

6

Comparações entre as Heurísticas GRASP e VNS

O objetivo desse capítulo é comparar a heurística GRASP com filtro e reconexão por caminhos, apresentada no Capítulo 4, com a heurística VNS com filtro e reconexão por caminhos, apresentada no Capítulo 5, em termos da qualidade das soluções encontradas e dos tempos de processamento obtidos para diversas classes de instâncias. Três diferentes tipos de comparações serão realizadas.

Primeiramente, as heurísticas serão comparadas obtendo-se a qualidade das soluções encontradas em três diferentes tempos de processamento para cada instância, possibilitando estudar o comportamento dos algoritmos em tempos distintos de execução.

Em seguida, gráficos de distribuição de probabilidade empírica do tempo gasto para encontrar um valor alvo serão utilizados para a comparação das heurísticas.

O modelo baseado em árvore geradora mínima restrita por grau para o problema das p -medianas conectadas, apresentado na Seção 2.4.2, possibilitou a obtenção de soluções ótimas para algumas instâncias testadas. Assim, por último, comparações em termos absolutos serão realizadas entre as heurísticas nas instâncias em que o ótimo é conhecido. Para as instâncias em que não se conhece o ótimo, a melhor solução encontrada pelas heurísticas GRASP e VNS será a melhor solução conhecida para aquela instância. Essas comparações mostraram a necessidade de incorporar um passo adicional em ambos os algoritmos. O passo de pós-otimização consiste em executar um algoritmo ótimo para o problema de Steiner tendo como entrada as p facilidades abertas da melhor solução encontrada pelas heurísticas, após todas as iterações terem sido realizadas.

Este capítulo está organizado como se segue. A Seção 6.1 mostra o ambiente de teste e as instâncias utilizadas na comparação das heurísticas. A Seção 6.2 compara a qualidade das soluções encontradas pelas heurísticas fixando-se três diferentes tempos de processamento para cada instância. A Seção 6.3 apresenta gráficos de distribuição de probabilidade do tempo gasto para atingir o valor alvo para ambas as heurísticas. A Seção 6.4 mostra as comparações em termos absolutos entre as heurísticas para todas as instâncias e descreve o passo de pós-otimização, assim como seus resultados computacionais. Por

último, a Seção 6.5 apresenta as conclusões e considerações finais do capítulo.

6.1

Ambiente de Teste e Instâncias Utilizadas

As heurísticas GRASP e VNS foram implementadas em C com o parâmetro de otimização -O3 e executadas em uma máquina Pentium IV 3.2 GHz com 1 Gbyte de memória RAM sob o sistema operacional Linux RedHat 9.0. O gerador de números aleatórios utilizado foi o de Matsumoto e Nishimura [53].

Para a execução do modelo baseado em árvore geradora mínima restrita por grau, utilizou-se a mesma máquina e a versão 8.0 do resolvidor de programação linear inteira ILOG CPLEX. A restrição de eliminação de sub-rotas foi substituída pela restrição de Miller-Tucker-Zemlin utilizada no problema do caixeiro viajante [38].

Para a comparação das heurísticas, dois grupos de instâncias serão utilizados. O primeiro grupo são as instâncias proporcionais das classes ORM_P, GRM_P e SLM_P ($w = 2$, $w = 5$ e $w = 10$), apresentadas na Seção 3.3.4, totalizando-se 189 problemas. O segundo grupo, denominado de instâncias não proporcionais, foi obtido a partir do grupo de instâncias proporcionais ORM_P e GRM_P ($w = 2$) da seguinte maneira. Dado um grafo $G = (V, E)$ com custos associados às arestas, o custo de servir cada usuário i a partir de cada facilidade j é dado pelo valor do caminho mais curto d_{ij} no grafo G , como no grupo de instâncias proporcionais. Porém, o custo de instalação c_{ij} de cada aresta é escolhido aleatoriamente no intervalo $[(w \times a_{ij}) - (w \times a_{ij})/2, (w \times a_{ij}) + (w \times a_{ij})/2]$, onde a_{ij} é o custo associado à aresta (i, j) nas instâncias proporcionais ORM_P e GRM_P com $w = 2$. As classes de instâncias não-proporcionais serão denominadas de ORM_NP e GRM_NP, totalizando 60 problemas.

6.2

Tempo de Processamento Limitado

Nesta seção, as heurísticas serão comparadas com base nas soluções encontradas em três tempos de processamento distintos para cada instância: um tempo menor de execução (tp), um tempo intermediário de execução (ti) e um tempo maior de execução (tg). Como ambas as heurísticas utilizam a mesma busca local, adotou-se o seguinte procedimento para a fixação dos tempos. Para cada instância das classes proporcionais ORM_P, GRM_P e SLM_P ($w = 2$, $w = 5$ e $w = 10$), utilizou-se o tempo médio de processamento de cada iteração da busca local concatenada em 15 execuções realizadas, obtido dos experimentos no Capítulo 3 (medida tam). Fixou-se então $tp = 10 \times tam$, $ti = 40 \times tam$ e $tg = 70 \times tam$. Os tempos tp , ti e tg das classes ORM_NP

e GRM_NP serão os mesmos das respectivas classes proporcionais ORM_P e GRM_P ($w = 2$).

Como exemplo, o valor de tam para a instância ORM_P19 ($w = 2$) é 6,28 segundos. Assim, executam-se as heurísticas para essa instância com a mesma semente do gerador de números aleatórios fixando-se os seguintes tempos: $tp = 62,80$ segundos, $ti = 251,20$ segundos e $tg = 439,60$ segundos. Para as instâncias em que o valor de tam é muito pequeno, fixou-se $tam = 1,00$ segundo. Por exemplo, o valor de tam para a instância ORM_P1 ($w = 2$) é 0,003 segundos. Assim, executam-se as heurísticas para essa instância com a mesma semente com os tempos: $tp = 10,00$ segundos, $ti = 40,00$ segundos e $tg = 70,00$ segundos.

Para cada instância, heurística e tempo de processamento, cinco execuções diferentes foram realizadas. As Tabelas 6.1 a 6.9 mostram os valores obtidos para as medidas relativas $drpm$, cm e $melhor$ para todos os grupos de instâncias (proporcionais e não proporcionais) e tempos de processamento testados. As heurísticas GRASP e VNS acrescidas da estratégia de filtro e do procedimento de reconexão por caminhos serão denominadas, respectivamente, de GRASPF_RC e VNSF_RC. Os valores em negrito destacam o melhor resultado obtido pelas heurísticas em cada medida para cada grupo de instâncias nos tempos tp , ti e tg .

Classe	Algoritmo	$drpm$	cm	$melhor$
ORM_P (40 instâncias)	GRASPF_RC	0,003	1,60	33
	VNSF_RC	-0,003	1,40	40
GRM_P (20 instâncias)	GRASPF_RC	0,004	1,63	19
	VNSF_RC	-0,004	1,38	20
SLM_P (3 instâncias)	GRASPF_RC	-0,020	1,00	2
	VNSF_RC	0,020	2,00	2

Tabela 6.1: Qualidade relativa das heurísticas no tempo tp para as instâncias proporcionais ($w = 2$).

Classe	Algoritmo	$drpm$	cm	$melhor$
ORM_P (40 instâncias)	GRASPF_RC	-0,005	1,42	38
	VNSF_RC	0,005	1,58	37
GRM_P (20 instâncias)	GRASPF_RC	0,0006	1,45	19
	VNSF_RC	-0,0006	1,55	20
SLM_P (3 instâncias)	GRASPF_RC	-0,10	1,00	3
	VNSF_RC	0,10	2,00	0

Tabela 6.2: Qualidade relativa das heurísticas no tempo tp para as instâncias proporcionais ($w = 5$).

