico da Figura 8.1, as duas últimas soluções de AG_DT independentes são iguais ao resultado global de HGC_DT, onde o valor de DT foi de 591.56 unidades de distância.

O gráfico da Figura 8.2 mostra a mesma evolução dos resultados em DT mas agora para a instância R101. O resultado final do PPC, último ponto à direita, apresentou um valor de DT menor que os resultados individuais de AG_DT, conhecendo o problema completo.

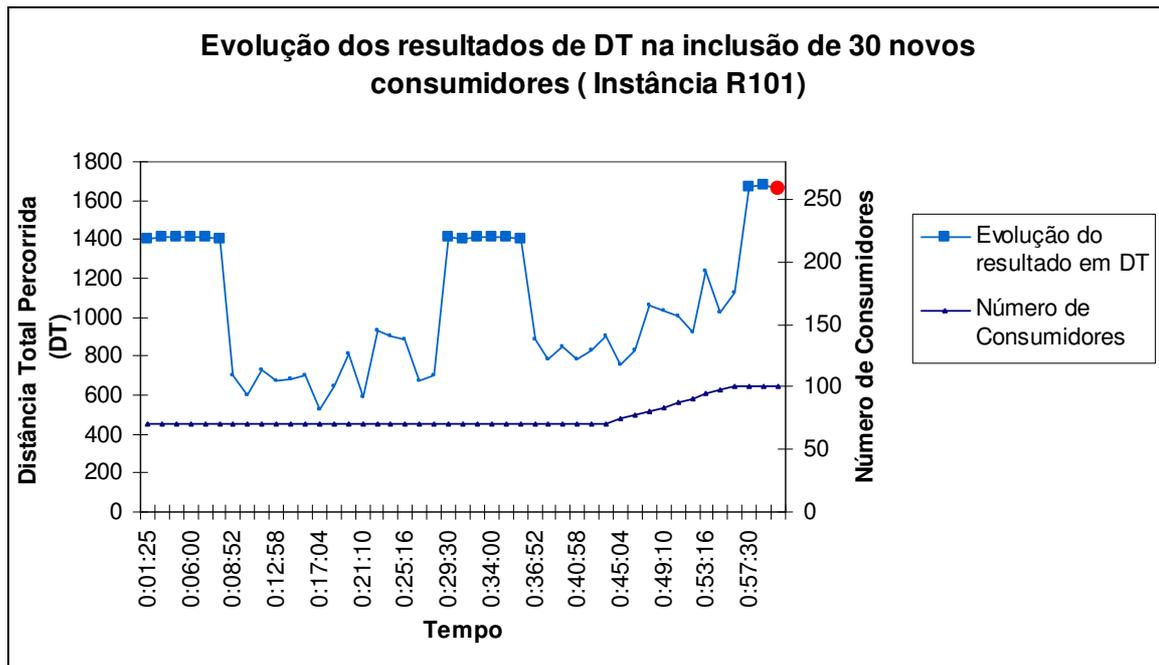


Figura 8.2 : Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema R101.

Um resultado semelhante foi a evolução de HGC_DT com a instância R201, gráfico da Figura 8.3, inserindo também 30 novos consumidores. Conforme pode ser visto no resultado final, as rotas geradas durante o processo de otimização, mesmo sem o conhecimento de todos os consumidores, proporcionaram uma redução em DT, quando na solução do PPC. A evolução é muito semelhante se o número de consumidores informados posteriormente é alterado de 30 para 50, conforme gráfico da Figura 8.4. Entretanto, os resultados finais são piores para 50 consumidores, 1244.33 em unidades de distância, contra 1198.96 do primeiro.

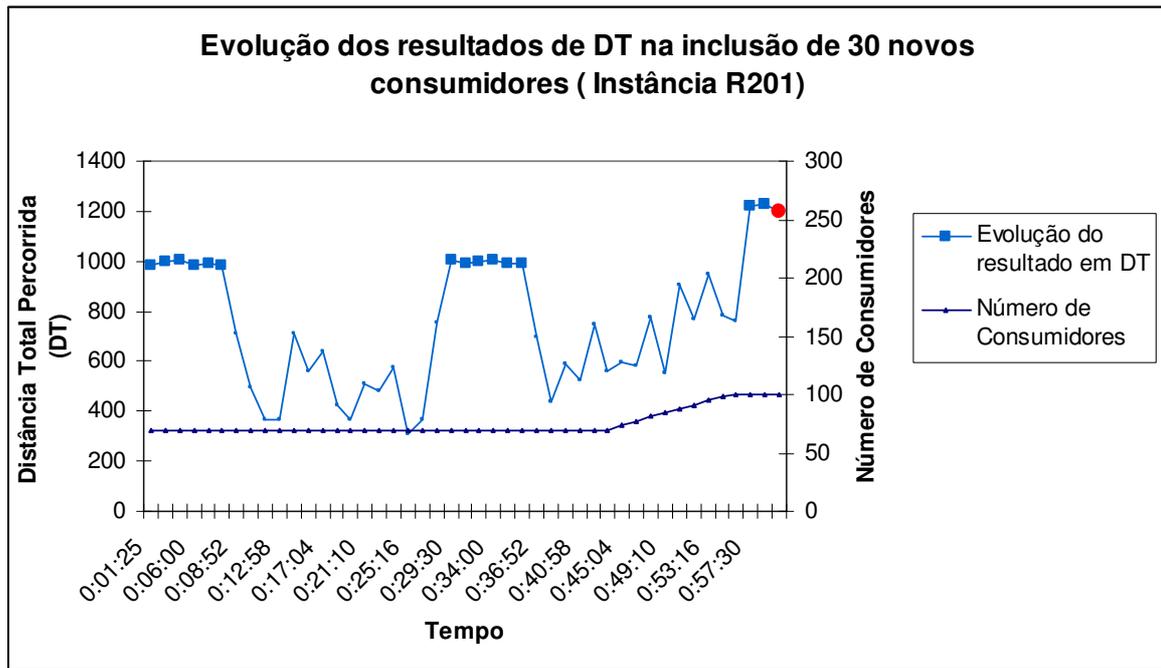


Figura 8.3 Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema R201, com 30 consumidores informados após 45 minutos de execução.