Primeiramente serão analisados os resultados obtidos pelas heurísticas nas instâncias proporcionais. As Tabelas 6.1 a 6.3 apresentam os resultados obtidos pelo GRASPF_RC e VNSF_RC no menor tempo de execução (tp).

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_P (40 instâncias)	GRASPf_RC	-0,027	1,44	38
	VNSf_RC	0,027	1,56	35
GRM_P (20 instâncias)	GRASPf_RC	0,00006	1,55	20
	VNSf_RC	-0,00006	1,45	18
SLM_P (3 instâncias)	GRASPf_RC	-0,37	1,00	3
	VNSf_RC	0,37	2,00	0

Tabela 6.3: Qualidade relativa das heurísticas no tempo tp para as instâncias proporcionais ($w = 10$).

As medidas *drpm* e *cm* mostram que, em média, o algoritmo GRASPf_RC obteve resultados melhores do que a heurística VNSf_RC no tempo tp para as instâncias testadas. Na medida *drpm*, o GRASPf_RC foi melhor do que o VNSf_RC em cinco grupos de instâncias: ORM_P ($w = 5$ e $w = 10$) e SLM_P ($w = 2$, $w = 5$ e $w = 10$). Na medida *cm*, o GRASPf_RC foi melhor do que o VNSf_RC em seis grupos de instâncias: ORM_P ($w = 5$ e $w = 10$), GRM_P ($w = 5$) e SLM_P ($w = 2$, $w = 5$ e $w = 10$). Na medida *melhor*, das 189 instâncias proporcionais testadas, GRASPf_RC (resp. VNSf_RC) obteve o menor valor em 175 problemas (resp. 172 problemas).

Sejam agora os resultados obtidos pelas heurísticas no tempo intermediário de execução (ti), como mostram as Tabelas 6.4 a 6.6.

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_P (40 instâncias)	GRASPf_RC	0,002	1,54	34
	VNSf_RC	-0,002	1,46	39
GRM_P (20 instâncias)	GRASPf_RC	0,0009	1,53	20
	VNSf_RC	-0,0009	1,47	20
SLM_P (3 instâncias)	GRASPf_RC	-0,0007	1,33	3
	VNSf_RC	0,0007	1,67	3

Tabela 6.4: Qualidade relativa das heurísticas no tempo ti para as instâncias proporcionais ($w = 2$).

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_P (40 instâncias)	GRASPf_RC	-0,002	1,46	40
	VNSf_RC	0,002	1,54	38
GRM_P (20 instâncias)	GRASPf_RC	0,0004	1,47	20
	VNSf_RC	-0,0004	1,53	20
SLM_P (3 instâncias)	GRASPf_RC	-0,044	1,00	3
	VNSf_RC	0,044	2,00	0

Tabela 6.5: Qualidade relativa das heurísticas no tempo ti para as instâncias proporcionais ($w = 5$).

Analisando-se os resultados obtidos no tempo ti , praticamente o mesmo padrão de comportamento ocorreu nas medidas *drpm* e *cm* em relação ao menor tempo de execução (tp). Na medida *drpm*, GRASPf_RC foi melhor do

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_P (40 instâncias)	GRASPf_RC	-0,009	1,42	39
	VNSf_RC	0,009	1,58	38
GRM_P (20 instâncias)	GRASPf_RC	-0,0001	1,50	20
	VNSf_RC	0,0001	1,50	20
SLM_P (3 instâncias)	GRASPf_RC	-0,18	1,00	3
	VNSf_RC	0,18	2,00	0

Tabela 6.6: Qualidade relativa das heurísticas no tempo t_i para as instâncias proporcionais ($w = 10$).

que o VNSf_RC em seis grupos de instâncias: ORM_P ($w = 5$ e $w = 10$), GRM_P ($w = 10$) e SLM_P ($w = 2$, $w = 5$ e $w = 10$). Na medida *cm*, o GRASPf_RC foi melhor do que o VNSf_RC em seis grupos de instâncias: ORM_P ($w = 5$ e $w = 10$), GRM_P ($w = 5$) e SLM_P ($w = 2$, $w = 5$ e $w = 10$). Na medida *melhor*, dos 189 problemas testados, GRASPf_RC e VNSf_RC obteve o menor valor em 182 e 178 problemas, respectivamente. Executando-se os algoritmos em um tempo maior em relação a tp , as heurísticas convergem para as mesmas soluções em alguns grupos de instâncias, como, por exemplo, GRM_P ($w = 2$, $w = 5$ e $w = 10$).

Sejam os resultados obtidos pelas heurísticas no maior tempo de execução (tg), como mostram as Tabelas 6.7 a 6.9.

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_P (40 instâncias)	GRASPf_RC	0,001	1,54	35
	VNSf_RC	-0,001	1,46	38
GRM_P (20 instâncias)	GRASPf_RC	0,0003	1,53	20
	VNSf_RC	-0,0003	1,47	20
SLM_P (3 instâncias)	GRASPf_RC	0,00	1,5	3
	VNSf_RC	0,00	1,5	3

Tabela 6.7: Qualidade relativa das heurísticas no tempo tg para as instâncias proporcionais ($w = 2$).

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_P (40 instâncias)	GRASPf_RC	-0,0012	1,47	39
	VNSf_RC	0,0012	1,53	38
GRM_P (20 instâncias)	GRASPf_RC	0,0004	1,55	20
	VNSf_RC	-0,0004	1,45	20
SLM_P (3 instâncias)	GRASPf_RC	-0,031	1,00	3
	VNSf_RC	0,031	2,00	1

Tabela 6.8: Qualidade relativa das heurísticas no tempo tg para as instâncias proporcionais ($w = 5$).

Analisando-se as tabelas do tempo tg , ocorreu uma pequena melhora dos resultados obtidos pelo VNSf_RC em relação ao GRASPf_RC nas medidas *drpm* e *cm*, comparando-se com os tempos de execução anteriores (tp e t_i). Nas

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_P (40 instâncias)	GRASPf_RC	-0,006	1,44	40
	VNSf_RC	0,006	1,56	39
GRM_P (20 instâncias)	GRASPf_RC	0,0002	1,53	20
	VNSf_RC	-0,0002	1,47	20
SLM_P (3 instâncias)	GRASPf_RC	-0,14	1,00	3
	VNSf_RC	0,14	2,00	0

Tabela 6.9: Qualidade relativa das heurísticas no tempo *tg* para as instâncias proporcionais ($w = 10$).

medidas *drpm* e *cm*, GRASPf_RC foi melhor do que VNSf_RC nos mesmos quatro grupos de instâncias: ORM_P ($w = 5$ e $w = 10$) e SLM_P ($w = 5$ e $w = 10$). Na medida *melhor*, das 189 instâncias testadas, GRASPf_RC (resp. VNSf_RC) obteve o menor valor em 183 problemas (resp. 179 problemas). Quando aumenta-se o tempo de execução, a tendência é de que, os algoritmos apresentem praticamente os mesmos resultados, convergindo para as mesmas soluções na maioria das instâncias testadas.

Sejam agora os resultados obtidos pelas heurísticas nas instâncias não proporcionais nos tempos *tp*, *ti* e *tg* (Tabelas 6.10 a 6.12).

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_NP (40 instâncias)	GRASPf_RC	0,005	1,56	36
	VNSf_RC	-0,005	1,44	37
GRM_NP (20 instâncias)	GRASPf_RC	0,006	1,55	19
	VNSf_RC	-0,006	1,45	20

Tabela 6.10: Qualidade relativa das heurísticas no tempo *tp* para as instâncias não proporcionais.