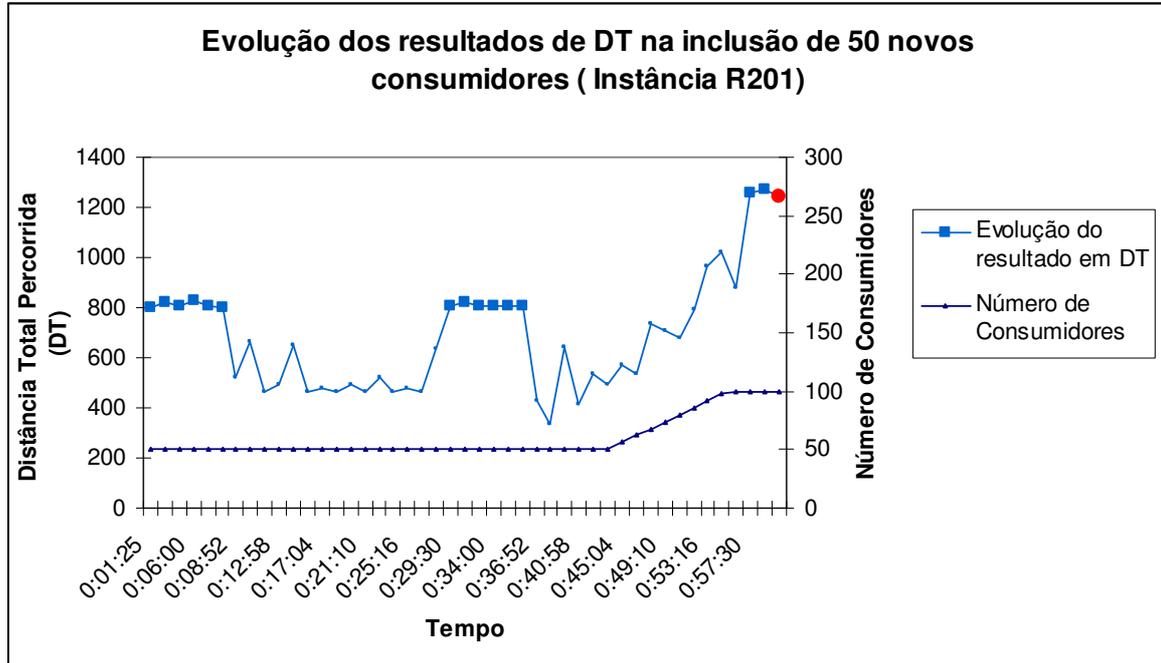


Figura 8.4 Resultados individuais produzidos por AG_DT durante a busca em HGC_DT para o problema R201, com 50 consumidores informados após 45 minutos de execução.

Para avaliar a influência do grau de dinamismo do problema com a qualidade da solução de HGC_DT, diferentes percentuais de consumidores são inseridos *a posteriori* na instância R101. O intervalo de conhecimento dos novos consumidores desta vez foi expandido, considerando desde o início do processo de otimização (instante zero) até 58 minutos após o início. Desta forma somente uma execução de AG_DT é possível com o conhecimento completo dos consumidores. Os resultados finais são apresentados no gráfico da Figura 8.5. Resultados de HGC_DT para diferentes percentuais de consumidores são apresentados. A curva superior no gráfico da Figura 8.5, com pontos retangulares, mostra os resultados das execuções individuais de AG_DT com o conhecimento de todos os consumidores, representando o que seria um algoritmo do tipo re-otimização ou *restart*, já que este começa com o conhecimento completo de toda a instância, após a chegada do último consumidor. É possível observar que à medida que cresce o número de consumidores, aumentando o grau de dinamismo do problema, os resultados de AG_DT se aproximam de HGC_DT. Para a instância utilizada, R101, onde são muitos os veículos presentes na solução (o menor NV conhecido é 19), há boa probabilidade de aproveitamento das rotas geradas em R_{GLOBAL} , mesmo àquelas produzidas com um número menor de consumidores.

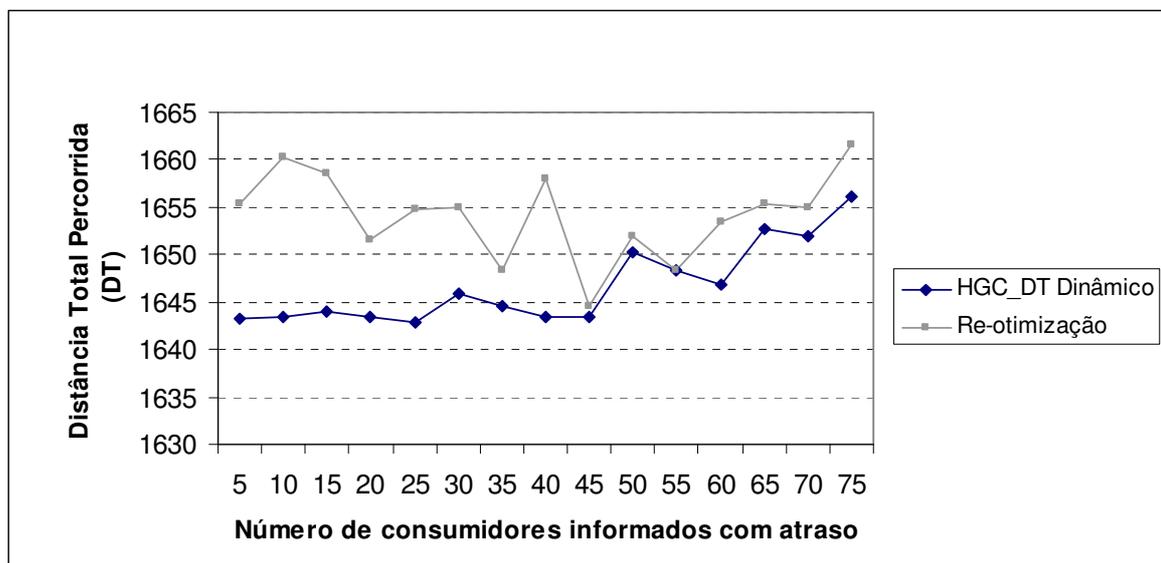


Figura 8.5 : Resultado médio de HGC_DT (3 execuções), variando o número de consumidores conhecidos após o início do processo de otimização; e resultado médio do melhor AG_DT, correspondendo a um algoritmo de re-otimização.

O grau de dinamismo pode ser alterado também modificando o momento em que os novos consumidores são conhecidos. Ainda utilizando a instância R101, foram inseridos 30 novos consumidores, agora alterando o momento inicial de inserção, conforme pode ser visto no gráfico da Figura 8.6. A curva de HGC_DT mostra que os resultados finais pioram à medida que a inclusão se aproxima do final do algoritmo. Já a curva de AG_DT, representando um algoritmo de re-otimização, conhecendo todos os consumidores, é mais próximo de uma constante, embora possa parecer haver uma ligeira tendência para cima, não se justifica, uma vez que todos os AG_DT foram executados uma única vez sob as mesmas condições. Novamente os resultados de HGC_DT superam um algoritmo de re-otimização utilizando AG_DT, que foi utilizado para produzir as rotas em HGC_DT.

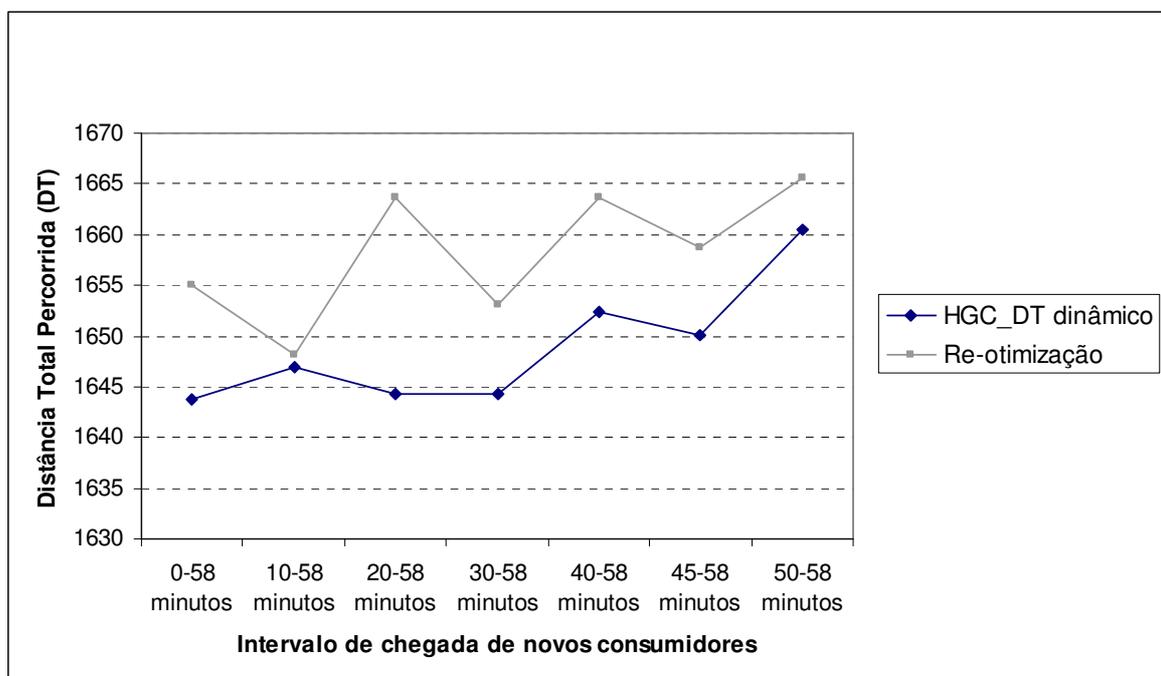


Figura 8.6: Resultados médios de HGC_DT e médios de AG_DT (re-otimização), com a variação do intervalo de inclusão de novos consumidores.

Um teste adicional foi feito com a instância R101 para verificar o aproveitamento de rotas geradas na fase inicial de execução de HGC_DT, onde não são conhecidos todos os consumidores. Para isso foi utilizada uma instância onde os últimos 12 consumidores de R101 foram omitidos inicialmente nos primeiros 15 minutos de execução. Estes consumidores foram inseridos um a um, em intervalos iguais até o final do tempo de execução, restando apenas 1 execução de AG_DT, ou seja, restando 2 minutos para o término do período total, que foi de 30 minutos. A Tabela 8.4 mostra o resultado otimizado

(PPC) no último instante antes do recebimento do primeiro novo consumidor (Fase I), utilizando as rotas já produzidas para o conjunto R_{GLOBAL} , e também o resultado final em 30 minutos (Fase II). É possível observar que há 13 rotas idênticas, 2 rotas com variação de apenas um consumidor, 2 rotas com variação de apenas dois consumidores, e 3 rotas com uma maior diferença. Este resultado mostra que rotas produzidas nas etapas iniciais são aproveitadas na solução do PPC, mesmo após a inserção de novos consumidores.

Tabela 8.4: Comparação das rotas produzidas aos 15 minutos de execução, Fase I, (sem conhecimento de todos os consumidores) e aquelas encontradas na solução final do problema R101, Fase II.

Fase I – Conhecimento parcial dos consumidores	Fase II – Solução final com 100% dos consumidores
63 64 49 48	63 64 49 48
52 6	52 6
14 44 38 43 13	14 44 38 43 13
31 88 7	31 88 7
65 71 9 66 1	65 71 9 66 1
36 47 19 8 46	36 47 19 8 46 17
40 53 26	40 53 26
33 81 50 68	33 81 50 68
2 21 73 41 56 4	2 21 73 41 56 4
28 29 78 34 35 77	28 29 78 34 35 77
27 69 30 51 20 32 70	27 69 30 51 20 32 70
12 76 79 3 54 24 80	12 76 79 3 54 24 80
39 23 67 55 25	39 23 67 55 25
62 11 10	62 11 90 10
72 75 22 74	72 75 22 74 58
5 61 85 37	5 83 61 85 37 93
82 18 84 60	45 82 18 84 60 89
45 83 16 86 17	95 98 16 86 91 100
59 42 15 87 57 58	92 42 15 87 57 97
	59 99 94 96

8.2.2 Cancelamento de consumidores

O segundo tipo de problema dinâmico tratado considera o cancelamento de consumidores. Embora no mundo real, os cancelamentos aconteçam ao mesmo tempo em que surgem outros novos consumidores, aqui foi tratado de forma isolada para permitir uma melhor avaliação do efeito destes cancelamentos na qualidade da solução de HGC_DT. Apesar de parecer mais fácil o tratamento de cancelamentos, uma vez que é simples e rápido retirar as

encomendas de uma solução estática otimizada, sempre haverá uma perda associada ao não conhecimento do problema real, uma vez que rotas foram modificadas pela necessidade de atender um consumidor que depois foi retirado.

Para permitir uma melhor comparação com os resultados estáticos, as instâncias de Solomon da classe R1 foram modificadas, inserindo novas encomendas, gerando dois tipos de instâncias R1A, com 130 consumidores e R1B com 150 consumidores. Os novos consumidores foram gerados aleatoriamente, com as mesmas características dos anteriores, 10 unidades de tempo para o tempo de serviço, 10 unidades de tempo para cada janela de tempo, demandas também aleatórias inferiores a 40 unidades e também posicionando os consumidores dentro da região já utilizada, neste caso no quadrante limitado pelos pontos (0,0) e (85,85).

O problema começa com o número total de consumidores 130 e 150 para as duas classes R1A e R1B, respectivamente. Também no intervalo de tempo entre 45 e 55 minutos, os consumidores não presentes na classe original R1 de Solomon (30 e 50 respectivamente) são eliminados.

Novamente somente 2 execuções e meia de AG_DT (após os 55 minutos) conhecem a instância final, com 100 consumidores. As demais rotas presentes em R_{GLOBAL} foram produzidas com um número superior de consumidores. Um procedimento adicional é necessário nas rotas de R_{GLOBAL} antes de se resolver o PPC, que é a eliminação de consumidores cancelados. Observando o modelo PCC, equações 3.1 a 3.3, observa-se que é necessário eliminar os consumidores cancelados de todas as rotas presentes neste conjunto. Isso permite que estas rotas se tornem novamente opções para a solução do problema, caso contrário, uma violação das restrições do modelo seria causada por tais rotas.

Novamente três execuções são realizadas utilizando dupla precisão, possibilitando comparar os resultados com aqueles do problema estático. Como a instância final é a mesma do problema estático, também é possível ter um conjunto de soluções desejadas, que neste caso é o resultado de HGC_DT do capítulo 6. A Tabela 8.5 apresenta os resultados para cada grau de dinamismo, 30 e 50 consumidores, eliminados posteriormente na fase final do período de otimização.

Tabela 8.5: Resultados da execução de HGC_DT retirando 30 consumidores no intervalo [45-55] minutos. O tempo de execução foi de 60 minutos, utilizando um computador utilizado Pentium IV, 2,4 MHz, 512 de RAM.