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_NP (40 instâncias)	GRASPf_RC	0,004	1,57	32
	VNSf_RC	-0,004	1,43	40
GRM_NP (20 instâncias)	GRASPf_RC	-0,002	1,48	20
	VNSf_RC	0,002	1,52	17

Tabela 6.11: Qualidade relativa das heurísticas no tempo *ti* para as instâncias não proporcionais.

Classe	Algoritmo	<i>drpm</i>	<i>cm</i>	<i>melhor</i>
ORM_NP (40 instâncias)	GRASPf_RC	0,004	1,60	33
	VNSf_RC	-0,004	1,40	40
GRM_NP (20 instâncias)	GRASPf_RC	-0,002	1,45	19
	VNSf_RC	0,002	1,55	18

Tabela 6.12: Qualidade relativa das heurísticas no tempo *tg* para as instâncias não proporcionais.

Analisando-se os resultados no menor tempo de execução (*tp*), em média, VNSf_RC apresentou resultados melhores do que GRASPf_RC nas medidas

drpm e *cm* em ambas as classes. Na medida *melhor*, das 60 instâncias testadas, GRASPf_RC (resp. VNSf_RC) obteve o menor valor em 55 problemas (resp. 57 problemas). Aumentando-se o tempo de execução dos algoritmos (tempo *ti*), o VNSf_RC continua sendo melhor na classe ORM_NP, porém, apresenta resultados piores do que o GRASPf_RC na classe GRM_NP. Na medida *melhor*, dos 60 problemas, GRASPf_RC (resp. VNSf_RC) obteve o menor valor em 52 problemas (resp. 57 problemas). Por último, no maior tempo de execução (*tg*), as heurísticas mantiveram o mesmo padrão de comportamento do tempo *ti* nas medidas *drpm* e *cm*. Na medida *melhor*, dos 60 problemas, GRASPf_RC (resp. VNSf_RC) obteve o menor valor em 52 problemas (resp. 58 problemas).

6.3

Tempo para Atingir um Valor Alvo

Esta seção compara as heurísticas através de gráficos que mostram a distribuição de probabilidade empírica da variável aleatória tempo gasto para encontrar um valor alvo. Como explicado na Seção 4.5.2, primeiramente fixa-se o valor alvo e faz-se 200 execuções independentes para cada heurística. Em cada execução, termina-se o algoritmo quando uma solução de valor menor ou igual ao valor alvo é encontrada. O *i*-ésimo menor tempo de execução é associado à probabilidade $prob_i = (i - 1/2)/200$ para $i = 1, 2, \dots, 200$. Através desses gráficos, pode-se comparar experimentalmente diferentes algoritmos aleatórios ou diferentes versões do mesmo algoritmo aleatório [1].

As seguintes instâncias proporcionais e não proporcionais, com seus respectivos valores alvo entre parênteses, foram avaliadas: ORM_P18 (5990) para $w = 2$, ORM_P30 (7530) para $w = 5$, ORM_P20 (10600) para $w = 10$, GRM_NP17 (7370) e ORM_NP37 (6195). Instâncias maiores foram escolhidas porque nesses problemas os algoritmos apresentaram diferenças significativas. Executando-se as heurísticas em instâncias menores, em geral, as curvas de ambas se equivalem, como mostra a Figura 6.1 (instância GRM_P5 com $w = 5$ e valor alvo 8379). Pelo gráfico, em um tempo inferior à 3,5 segundos, as heurísticas encontram um valor menor ou igual ao alvo 8379 com 100% de probabilidade.

Analisando-se as instâncias proporcionais maiores, no problema ORM_P18 com $w = 2$ (Figura 6.2), a probabilidade de encontrar uma solução com valor pelo menos tão bom quanto o valor alvo 5990 aumenta do GRASPf_RC para o VNSf_RC. A probabilidade de encontrar o valor alvo em menos de 20 segundos é aproximadamente 90% para o VNSf_RC e aproximadamente 50% para o GRASPf_RC.

Quando aumenta-se o fator w nas instâncias proporcionais, o GRASPf_RC é, em geral, superior ao VNSf_RC, como mostram as Figuras 6.3 e 6.4. No primeiro (resp. segundo) gráfico, a probabilidade de encontrar o

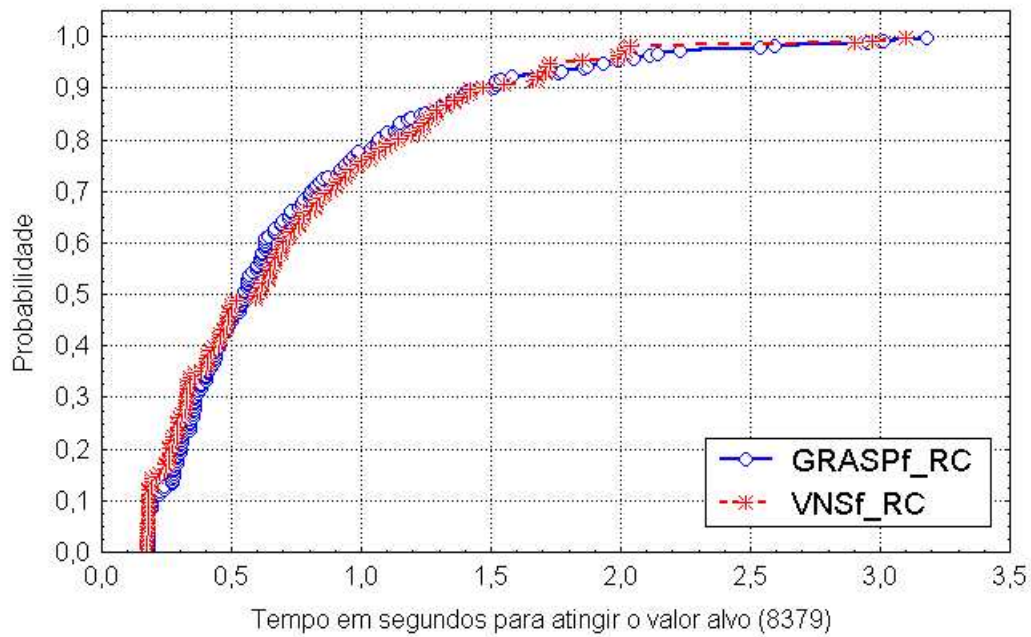


Figura 6.1: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 8379 para a instância GRM_P5 ($w = 5$).