Instância	Melhor Estático		130 Consumidores						Distância do Estático (DT)	
	NV	DT	Dinâmico 1		Dinâmico 2		Dinâmico 3		NV	DT
R101	20	1642,870	20	1642,877	20	1646,184	20	1660,619	0	0,00%
R102	18	1472,620	19	1485,740	18	1476,491	18	1475,073	0	0,17%
R103	14	1213,620	14	1225,058	15	1226,553	15	1225,420	0	0,94%
R104	11	986,096	12	1011,629	12	1046,923	12	1016,588	1	2,59%
R105	15	1360,783	15	1371,151	15	1384,684	15	1384,954	0	0,76%
R106	13	1241,518	13	1253,443	13	1254,849	14	1274,720	0	0,96%
R107	11	1076,125	12	1098,848	13	1129,932	12	1099,821	1	2,11%
R108	10	948,573	11	994,061	11	973,796	10	976,375	1	2,66%
R109	13	1151,839	13	1168,680	13	1172,981	14	1190,490	0	1,46%
R110	12	1092,347	12	1129,451	12	1101,742	12	1123,497	0	0,86%
R111	12	1053,496	12	1073,922	11	1109,179	12	1100,028	0	1,94%
R112	10	960,675	11	1005,223	11	991,892	11	1013,127	1	3,25%
Total	159	14201	164	14460	164	14515	165	14541		1,83%

Tabela 8.6: Resultados da execução de HGC_DT retirando 50 consumidores no intervalo [45-55] minutos. O tempo de execução foi de 60 minutos, utilizando um computador utilizado Pentium IV, 2,4 MHz, 512 de RAM.

Instância	Melhor Estático		150 Consumidores						Distância do Estático (DT)	
	NV	DT	Dinâmico 1		Dinâmico 2		Dinâmico 3		NV	DT
R101	20	1642,870	19	1650,799	20	1647,841	20	1656,119	0	0,30%
R102	18	1472,620	18	1480,812	18	1474,108	18	1485,191	0	0,10%
R103	14	1213,620	15	1235,779	15	1244,402	14	1228,783	0	1,25%
R104	11	986,096	11	998,465	12	1015,004	11	1026,407	0	1,25%
R105	15	1360,783	16	1387,468	15	1383,150	16	1383,469	0	1,64%
R106	13	1241,518	13	1242,614	13	1250,368	13	1252,546	0	0,09%
R107	11	1076,125	12	1109,488	12	1122,677	12	1102,813	1	2,48%
R108	10	948,573	10	968,674	11	982,517	11	988,166	0	2,12%
R109	13	1151,839	14	1223,532	14	1216,046	13	1171,890	0	1,74%
R110	12	1092,347	13	1147,078	13	1140,321	12	1131,489	0	3,58%
R111	12	1053,496	13	1107,084	12	1094,178	12	1079,340	0	2,45%
R112	10	960,675	10	989,888	11	995,033	11	993,788	0	3,04%
Total		14200,56	164	14541,68	166	14565,65	163	14500,00		2,11%

Os resultados das tabelas Tabela 8.5 e Tabela 8.6, mostram que os valores em DT foram 1.83% e 2.11% acima do caso estático, para 30 e 50 consumidores retirados, respectivamente, para toda a classe R1. A inserção de 30 e 50 consumidores causaram um aumento em DT nos resultados de 1.42% e 2.35%, respectivamente. Enquanto para 30 consumidores a retirada parece ter sido mais crítica, para 50 consumidores os resultados da inclusão foram piores. Entretanto, como são muito próximos e alternados, não se pode concluir qual tipo de variação, retirada ou inclusão de consumidores, é mais crítico no dinamismo do PRVDJT.

8.3 Conclusões

Neste capítulo a utilização de HGC_DT para tratamento do problema dinâmico foi avaliada. Os resultados finais de HGC_DT são melhores (inferiores em DT) comparados com um algoritmo de re-otimização baseado no algoritmo AG_DT proposto. Uma vez que AG_DT foi desenvolvido e otimizado com AG_PARAM para produzir resultados tão bons quanto possíveis em tempo de execução de 2 minutos, este poderia ser considerado uma proposta atraente para uma estratégia de re-otimização no tratamento do problema dinâmico. Isso confirma a validade da proposta de HGC_DT para o problema dinâmico, pois foi possível constatar a utilidade do conjunto R_{GLOBAL} produzido previamente.

A vantagem da utilização das instâncias de Solomon, adaptadas ao cenário dinâmico proposto, possibilitou uma melhor avaliação dos resultados de HGC_DT para o PRVDJT, comparando-os com os resultados estáticos.

Trabalhos na literatura, utilizando também instâncias de Solomon modificadas, como a R101, apresentam resultados muito distantes daquelas correspondentes estáticas, não sendo possível avaliar se a diferença corresponde ao algoritmo ou se é inerentemente ao novo problema dinâmico criado. Na proposta atual, ao contrário, as diferenças dos valores em DT dos resultados estáticos para aqueles do problema dinâmico são 100% impossibilidade do algoritmo de tratar em tempo hábil a chegada das informações e aproveitando a busca já realizada.

Não foi utilizado um cenário que informações de novos consumidores chegam após os veículos já estarem em movimentação fora do depósito central. Embora este cenário seja muito comum e tão desejável quanto o que foi apresentado, observa-se que sob o aspecto de proposta de solução, ou seja, em termos de avaliação do algoritmo, eles são muito semelhantes. A diferença é que a partir do momento que os veículos saem do depósito, problema inicial (PRVJT) se transforma no Problema de Coleta e Entrega com Janela de Tempo (PCEJT), onde os veículos têm origens diferentes. No cenário utilizado, quando os novos consumidores são informados, os veículos ainda estarão no depósito, mas as rotas otimizadas já encontradas no processo de busca necessitam serem alteradas para contemplá-los.

Por outro lado, se os veículos já partiram, os consumidores já visitados não são considerados no problema, mas assim como no cenário aqui tratado, existem rotas atuais já otimizadas (correspondente aos consumidores ainda não visitados), que devem ser alteradas para incluir os novos consumidores. Os consumidores já visitados não dificultam o trabalho do algoritmo.

Para um algoritmo que trate o problema dinâmico, é de suma importância aproveitar de alguma forma o tempo já utilizado até a chegada de novas informações. A grande vantagem do cenário escolhido é que a solução final desejada é o resultado já muito trabalhado e conhecido na literatura do problema estático. Isso possibilitou mostrar que a heurística proposta é competitiva para o PRVDJT.

Capítulo 9

Conclusões

Este capítulo resume as principais contribuições deste trabalho e apresenta as conclusões finais.