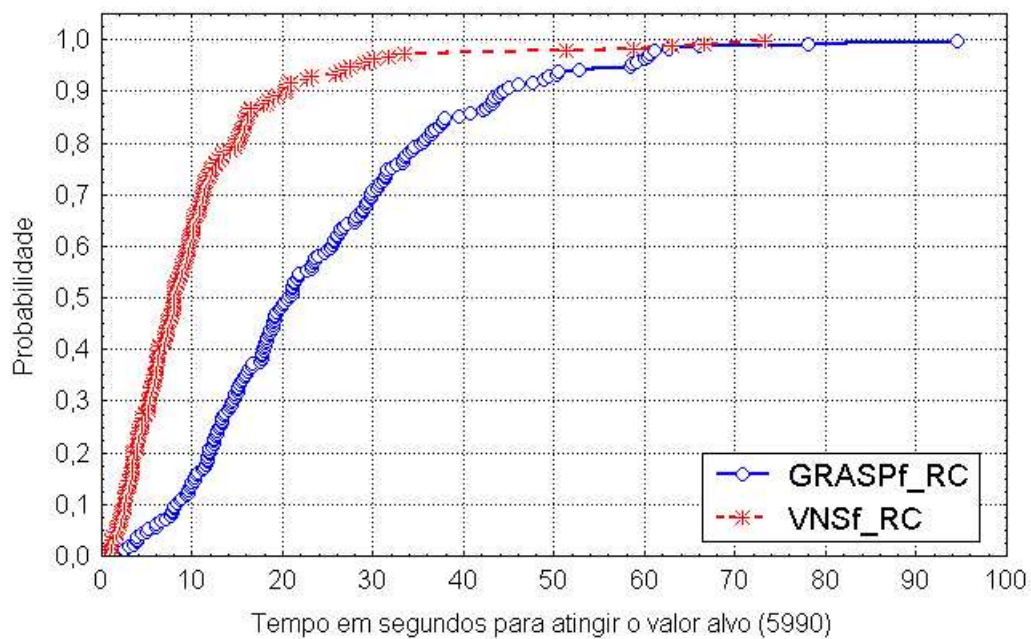


Figura 6.2: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 5990 para a instância ORM_P18 ($w = 2$).

valor alvo em menos de 250 segundos (resp. 95 segundos) é de 100% para o GRASPf_RC e de aproximadamente 85% (resp. 64%) para o VNSf_RC.

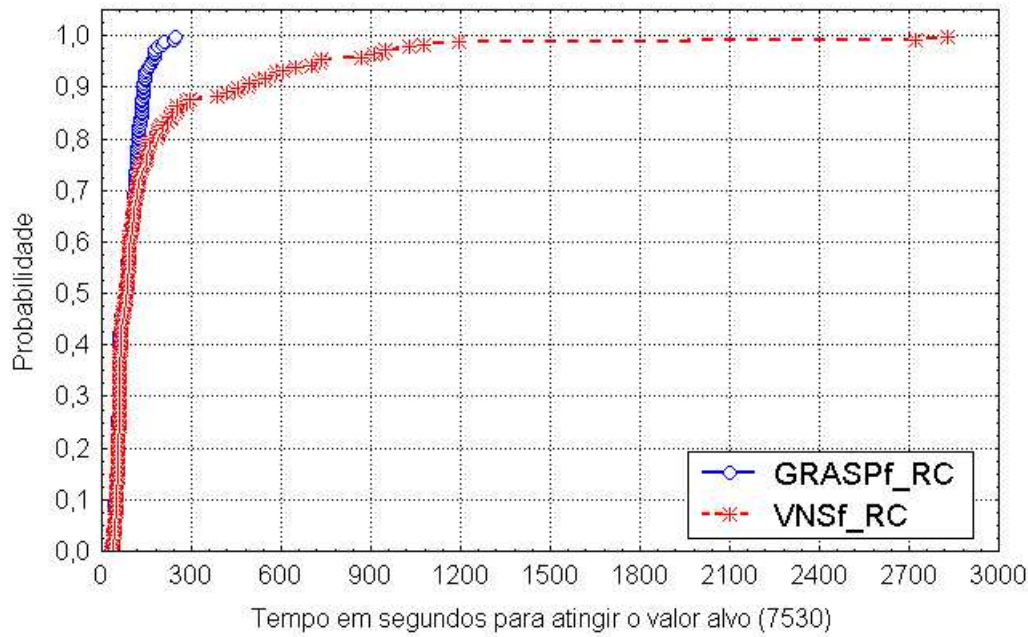


Figura 6.3: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 7530 para a instância ORM_P30 ($w = 5$).

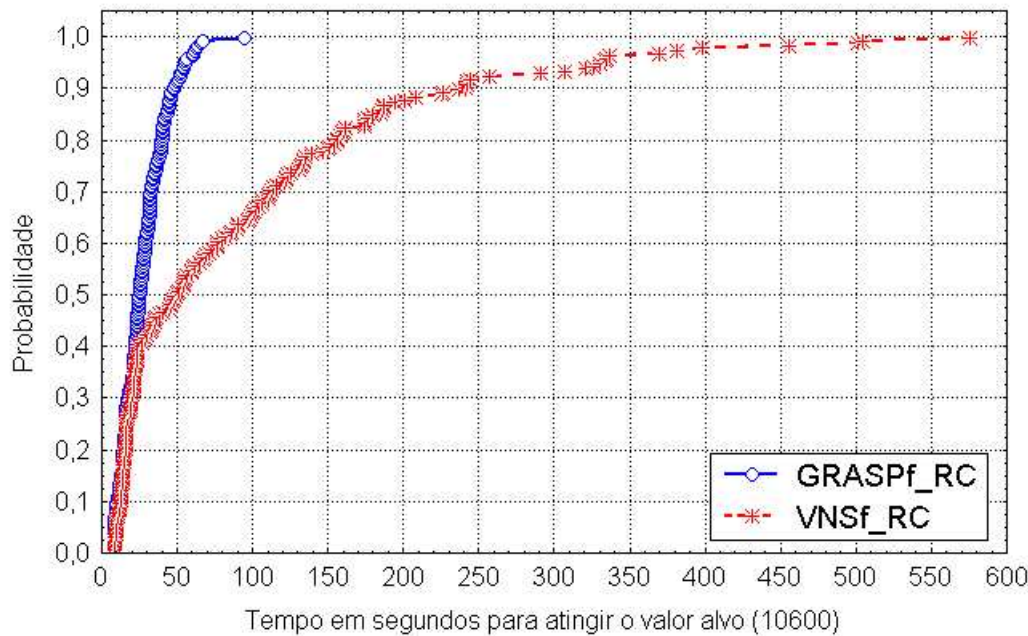


Figura 6.4: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 10600 para a instância ORM_P20 ($w = 10$).

Analisando-se agora as duas instâncias não proporcionais testadas, observou-se que, GRASPf_RC é superior ao VNSf_RC no problema GRM_NP17 e que, VNSf_RC é superior ao GRASPf_RC na instância ORM_NP37. Na Figura 6.5, a probabilidade de encontrar o valor alvo em menos de 100 segun-

dos é de aproximadamente 90% para o GRASPf_RC e de aproximadamente 71% para o VNSf_RC, enquanto que, na Figura 6.6, um valor menor ou igual ao alvo 6195 é encontrado em um tempo inferior à 100 segundos pelo VNSf_RC e GRASPf_RC com 95% e 45% de probabilidade, respectivamente.

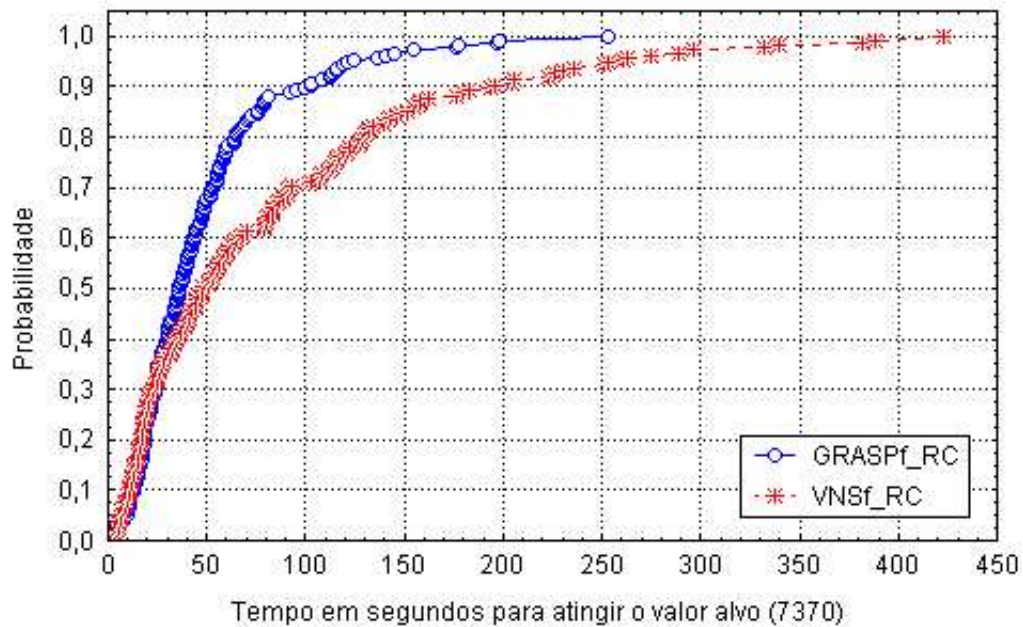


Figura 6.5: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 7370 para a instância GRM_NP17.

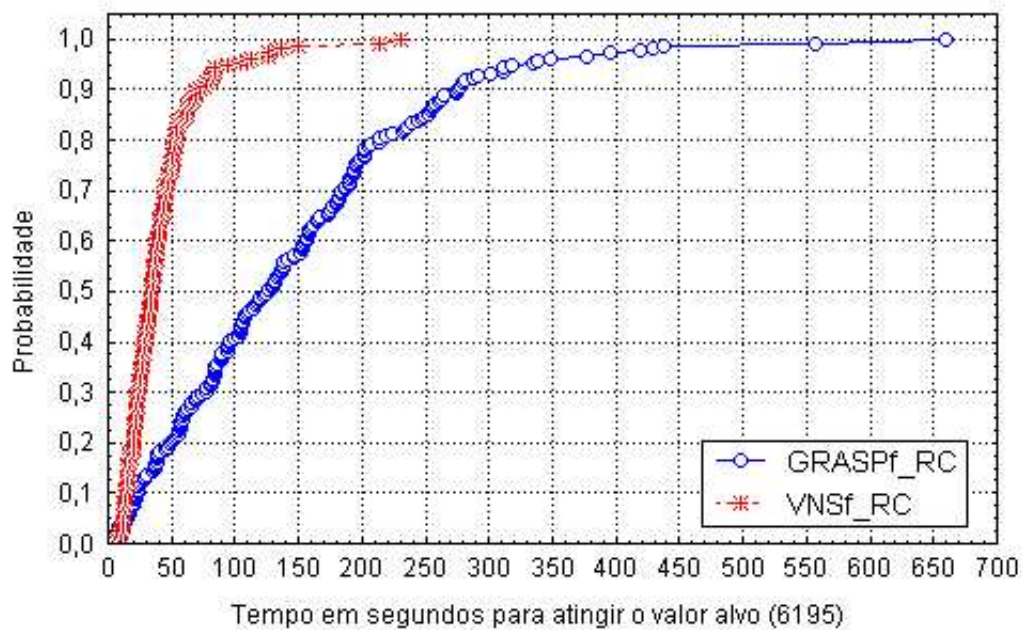


Figura 6.6: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 6195 para a instância ORM_NP37.

6.4

Soluções Ótimas e Passo de Pós-Otimização

O modelo baseado em árvore geradora mínima restrita por grau possibilitou a obtenção de soluções ótimas para 49 das 189 instâncias proporcionais testadas, como mostram as Tabelas 6.13 e 6.14. As colunas indicam o nome do problema (Instância), o número de vértices do grafo (n), o número de arestas do grafo ($|E|$), o número de facilidades a serem instaladas (p), o valor da solução ótima do problema (F_O), o tempo gasto em segundos para encontrar a solução ótima (Tempo (s)) e o número de nós explorados pelo algoritmo *branch and bound* (Nós).

Observando-se as Tabelas 6.13 e 6.14, pode-se notar que a solução ótima foi encontrada somente para instâncias proporcionais relativamente pequenas em termos de número de vértices e facilidades a serem instaladas, o que mostra a dificuldade para a resolução exata de instâncias até mesmo menores. Gastou-se mais de oito dias de tempo de processamento para encontrar a solução ótima da instância ORM_P27 ($w = 2$) e quase dez dias de tempo de processamento para encontrar a solução ótima da instância GRM_P12 ($w = 2$). O problema ORM_P35 ($w = 2$) não foi resolvido devido à falta de memória. A solução ótima do problema ORM_P12 com $w = 2$ (instância com asterisco na Tabela 6.13) foi encontrada inicializando-se o resolvidor de programação linear inteira com o valor da solução obtida pelas heurísticas GRASPf_RC e VNSf_RC no tempo tg (7074). Mesmo utilizando o valor da solução obtida pelas heurísticas como limite inicial, gastou-se aproximadamente três dias de tempo de processamento para encontrar a solução ótima do problema.

A obtenção de soluções ótimas para esses problemas possibilitou realizar comparações em termos absolutos com as heurísticas. Para cada instância, a melhor solução obtida pelo GRASPf_RC e pelo VNSf_RC em cinco execuções com o tempo limitado a tg (em segundos) será utilizada para a comparação com a solução ótima. Para as instâncias em que não se conhece o ótimo, a melhor solução encontrada pelo algoritmo GRASPf_RC ou pela heurística VNSf_RC será a melhor solução conhecida. As Tabelas 6.15 a 6.25 apresentam esses resultados para as instâncias proporcionais e não proporcionais, onde as colunas F_G^* e F_V^* correspondem ao valor da melhor solução encontrada pelos algoritmos GRASPf_RC e VNSf_RC, respectivamente. As colunas $\%_G = 100 \times \frac{F_G^* - F_O}{F_O}$ e $\%_V = 100 \times \frac{F_V^* - F_O}{F_O}$ indicam, respectivamente, o desvio relativo percentual do valor das soluções obtidas pelos algoritmos GRASPf_RC e VNSf_RC em relação ao valor ótimo de cada instância proporcional. Os valores em negrito representam o melhor resultado obtido pelo GRASPf_RC ou VNSf_RC nas instâncias em que não ocorreram empates.

Comparando-se a qualidade das soluções e os tempos de processamento do algoritmo exato (Tabelas 6.13 e 6.14) e das heurísticas (Tabelas 6.15

a 6.20) nas instâncias proporcionais, observa-se que, em poucos segundos, o GRASPf_RC e o VNSf_RC encontraram a solução ótima na maioria das instâncias testadas, enquanto o algoritmo exato gastou horas ou até mesmo dias de tempo de processamento para encontrá-la. Porém, em alguns problemas como, por exemplo, GRM_P1 e ORM_P27 ($w = 2$), as heurísticas não conseguiram encontrar a solução ótima.