9.1 Comparação dos resultados de HGC_DT com as soluções exatas da literatura

Pelo que se tem conhecimento, os resultados de uma heurística foram comparados pela primeira vez com aqueles encontrados por métodos exatos, considerando as instâncias de Solomon [SM87] sob os mesmos critérios de truncamento e função objetivo. Isso possibilitou avaliar melhor as vantagens e desvantagens da utilização de uma heurística. A principal vantagem da heurística ainda continua sendo a garantia de retorno em um tempo aceitável. No entanto, até então se conhecia pouco o preço a ser pago por esta vantagem em termos de distância dos resultados ótimos. Para o PRVJT foi possível constatar que a média dos resultados de HGC_DT e, principalmente, os melhores entre três execuções estão muito próximos dos resultados ótimos. Por outro lado, o principal desafio no projeto de uma heurística está na tentativa de obter uma melhor robustez no algoritmo, sendo capaz de gerar resultados de qualidade próximos em diferentes execuções e utilizando diferentes instâncias. A estratégia híbrida utilizada em HGC_DT, fazendo uso do algoritmo genético e de um pacote de programação linear inteira (GLPK) para resolver um PPC, possibilitou um desvio padrão aceitável nos resultados e também permitiu alcançar vários resultados ótimos conhecidos da literatura. Considerando que para as instâncias de 100 consumidores de Solomon ainda existem muitos problemas em aberto, mesmo depois da utilização de modernas técnicas de planos de corte e geração de colunas, HGC_DT se torna uma boa opção para uso, pois os resultados globais ficaram somente a 0.29% dos ótimos conhecidos. Adicionalmente o tempo de execução de HGC_DT é bem menor que aqueles empregados nos métodos exatos.

9.2 Resultados de HGC_DT com dupla precisão

Os resultados alcançados por HGC_DT utilizando dupla precisão, minimizando distância percorrida foram muito expressivos. Embora a redução de DT não tenha sido tão trabalhada quanto NV na literatura, o que possibilitou alcançar ou ultrapassar 53, das 56 instâncias de Solomon, estes novos resultados são de extrema importância, haja visto que a distância total percorrida parece ser tão importante quanto o número de veículos para empresas do mundo real. Por isso, ainda há espaço na literatura para a propostas de novas heurísticas minimizando DT. Acredita-se que o resultado de HGC_DT poderá ser ultrapassado utilizando mais tempo em cada execução do algoritmo genético para as instâncias com poucos veículos, pois nestes casos o ganho proporcionado pelo PPC foi nulo em alguns problemas, como alguns da classe R2 e RC2. Nestes casos poucos resultados ótimos são conhecidos e como não foi trabalhado um limite inferior de qualidade, não se sabe quanto ainda há de espaço para melhorias.

9.3 Melhorando a distância percorrida em HGC_NV

Embora HGC_DT e AG_DT tenham sido utilizados internamente em HGC_NV, os resultados DT após o número de veículos ter sido minimizado previamente (objetivo principal de HGC_DT) não foram tão bons quanto no caso anterior, onde o número de veículos era livre. Uma das possíveis causas desses resultados poderá ser a falta de diversidade na população deixada por AG_NV para AG_DT, que é utilizada sempre, a cada nova execução de AG_DT em HGC_DT. Resolver este problema de diversidade não é simples, pois esta população representa a última geração de AG_NV, onde uma maior diversificação poderá piorar a qualidade do objetivo principal nesta etapa, que é a de eliminação de veículos. Talvez uma melhor opção seria trabalhar os operadores de AG_DT, para que possam realizar uma etapa inicial de diversificação da população deixada por AG_NV, mantendo o número de veículos minimizado. Este papel é realizado pelo PFIH estocástico proposto, mas que deixa de ser empregado em AG_DT por necessidade de manter NV reduzido.

9.4 Estabelecimento de “Problemas Testes” para o PRVDJT

Os problemas testes propostos para transformar os problemas estáticos de Solomon em dinâmicos têm como principal vantagem a de possibilitar que toda e qualquer diferença de resultado em DT, comparando com os resultados estáticos já bem conhecidos, seja atribuída as características do algoritmo proposto. Ao contrário, se consumidores são inseridos probabilisticamente após outros já terem sido visitados, têm-se um problema onde não existe resultado esperado na literatura. Adicionalmente, sob o ponto de vista do algoritmo, como mostrado no capítulo 8, os dois problemas apresentam o mesmo grau de dificuldade, qual seja o de tratar a chegada de novos consumidores e aproveitar de forma eficiente rotas já otimizadas previamente, uma vez que decisões de roteamento devem ser tomadas em curto período de tempo, aproveitando ao máximo a busca já realizada.

9.5 Distância Percorrida *versus* Número de Veículos

Larsen [La99], sugeriu a padronização das propostas de solução para o PRVJT estático, utilizando DT como único objetivo e cálculos com inteiros, possibilitando uma facilidade de comparação. Realmente seria de extrema utilidade para os pesquisadores interessados neste problema a padronização quanto aos cálculos utilizados. Já quanto a função objetivo, não se pode fechar definitivamente em um objetivo prioritário. Existem diversos cenários no mundo real onde NV é prioridade, e tantos outros onde DT deve ser considerada mais importante. Conseqüentemente, o mais interessante para a comunidade internacional seria abordar estes e outros objetivos do PRVJT nos mesmos trabalhos, pois uma solução capaz de lidar somente com um destes teria uma aplicação no mundo real muito restrita.

9.6 Limites Inferiores para HGC_DT

Uma característica desejável para uma solução heurística é a informação da distância máxima que sua solução está do ótimo, ou seja, o limite inferior (LI). Nesta tese, uma atenção devida ao tema não foi dada devido à qualidade razoável do LI resultante da

simples relaxação linear no modelo compacto apresentado no capítulo 2. Diversos autores têm mostrado que o LI da relaxação linear, para as instâncias de Solomon, apresenta uma qualidade próxima a diversas outras técnicas [La99] [KM97]. Embora pequenas diferenças no LI podem trazer grandes diferenças no desempenho de métodos exatos, este é menos importante no contexto das heurísticas, foco principal deste trabalho. Entretanto, o cálculo do LI seria um importante assunto na continuidade dos trabalhos aqui apresentados.

9.7 Conclusões Finais

A proposta de uma solução híbrida, tratando o PRVJT, possibilitou resultados inéditos em DT para o problema estático. Uma ampla gama de operadores de busca possibilitou gerar diferentes mínimos locais de qualidade razoável em intervalos de curta duração através de AG_DT. Adicionalmente, o tratamento do problema em fases distintas, ora tratando o problema completo e diversificando a busca, ora tratando o problema reduzido e intensificando a busca no melhoramento das rotas já encontradas, possibilitou a geração de um conjunto de rotas R_{GLOBAL} suficientemente bom, para gerar resultados de boa qualidade no PPC final.

Quanto ao objetivo minimização do número de veículos no PRVJT, uma grande contribuição foi a proposta de novos critérios de avaliação de indivíduos solução, possibilitando identificar àqueles onde há maior facilidade de redução de mais um veículo. Também foi confirmado a necessidade de tratar os objetivos concorrentes DT e NV em fases distintas, pois foi necessário considerar o critério DT em sétima prioridade, para que este objetivo não provocasse a estagnação da solução em um mínimo local prematuro em NV.

Outra grande vantagem da heurística proposta foi a possibilidade de tratar também o problema dinâmico, sendo necessário pouquíssimas alterações. As diversas propostas anteriores para o tratamento do PRVJTD estão mais próximas das heurísticas clássicas de aperfeiçoamento contínuo, do que das metaheurísticas tão consagradas para o problema estático. Já HGC_DT utiliza o grande avanço e conhecimento da literatura nas modernas metaheurísticas, como o AG, para tratar este problema. Adicionalmente, os seus resultados

são validados com os problemas estáticos, transformando-a em uma proposta competitiva para o PRVDJT.