	Instância	n	$ E $	p	F_O	Tempo (s)	Nós
$w = 2$	ORM_P1	100	198	5	6443	1.072,39	59.893
	ORM_P2	100	193	10	5228	4.792,11	140.621
	ORM_P3	100	198	10	5368	9.366,33	372.665
	ORM_P4	100	196	20	5123	108.305,75	5.477.384
	ORM_P6	200	786	5	8180	58.447,19	1.016.864
	ORM_P7	200	779	10	6217	166.072,09	2.098.362
	ORM_P11	300	1.772	5	7855	5.072,67	22.906
	ORM_P12*	300	1.758	10	7074	266.225,22	910.891
	ORM_P16	400	3.153	5	8292	3.843,13	2.350
	ORM_P17	400	3.142	10	7279	40.988,16	64.417
	ORM_P21	500	4.909	5	9272	10.817,77	9.453
	ORM_P22	500	4.896	10	8839	169.399,73	209.789
	ORM_P26	600	7.068	5	10022	41.627,77	45.071
	ORM_P27	600	7.072	10	8553	720.582,59	460.940
	ORM_P31	700	9.601	5	10185	102.431,89	50.702
$w = 5$	ORM_P1	100	198	5	7184	1.599,26	54.990
	ORM_P2	100	193	10	6572	28.168,31	1.067.082
	ORM_P3	100	198	10	6776	76.090,34	1.749.654
	ORM_P11	300	1.772	5	8062	14.966,63	86.414
	ORM_P16	400	3.153	5	8458	10.551,84	3.675
	ORM_P21	500	4.909	5	9473	54.611,31	94.832
	ORM_P26	600	7.068	5	10169	179.345,39	237.600
$w = 10$	ORM_P1	100	198	5	8146	1.310,67	33.641
	ORM_P2	100	193	10	7706	7.315,97	219.336
	ORM_P3	100	198	10	8265	73.774,47	1.732.743
	ORM_P11	300	1.772	5	8383	29.226,83	56.750
	ORM_P16	400	3.153	5	8728	56.704,55	16.187
	ORM_P21	500	4.909	5	9808	72.600,92	47.605

Tabela 6.13: Valor da solução ótima para algumas instâncias da classe proporcional ORM_P ($w = 2$, $w = 5$ e $w = 10$).

Comparando-se as heurísticas entre si (Tabelas 6.15 a 6.25), observa-se que as mesmas obtiveram praticamente os mesmos resultados no maior tempo de execução (tg). Nas instâncias proporcionais (189 problemas), GRASPf_RC e VNSf_RC encontraram os mesmos resultados em 173 instâncias. GRASPf_RC foi superior ao VNSf_RC em dez instâncias e VNSf_RC foi melhor do que GRASPf_RC em seis instâncias. Nas instâncias não proporcionais (60 problemas), GRASPf_RC e VNSf_RC encontraram os mesmos resultados em

49 instâncias. GRASPf_RC foi superior ao VNSf_RC em duas instâncias e VNSf_RC foi melhor do que GRASPf_RC em oito instâncias.

Em relação às 49 instâncias proporcionais cujo valor ótimo é conhecido, os resultados apresentados pelas heurísticas são idênticos. GRASPf_RC e VNSf_RC conseguiram encontrar a solução ótima em 42 problemas. Nas sete instâncias restantes, as heurísticas ficaram, no máximo, a 0,54% do valor ótimo em GRM_P10 com $w = 2$.

Analisando-se o conjunto de facilidades abertas encontradas na solução ótima e comparando-se com as facilidades abertas encontradas pelas heurísticas nas sete instâncias, foi observado que, em algumas delas, as mesmas facilidades foram abertas pelo algoritmo exato e pela heurística, mas a árvore de Steiner obtida pela heurística Prim ao conectá-las não era ótima. A partir dessas observações, propõe-se então um passo adicional de pós-otimização, que consiste em utilizar um algoritmo exato para o problema de Steiner usando como entrada as facilidades abertas na melhor solução encontrada pelos algoritmos GRASPf_RC e VNSf_RC após todas as iterações terem sido realizadas.

	Instância	n	$ E $	p	F_O	Tempo (s)	Nós
$w = 2$	GRM_P1	100	4.950	5	6391	6.496,75	84.575
	GRM_P2	100		10	5475	45.788,98	154.709
	GRM_P3	100		15	5157	33.666,48	95.137
	GRM_P4	100		20	5246	269.193,56	1.559.511
	GRM_P10	150	11.175	5	11413	15.056,66	17.124
	GRM_P11	150		10	9823	83.297,28	51.512
	GRM_P12	150		15	8867	856.170,36	361.340
$w = 5$	GRM_P1	100	4.950	5	7036	1.383,08	2.104
	GRM_P2	100		10	6662	35.836,39	86.685
	GRM_P3	100		15	6923	19.166,23	57.150
	GRM_P4	100		20	7602	20.264,56	175.112
	GRM_P10	150	11.175	5	12202	7.655,94	5.596
	GRM_P11	150		10	11086	79.751,42	53.927
	GRM_P12	150		15	10767	319.858,98	187.331
$w = 10$	GRM_P1	100	4.950	5	7884	1.466,13	2.613
	GRM_P2	100		10	8580	22.530,55	76.213
	GRM_P3	100		15	9793	2.949,95	7.896
	GRM_P4	100		20	11487	16.681,17	191.562
	GRM_P10	150	11.175	5	13083	8.693,39	6.717
	GRM_P11	150		10	13040	28.675,19	21.631
	GRM_P12	150		15	13845	373.131,34	312.704

Tabela 6.14: Valor da solução ótima para algumas instâncias da classe proporcional GRM_P ($w = 2$, $w = 5$ e $w = 10$).

Instância	n	p	tg (s)	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
ORM_P1	100	5	70,00	6443	6443	0,00	6443	0,00
ORM_P2	100	10	70,00	5228	5228	0,00	5228	0,00
ORM_P3	100	10	70,00	5368	5368	0,00	5368	0,00
ORM_P4	100	20	70,00	5123	5123	0,00	5123	0,00
ORM_P5	100	33	70,00	—	3663	—	3663	—
ORM_P6	200	5	70,00	8180	8185	0,06	8185	0,06
ORM_P7	200	10	70,00	6217	6217	0,00	6217	0,00
ORM_P8	200	20	70,00	—	5865	—	5865	—
ORM_P9	200	40	70,00	—	4699	—	4699	—
ORM_P10	200	67	110,60	—	3457	—	3457	—
ORM_P11	300	5	70,00	7855	7855	0,00	7855	0,00
ORM_P12	300	10	70,00	7074	7074	0,00	7074	0,00
ORM_P13	300	30	70,00	—	5539	—	5539	—
ORM_P14	300	60	136,50	—	4981	—	4981	—
ORM_P15	300	100	565,60	—	4477	—	4477	—
ORM_P16	400	5	70,00	8292	8292	0,00	8292	0,00
ORM_P17	400	10	70,00	7279	7279	0,00	7279	0,00
ORM_P18	400	40	70,00	—	5982	—	5984	—
ORM_P19	400	80	439,60	—	4787	—	4787	—
ORM_P20	400	133	1.397,20	—	4860	—	4860	—
ORM_P21	500	5	70,00	9272	9272	0,00	9272	0,00
ORM_P22	500	10	70,00	8839	8839	0,00	8839	0,00
ORM_P23	500	50	105,70	—	5799	—	5799	—
ORM_P24	500	100	1.129,80	—	4983	—	4982	—
ORM_P25	500	167	3.789,10	—	4808	—	4808	—
ORM_P26	600	5	70,00	10022	10022	0,00	10022	0,00
ORM_P27	600	10	70,00	8553	8555	0,02	8555	0,02
ORM_P28	600	60	223,30	—	5599	—	5598	—
ORM_P29	600	120	2.313,50	—	5078	—	5079	—
ORM_P30	600	200	6.317,50	—	5133	—	5133	—
ORM_P31	700	5	70,00	10185	10185	0,00	10185	0,00
ORM_P32	700	10	70,00	—	9509	—	9509	—
ORM_P33	700	70	225,40	—	5920	—	5919	—
ORM_P34	700	140	3.195,50	—	5099	—	5099	—
ORM_P35	800	5	70,00	—	10485	—	10485	—
ORM_P36	800	10	70,00	—	10157	—	10157	—
ORM_P37	800	80	504,70	—	6403	—	6402	—
ORM_P38	900	5	70,00	—	11156	—	11156	—
ORM_P39	900	10	70,00	—	9591	—	9591	—
ORM_P40	900	90	1.268,40	—	6438	—	6437	—

Tabela 6.15: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe ORM_P ($w = 2$).