Dentre os principais pontos a serem considerados na continuação do trabalho para o problema dinâmico, está a comparação de HGC_DT com outros resultados dinâmicos da literatura. Atualmente, como estes problemas vêm sendo apresentados e propostos, esta comparação é muito difícil e relativa, pois estão envolvidas variáveis estocásticas, fazendo com que cada execução trate um problema isolado, impossibilitando uma comparação direta. No entanto, cabe uma melhor avaliação nesse sentido.

Referências Bibliográficas

- [AEO+99] R.K. Ahuja, O. Ergun, J.B. Orlin, and A.B. Punnen. “*A survey of very large-scale neighborhood search techniques*”. Working Paper, Sloan School of Management, MIT, (1999).
- [AMS89] Y. Agarwal, K. Mathur and H.M. Salkin, “*A set-partitioning-based exact algorithm for the vehicle routing problem*”, *Networks* 19, 731-740, (1989).
- [AM04] G.B. Alvarenga e G. R. Mateus, “*A Two-phase Genetic and Set Partitioning Approach for the Vehicle Routing Problem with Time Windows*”. Fourth International Conference on Hybrid Intelligent Systems, Kitakyushu, Japan (2004).
- [BBB01] J. Berger, M. Barkaoui and O. Bräysy, “*A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows*”. Working paper, Defense Research Establishment Valcartier, Canada, (2001).
- [BG01] O. Bräysy and M. Gendreau. “*Metaheuristics for the Vehicle Routing Problem with Time Windows*”. Internal Report STF42 A01025, SINTEF Applied Mathematics, Department of Optimisation, Norway (2001).
- [Brä01] O. Bräysy, “*A Reactive Variable Neighborhood Search Algorithm for the Vehicle Routing Problem with Time Windows*”. To appear in *INFORMS Journal on Computing*, (2001).
- [BC98] J.E. Beasley and P.C. Chu “*Constraint handling in genetic algorithms: the set partitioning problem*”. *Journal of Heuristics*, vol. 4, 323-357 (1998).
- [Ber92] D.J. Bertsimas, “*Vehicle Routing Problem with Stochastic Demand*”. *Operations Research*, **40:3**, 574-585, (1992).
- [Bod90] L.D. Bodin, “*Twenty years of routing and scheduling*”, *Operations Research*, 38, 571-579, (1990).
- [BGA+83] L.D. Bodin, B.L. Golden, A.A. Assad and M. Ball, “*Routing and Scheduling of Vehicle and Crews, the State of the Art*”, *Computers and Operational Research*, 10, pp. 69-211. (1983).

- [Box57] Box, G. E. P., “*Evolutionary Operation: a Method of increasing industrial productivity*”. Applied Statistics, Vol. **6**, 81 -101, (1957).
- [BH04] R. Bent and P. Van Hentenrych. “*A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows*” Transportation Science, 38(4): 515-530, (2004).
- [Bra98] J. Brandão. “*Metaheuristic for the vehicle routing problem with time windows*”. S. Voss, S. Martello, I.H. Osman, and C. Roucairol, editors, *Meta Heuristics: Advances and Trends in Local Search Paradigms for Optimisation*, pages 19-36. Kluwer, Boston, (1998).
- [BJN98] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance, “*Branch-and-price: Column generation for solving huge integer programs*”, Operations Research, **46**, 316-329, (1998).
- [BKY02] J.F. Bard, G. Kontoravdis and G. Yu, “*A Branch-and-Cut Procedure for the Vehicle Routing Problem with Time Windows*,” Transportation Science 36(2), 250-269, (2002).
- [BS95] J.B. Bramel and D. Simchi-Levi, “*A Location Based Heuristics for General Routing Problems*”. Operations Research, **43**, 649-660, (1995).
- [BSL96] D. Bertsimas and D. Simchi-Levi, “*A new generation of vehicle routing research: robust algorithms, addressing uncertainty*”. Operations Research, vol. 44, pp. 286-304, (1996).
- [BW91] Jr. J. L. Blanton and R. L. Wainwright. “*Multiple vehicle routing with time and capacity constraints using genetic algorithms*”. Proceedings of the Fourth International Conference on Genetic Algorithm, 452-459, (1991).
- [NB84] N. Christophides and J. Beasley. “*The period routing problem*”. Networks, vol. 14, pp. 237-256 (1984).
- [Cas93] N. Castro. “*Intermodalidade, Intramodalidade e o Transporte de Longa Distância no Brasil*”. Rio de Janeiro: IPEA, Planejamento e Políticas Públicas, (1993).
- [CC02] Z. J. Czech and P. Czarnas, “*A Parallel Simulated Annealing for the Vehicle*

- Routing Problem with Time Windows*," Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, , 376-383, (2002).
- [CDD+01] J. F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon and F. Soumis. "*The VRP with Time Windows*". ACM, Society for Industrial and Applied Mathematics, 157 – 193, (2001).
- [CLM00] J.-F. Cordeau, G. Laporte, and A. Mercier, "*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).
- [CMT79] N. Christofides, A. Mingozzi and P. Toth, "*The Vehicle Routing Problem*". Combinatorial Optimization. Wiley, Chichester. (1979).
- [Coo71] S. A. Cook. "*The Complexity of Theorem Proving Procedures*". Proceedings Third Annual ACM Symposium on Theory of Computing, pp 151-158, (1971).
- [CR78] T.M. Cook and R.A. Russel, "*A Simulation and Statistical Analysis of Stochastic Vehicle Routing with Timing Constraints*", Decision Science, **9**, pp. 673-687, (1978).
- [CR96] W.-C. Chiang and R.A. Russell. "*Simulated annealing metaheuristics for the vehicle routing problem with time windows*". Annals of Operations Research, 63:3-27, (1996).
- [CR99] W. Cook and J. L. Rich, "*A parallel cutting plane algorithm for the vehicle routing problem with time windows*". Working Paper, Computational and Applied Mathematics, Rice University, Houston, TX, (1999).
- [CW64] G. Clarck and J.W.Wright, "*Scheduling of Vehicle from a Central Depot to a Number of Delivery Points*", Operational Research 12, 568-581, (1964).
- [Car95] W. B. Carlton. "*A tabu search approach to the general vehicle routing problem*". Ph.D. Thesis, The University of Texas at Austin, Austin (1995).
- [CW01] R. Cordone and R. Wolfler-Calvo, "*A Heuristic for the Vehicle Routing Problem with Time Windows*", Journal of Heuristics 7, 107–129, (2001).

- [Dar59] Darwin, C., 1859. "*On The Origin of Species*". 1st edition, Harvard University Press, MA.
- [DDS+95] J. Desrosiers, Y. Dumas, M. Solomon, and F. Soumis, "*Handbooks in Operations Research and Management Science*". Vol. 8, pp. 35-140. Elsevier Science, Amsterdam (1995).
- [Des88] M. Desrochers. "*An algorithm for the shortest path problem with resource constraints*". Technical Report G-88-27, GERAD, _Ecole des Hautes _Etudes Commerciales, Universite de Montreal, (1988).
- [DR59] G. B. Dantzig and R.H. Ramser, "*The Truck Dispatching Problem*", *Management Science*, **6**, 80 (1959).
- [DT86] M. Dror and P. Trudeau, "*Stochastic Vehicle Routing with Modified Savings Algorithm*", *European Journal of Operational Research*, **23**, pp. 228-235, (1986).
- [DSD+84] J. Desrosiers, F. Soumis and M. Desrochers. "*Routing with time windows by column generation*". *Networks*, **14**(4) : 545-565 (1984).
- [DW60] G.B. Dantzig and P. Wolfe, "*Decomposition principle for linear programs*", *Operational Research*, **8**, 101-111, (1960).
- [Fra57] A. S., Fraser, "*Simulation of Genetic System by Automatic Digital Computers*". *Australian Journal of Biological Science*, Vol. 10, 484-491, (1957).
- [FJM97] M. L. Fisher, K. O. Jornsteen, and O. B. G. Madsen. "*Vehicle routing with time windows: Two optimization algorithms*". *Operations Research*, **45**(3):488-492, (1997).
- [FJW81] M. Fisher, R. Jaikumar, and L. V. Wassenhove. "*An generalized assignment heuristic for vehicle routing*". *Networks*, vol. 11, pp 109-124 (1981).
- [FLP2004] R. Fukasawa, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa and R.F. Werneck, "*Robust branch-and-cut-and-price for the capacitated vehicle routing problem*". Proceedings of the tenth Integer Programming and Combinatorial Optimization conference IPCO'04, New York, *Lecture Notes in*

- Computer Science*, Vol.3064 pp.1-15, (2004).
- [FOW66] L. J. Fogel, A.J. Owens and M.J. Walsh. “*Artificial Intelligence Through Simulated Evolution*”. New York: Wiley Publishing. (1966).
- [Gol89] D. E. Goldberg. “*Genetic Algorithms in Search, Optimization, and Machine Learning*”. Addison-Wesley, New York, (1989).
- [GDD+95] S. Gelinas, M. Desrochers, J. Desrosiers, and M. M. Solomon. “*A new branching strategy for time constrained routing problems with application to backhauling*”. *Annals of Operations Research*, 61:91-109 (1995).
- [Glo86] F. Glover. “*Future paths for integer programming and links to artificial intelligence*”. *Computers & Operations Research*, 13:533–549, (1986).
- [GG61] Gilmore, P.C. and R.E. Gomory, “*A Linear Programming Approach to the Cutting Stock Problem*”, *Operations Research*, 9, 849-859, (1961).
- [GH99] H. Gehring and J. Homberger (1999), “*A Parallel Hybrid Evolutionary Metaheuristic for the Vehicle Routing Problem with Time Windows*”, In Proceedings of EUROGEN99 - Short Course on Evolutionary Algorithms in Engineering and Computer Science, Reports of the Department of Mathematical Information Technology, Series A. Collections, No. A 2/1999, K. Miettinen, M. Mäkelä and J. Toivanen (eds.). University of Jyväskylä, Jyväskylä, 57–64 (1999).
- [GH01] Gehring, H. and J. Homberger, “*Parallelization of a Two-Phase Metaheuristic for Routing Problems with Time Windows*”, *Asia-Pacific Journal of Operational Research* 18, 35–47, (2001).
- [GP97] M. Gendreau and J.-Y. Potvin, “*Dynamic vehicle routing and dispatching*”, Tech. Rep. CRT 97-38, Centre de Recherche sur les Transport, Université de Montréal, November (1997).
- [GS96] Gondzio J. and R. Sarkissian, “*Column Generation with a Primal-Dual Method*”, Logilab Technical Report 96.6, Department of Management Studies, University of Geneva, Switzerland, (1996).
- [GTA99] L. M. Gambardella, E. Taillard, and G. Agazzi, “*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), 63-76, McGraw-Hill, London (1999).
- [HAL92] K. Halse. "*Modeling and Solving Complex Vehicle Routing Problems*". PhD thesis, Department for Mathematical Modeling, Technical University of Denmark, (1992).
- [HG99] J. Homberger and H. Gehring, "*Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows*". *INFOR*, VOL. 37, 297-318, (1999).
- [Hol75] Holland, J. H. "*Adaptation in Natural and Artificial System*". Ann Arbor, Michigan: The University of Michigan Press (1975).
- [HIO95] C.A. Hjorring. "*The vehicle routing problem and local search metaheuristics*." PhD thesis, University of Auckland, Auckland, New Zealand, (1995).
- [KDM+99] N. Kohl, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis, "*2-Path Cuts for the Vehicle Routing Problem with Time Windows*" *Transportation Science*, Vol. 33 (1), 101-116 (1999).
- [KLM01] B. Kallehauge, J. Larsen and O.B.G. Madsen. "*Lagrangian Duality Applied on Vehicle Routing with Time Windows - Experimental Results*." Technical Report IMM-REP-2001-9, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark (2001).
- [KM97] N. Kohl and O. B. G. Madsen. "*An optimization algorithm for the vehicle routing problem with time windows based on lagrangean relaxation*". *Operations Research*, 45(3):395-406, May-June (1997).
- [Koh95] Niklas Kohl. "*Exact methods for Time Constained Routing and Related Scheduling Problems*". PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, (1995).
- [KPS98] P. Kilby, P. Prosser and P. Shaw. "*Dynamic VRPs: A Study of Scenarios*". Report APES-06-1998 (1998).
- [KRT87] A. W. J. Kolen, A. H. G. Rinnooy Kaan, and H. W. J. M. Trienekens. "*Vehicle*

- routing with time windows*". Operations Research, 35(2):266-273, March-April (1987).
- [La99] J. Larsen. "*Parellellization 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).
- [Lar00] A. Larsen, "*The Dynamic Vehicle Routing*", Phd. Thesis. Department of Mathematical Modelling (IMM) at the Techical University of Denmark (DTU), (2000).
- [LMR96] K. Lund, O. B. G. Madsen, and J. M. Rygaard, "*Vehicle Routing Problems with Varying Degrees of Dynamism*". Technical report, IMM, The Department of Mathematical Modelling, Technical University of Denmark, (1996).
- [LS99] F.-H. Liu and Shen S.-Y. "*A Route-Neighborhood-based Metaheuristic for Vehicle Routing Problem with Time Windows*". European Journal of Operational Research 118, 485-504, (1999).
- [LGP+00] G. Laporte, M. Gendreau, J-Y Potvin and F Semet, "*Classical and Modern Heuristics for the Vehicle Routing Problem*", Intl. Trans. in Operational Research, 7, pp 285-300, (2000).
- [LLS98] Q. Z. Liu, H.C. Lau, Dennis Seah, Sidney Chong, "*An Efficient Near-exact Algorithm for Large-Scale Vehicle Routing with Time Windows*", Proc. 5th World Congress on ITS, Korea, (1998).
- [LLH03] H. Li, A. Lim, and J. Huang, "*Local Search with Annealing-like Restarts to Solve the VRPTW*," European Journal of Operational Research 150, 115-127, (2003).
- [Lin65] S. Lin, "*Computer Solution of the Traveling Salesman Problem*". Bell System Technical Journal. 44, 2245-2269 (1965).
- [Lev44] K. Levenberg. "*A method for the solution of certain problems in least squares*". Quart. Appl. Math. 2, 164-168, (1944).
- [LO95] G. Laporte and I.H. Osmar. "*Routing Problems: A Bibliography*". Annals of Operational Research, 61,227-262, (1995).