Instância	n	p	tg (s)	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
ORM_P1	100	5	70,00	7184	7184	0,00	7184	0,00
ORM_P2	100	10	70,00	6572	6572	0,00	6572	0,00
ORM_P3	100	10	70,00	6776	6776	0,00	6776	0,00
ORM_P4	100	20	70,00	—	6944	—	6944	—
ORM_P5	100	33	70,00	—	5292	—	5292	—
ORM_P6	200	5	70,00	—	8662	—	8662	—
ORM_P7	200	10	70,00	—	6977	—	6967	—
ORM_P8	200	20	70,00	—	7203	—	7203	—
ORM_P9	200	40	83,30	—	6436	—	6436	—
ORM_P10	200	67	94,50	—	5004	—	5004	—
ORM_P11	300	5	70,00	8062	8062	0,00	8062	0,00
ORM_P12	300	10	70,00	—	7666	—	7666	—
ORM_P13	300	30	114,80	—	6693	—	6693	—
ORM_P14	300	60	277,90	—	6632	—	6632	—
ORM_P15	300	100	441,00	—	6363	—	6363	—
ORM_P16	400	5	70,00	8458	8458	0,00	8458	0,00
ORM_P17	400	10	70,00	—	7676	—	7676	—
ORM_P18	400	40	304,50	—	7259	—	7259	—
ORM_P19	400	80	917,00	—	6537	—	6537	—
ORM_P20	400	133	1.570,10	—	7256	—	7258	—
ORM_P21	500	5	70,00	9473	9473	0,00	9473	0,00
ORM_P22	500	10	70,00	—	9219	—	9219	—
ORM_P23	500	50	954,80	—	7020	—	7020	—
ORM_P24	500	100	2.519,30	—	6648	—	6648	—
ORM_P25	500	167	3.541,30	—	6897	—	6897	—
ORM_P26	600	5	70,00	10169	10169	0,00	10169	0,00
ORM_P27	600	10	70,00	—	8893	—	8893	—
ORM_P28	600	60	925,40	—	6791	—	6791	—
ORM_P29	600	120	4.786,60	—	6739	—	6739	—
ORM_P30	600	200	9.401,70	—	7485	—	7486	—
ORM_P31	700	5	70,00	—	10332	—	10332	—
ORM_P32	700	10	70,00	—	9821	—	9821	—
ORM_P33	700	70	2.183,30	—	7224	—	7224	—
ORM_P34	700	140	7.136,50	—	6771	—	6771	—
ORM_P35	800	5	70,00	—	10610	—	10610	—
ORM_P36	800	10	70,00	—	10452	—	10452	—
ORM_P37	800	80	4.980,50	—	7897	—	7897	—
ORM_P38	900	5	70,00	—	11300	—	11300	—
ORM_P39	900	10	70,00	—	9839	—	9839	—
ORM_P40	900	90	7.820,40	—	7836	—	7836	—

Tabela 6.16: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe ORM_P ($w = 5$).

Instância	n	p	tg (s)	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
ORM_P1	100	5	70,00	8146	8146	0,00	8146	0,00
ORM_P2	100	10	70,00	7706	7706	0,00	7706	0,00
ORM_P3	100	10	70,00	8265	8265	0,00	8265	0,00
ORM_P4	100	20	70,00	—	9135	—	9135	—
ORM_P5	100	33	70,00	—	7485	—	7485	—
ORM_P6	200	5	70,00	—	9387	—	9387	—
ORM_P7	200	10	70,00	—	7878	—	7878	—
ORM_P8	200	20	70,00	—	8632	—	8632	—
ORM_P9	200	40	70,00	—	8174	—	8174	—
ORM_P10	200	67	70,00	—	6894	—	6894	—
ORM_P11	300	5	70,00	8383	8383	0,00	8383	0,00
ORM_P12	300	10	70,00	—	8426	—	8426	—
ORM_P13	300	30	70,00	—	7679	—	7679	—
ORM_P14	300	60	372,40	—	8505	—	8505	—
ORM_P15	300	100	593,60	—	8864	—	8864	—
ORM_P16	400	5	70,00	8728	8728	0,00	8728	0,00
ORM_P17	400	10	70,00	—	8317	—	8317	—
ORM_P18	400	40	197,40	—	8547	—	8547	—
ORM_P19	400	80	1.267,70	—	8313	—	8313	—
ORM_P20	400	133	1.599,50	—	10386	—	10397	—
ORM_P21	500	5	70,00	9808	9808	0,00	9808	0,00
ORM_P22	500	10	70,00	—	9776	—	9776	—
ORM_P23	500	50	679,00	—	8359	—	8359	—
ORM_P24	500	100	2.735,60	—	8548	—	8548	—
ORM_P25	500	167	3.761,80	—	9610	—	9610	—
ORM_P26	600	5	70,00	—	10414	—	10414	—
ORM_P27	600	10	70,00	—	9317	—	9317	—
ORM_P28	600	60	2.317,00	—	8077	—	8077	—
ORM_P29	600	120	4.769,10	—	8552	—	8552	—
ORM_P30	600	200	9.440,90	—	10735	—	10735	—
ORM_P31	700	5	70,00	—	10556	—	10556	—
ORM_P32	700	10	70,00	—	10304	—	10304	—
ORM_P33	700	70	3.476,20	—	8666	—	8666	—
ORM_P34	700	140	5.868,80	—	8551	—	8551	—
ORM_P35	800	5	70,00	—	10811	—	10811	—
ORM_P36	800	10	70,00	—	10898	—	10898	—
ORM_P37	800	80	6.720,70	—	9575	—	9575	—
ORM_P38	900	5	70,00	—	11525	—	11525	—
ORM_P39	900	10	70,00	—	10220	—	10220	—
ORM_P40	900	90	14.610,40	—	9380	—	9380	—

Tabela 6.17: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe ORM_P ($w = 10$).

Instância	n	p	$tg (s)$	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
GRM_P1	100	5	70,00	6391	6403	0,19	6403	0,19
GRM_P2	100	10	70,00	5475	5489	0,26	5489	0,26
GRM_P3	100	15	70,00	5157	5157	0,00	5157	0,00
GRM_P4	100	20	70,00	5246	5246	0,00	5246	0,00
GRM_P5	100	25	70,00	—	5403	—	5403	—
GRM_P6	100	30	70,00	—	5580	—	5580	—
GRM_P7	100	35	70,00	—	5766	—	5766	—
GRM_P8	100	40	70,00	—	5961	—	5961	—
GRM_P9	100	50	70,00	—	6364	—	6364	—
GRM_P10	150	5	70,00	11413	11475	0,54	11475	0,54
GRM_P11	150	10	70,00	9823	9862	0,40	9862	0,40
GRM_P12	150	15	70,00	8867	8910	0,48	8910	0,48
GRM_P13	150	20	70,00	—	8352	—	8352	—
GRM_P14	150	25	70,00	—	8174	—	8174	—
GRM_P15	150	30	70,70	—	8133	—	8133	—
GRM_P16	150	35	197,40	—	8243	—	8243	—
GRM_P17	150	40	182,70	—	8372	—	8372	—
GRM_P18	150	45	270,20	—	8530	—	8530	—
GRM_P19	150	50	333,20	—	8701	—	8701	—
GRM_P20	150	60	328,30	—	9080	—	9080	—

Tabela 6.18: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe GRM_P ($w = 2$).