- [Lin65] S. Lin. “*Computer solutions of the traveling salesman problem*”. Bell Systems Technical Journal, vol. 44, pp. 2245–2269, (1965).
- [Mar63] D. Marquardt. “*An algorithm for Least Squares Estimation on Nonlinear Parameters*”. SIAM J. Appl. Math. 11, 431-441, (1963).
- [Mes02] D. Mester, “*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).
- [MJ76] R.H. Mole and S.R. Jameson, “*A Sequential Route-building Algorithm Employing a Generalized Savings Criterion*”. Operational Research Quarterly **27**, 503-511. (1976).
- [Mil76] P. Miliotis, “*Integer Programming Approaches to the Travelling Salesman Problem*” . Mathematical Programming **10**, 367-378 (1976).
- [MM91] M. Mitchell. “*Review of Genetic Algorithms*”. Relatório Técnico. Santa Fé Intitute.(1991).
- [OR76] I. Or. “*Travelling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Blood-Banking*”. PhD thesis, Department of Industrial Engineering and Management Sciences, Northwest University, Evanston, IL., (1976).
- [OC94] I. H. Osman and N. Christofides. “*Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search*”. International Transactions in Operational Research 1, 3, (1994).
- [Pas80] H. Psaraftis. “*A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem*”. Transportation Science, vol. 14, pp. 130-154, (1980).
- [Pas95] H. Psaraftis. “*Dynamic Vehicle Routing: Status and Prospects*”. Annals of Operations Research, 61, pp 143-164, (1995).
- [Pas98] H. Psaraftis. “*Dynamic Vehicle Routing*”. Gonden, B.L. and A.A. Assad, Vehicle Routing: Methods and Studies, Elsevier (North-Holland), Amsterdam, pp 223-248, (1998).

- [PKG+96] J.-Y. Potvin, T. Kervahut, B. L. Garcia, and J.-M. Rousseau. “*The vehicle routing problem with time Windows – Part I: Tabu search*”. *INFORMS Journal on Computing*, 8:158-164, (1996).
- [PR95] J.-Y. Potvin and J.-M. Rousseau. “*An exchange heuristic for routing problems with time windows*”. *Journal of the Operations Research Society*, 46:1433–1446, (1995).
- [PSA80] H.N. Psaraftis. “*A dynamic programming solution to single vehicle many-to-many immediate request dial-a-ride problem*”, *Transportation Science*, Vol.14, No.2, May (1980).
- [PSA83] H.N. Psaraftis. “*An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows*”, *Transportation Science*, Vol.17, No.3, August (1983).
- [Rec73] Rechenberg, I., “*Evolutions strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*”. Frommann-Holzboog, Stuttgart. (1973).
- [SPR+95] Y. Shen, J.Y. Potvin, J. M. Rousseau and S. Roy, “*A Computer Assistant for Vehicle Dispatching with Learning Capabilities*”, *Annals of Operations Research*, 61, 189-211 (1995).
- [RGP99] L. M. Rousseau, M. Gendreau and G. Pesant. “*Using Constraint-based Operators with Variable Neighborhood Search to Solve the Vehicle Routing Problem with Time Windows*” Presented at the CP-AI-OR'99 Workshop, February 25.-26., University of Ferrara, Italy, (1999).
- [RR95] R. A. Russell. “*Hybrid heuristics for the vehicle routing problem with time windows*”. *Transportation Science*, 29, 156-166, (1995).
- [RT95] Y. Rochat and E.D. Taillard, “*Probabilistic Diversification and Intensification in Local Search for Vehicle Routing*”. *Journal of Heuristics* 1, 147-167, (1995).
- [SM87] M.M Solomon. “*Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints*”, *Operational Research*, 35, pp. 254-265 (1987).

- [SD88] M. Solomon and J. Desrosiers, "*Time window constrained routing and scheduling problems: a survey*", *Transportation Science*, 22, pp. 1-13 (1988).
- [SOL94] M. Sol. "*Column Generation Techniques for Pickup and Delivery Problems*". Phd. Thesis, Technische Universiteit Eindhoven (1994).
- [Sav92] M.W.P. Savelsbergh. "*The vehicle routing problem with time window: minimizing route duration*", *ORSA Journal on Computing* 4, 146-154, (1992).
- [SS95] M.W.P. Savelsbergh and M. Sol, "*The general pickup and delivery problem*", *Transportation Research*, 29, 17-29, (1995).
- [SS98] M.W.P. Savelsbergh and M. Sol, "*DRIVE: Dynamic routing of independent vehicles*", *Operations Research*, 46, 474-490, (1998).
- [Sha98] P. Shaw, "*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).
- [TAI94] E. Taillard. "*Parallel iterative search methods for vehicle-routing problems*". *Networks*, vol. 23(8), pp. 661-673 (1994).
- [TBG+97] E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J.Y. Potvin, "*A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows*," *Transportation Science*, 31, 170-186, (1997).
- [THA99] S. R. Thangiah. "*A hybrid genetic algorithm, simulated annealing and tabu search heuristic for vehicle routing problems with time window*", pages 348-383. CRC Press, Boca Raton, FL, (1999).
- [The98] The Danish Ministry of Transport. *Tra_kredegr_else 1997*, January (1998).
- [TLZ+01] K.C. Tan, L.H. Lee, Q.L. Zhu and K. Ou, "*Heuristic methods for vehicle routing problem with time windows*" *Artificial Intelligence in Engineering*, 15, 281-295, (2001).
- [TOS94] S. R. Thangiah, I. H. Osman, T. Sun. "*Hybrid Genetic Algorithm Simulated Annealing and Tabu Search Methods for Vehicle Routing Problem with Time*

- Windows*". Technical Report 27, Computer Science Department, Slippery Rock University, (1994).
- [Vas94] J. A. Vasconcelos, "*Optimisation de Forme des Structures Electromagnétiques*", Phd Thesis. Ecole Centrale de Lyon, ECL, França, (1994).
- [VMO+99] S. Voß, S. Martello, I.H Osman, and C. Roucairol, editors. "*Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*". Kluwer, Boston (1999).
- [XCR+01] H. Xu, Z.-L. Chen, S. Rajagopal, and S. Arunapuram "*Solving a practical pickup and delivery problem*". Tech. Rep., Department of Systems Engineering, University of Pennsylvania, (2001).
- [YG80] J.R. Yee and B.L. Gonden, "*A Note on Determining Operating Strategies for Probabilistic Vehicle Routing*", Naval Research Logistics Quarterly, 27, pp. 159-163, (1980).
- [YJM04] J. Yang, P. Jaillet and H. S. Mahmassani, "*Real-time Multi-vehicle Truckload Pick-up and Delivery Problems*". Transportation Science, 38:2, pp. 135-148, (2004).
- [ZO00] K.Q. Zhu and K. L. Ong, "*A Reactive Method for Real Time Dynamic Vehicle Routing Problem with Time Window*". 12th IEEE International Conference on Tools with Artificial Intelligence, pp. 176-181, (2000).