Instância	n	p	$tg (s)$	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
GRM_P1	100	5	70,00	7036	7036	0,00	7036	0,00
GRM_P2	100	10	70,00	6662	6662	0,00	6662	0,00
GRM_P3	100	15	70,00	6923	6923	0,00	6923	0,00
GRM_P4	100	20	70,00	7602	7602	0,00	7602	0,00
GRM_P5	100	25	70,00	—	8379	—	8379	—
GRM_P6	100	30	70,00	—	9178	—	9178	—
GRM_P7	100	35	70,00	—	9984	—	9984	—
GRM_P8	100	40	70,00	—	10802	—	10802	—
GRM_P9	100	50	70,00	—	12447	—	12447	—
GRM_P10	150	5	70,00	12202	12202	0,00	12202	0,00
GRM_P11	150	10	70,00	11086	11086	0,00	11086	0,00
GRM_P12	150	15	70,00	10767	10767	0,00	10767	0,00
GRM_P13	150	20	70,00	—	10902	—	10902	—
GRM_P14	150	25	82,60	—	11293	—	11293	—
GRM_P15	150	30	149,10	—	11874	—	11874	—
GRM_P16	150	35	212,80	—	12539	—	12539	—
GRM_P17	150	40	197,40	—	13276	—	13276	—
GRM_P18	150	45	221,20	—	14036	—	14036	—
GRM_P19	150	50	426,30	—	14814	—	14814	—
GRM_P20	150	60	386,40	—	16422	—	16422	—

Tabela 6.19: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe GRM_P ($w = 5$).

Instância	n	p	$tg (s)$	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
GRM_P1	100	5	70,00	7884	7884	0,00	7884	0,00
GRM_P2	100	10	70,00	8580	8580	0,00	8580	0,00
GRM_P3	100	15	70,00	9793	9793	0,00	9793	0,00
GRM_P4	100	20	70,00	11487	11487	—	11487	—
GRM_P5	100	25	70,00	—	13273	—	13273	—
GRM_P6	100	30	70,00	—	15079	—	15079	—
GRM_P7	100	35	70,00	—	16909	—	16909	—
GRM_P8	100	40	70,00	—	18754	—	18754	—
GRM_P9	100	50	70,00	—	22507	—	22507	—
GRM_P10	150	5	70,00	13083	13083	0,00	13083	0,00
GRM_P11	150	10	70,00	13040	13040	0,00	13040	0,00
GRM_P12	150	15	70,00	13845	13845	0,00	13845	0,00
GRM_P13	150	20	70,00	—	14985	—	14985	—
GRM_P14	150	25	142,80	—	16355	—	16355	—
GRM_P15	150	30	119,70	—	17919	—	17919	—
GRM_P16	150	35	168,00	—	19579	—	19579	—
GRM_P17	150	40	157,50	—	21317	—	21317	—
GRM_P18	150	45	211,40	—	23095	—	23095	—
GRM_P19	150	50	247,80	—	24895	—	24895	—
GRM_P20	150	60	252,70	—	28566	—	28566	—

Tabela 6.20: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe GRM_P ($w = 10$).

Instância	n	p	$tg (s)$	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
SLM_P1	700	233	10.581,90	—	4851	—	4851	—
SLM_P2	800	267	16.639,00	—	5317	—	5317	—
SLM_P3	900	300	36.101,80	—	5409	—	5409	—

Tabela 6.21: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe SLM_P ($w = 2$).

Instância	n	p	$tg (s)$	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
SLM_P1	700	233	12.399,80	—	6961	—	6961	—
SLM_P2	800	267	24.776,50	—	7840	—	7841	—
SLM_P3	900	300	35.674,10	—	7880	—	7885	—

Tabela 6.22: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe SLM_P ($w = 5$).

Instância	n	p	$tg (s)$	F_O	F_G^*	$\%_G$	F_V^*	$\%_V$
SLM_P1	700	233	12.350,10	—	9767	—	9788	—
SLM_P2	800	267	33.688,90	—	11304	—	11319	—
SLM_P3	900	300	51.396,10	—	11273	—	11289	—

Tabela 6.23: Comparações absolutas entre as heurísticas para as instâncias proporcionais da classe SLM_P ($w = 10$).

Instância	n	p	tg (s)	F_G^*	F_V^*
ORM_NP1	100	5	70,00	6447	6447
ORM_NP2	100	10	70,00	5196	5196
ORM_NP3	100	10	70,00	5235	5235
ORM_NP4	100	20	70,00	4935	4930
ORM_NP5	100	33	70,00	3377	3377
ORM_NP6	200	5	70,00	8098	8098
ORM_NP7	200	10	70,00	6153	6153
ORM_NP8	200	20	70,00	5627	5623
ORM_NP9	200	40	70,00	4364	4364
ORM_NP10	200	67	110,60	3362	3362
ORM_NP11	300	5	70,00	7832	7832
ORM_NP12	300	10	70,00	6979	6979
ORM_NP13	300	30	70,00	5342	5342
ORM_NP14	300	60	136,50	4781	4781
ORM_NP15	300	100	565,60	4120	4120
ORM_NP16	400	5	70,00	8293	8293
ORM_NP17	400	10	70,00	7246	7246
ORM_NP18	400	40	70,00	5838	5835
ORM_NP19	400	80	439,60	4546	4546
ORM_NP20	400	133	1.397,20	4428	4428
ORM_NP21	500	5	70,00	9238	9238
ORM_NP22	500	10	70,00	8800	8800
ORM_NP23	500	50	105,70	5629	5628
ORM_NP24	500	100	1.129,80	4678	4678
ORM_NP25	500	167	3.789,10	4538	4538
ORM_NP26	600	5	70,00	9996	9996
ORM_NP27	600	10	70,00	8514	8514
ORM_NP28	600	60	223,30	5442	5438
ORM_NP29	600	120	2.313,50	4801	4801
ORM_NP30	600	200	6.317,50	4700	4700
ORM_NP31	700	5	70,00	10158	10158
ORM_NP32	700	10	70,00	9474	9474
ORM_NP33	700	70	225,40	5723	5718
ORM_NP34	700	140	3.195,50	4759	4759
ORM_NP35	800	5	70,00	10463	10463
ORM_NP36	800	10	70,00	10116	10116
ORM_NP37	800	80	504,70	6190	6189
ORM_NP38	900	5	70,00	11129	11129
ORM_NP39	900	10	70,00	9551	9551
ORM_NP40	900	90	1.268,40	6213	6213

Tabela 6.24: Comparações absolutas entre as heurísticas para as instâncias não proporcionais da classe ORM_NP.

Instância	n	p	tg (s)	F_G^*	F_V^*
GRM_NP1	100	5	70,00	6156	6156
GRM_NP2	100	10	70,00	5185	5185
GRM_NP3	100	15	70,00	4825	4825
GRM_NP4	100	20	70,00	4720	4720
GRM_NP5	100	25	70,00	4675	4675
GRM_NP6	100	30	70,00	4675	4675
GRM_NP7	100	35	70,00	4685	4685
GRM_NP8	100	40	70,00	4717	4717
GRM_NP9	100	50	70,00	4806	4806
GRM_NP10	150	5	70,00	11334	11334
GRM_NP11	150	10	70,00	9710	9710
GRM_NP12	150	15	70,00	8751	8759
GRM_NP13	150	20	70,00	8135	8135
GRM_NP14	150	25	70,00	7719	7719
GRM_NP15	150	30	70,70	7527	7527
GRM_NP16	150	35	197,40	7427	7431
GRM_NP17	150	40	182,70	7362	7360
GRM_NP18	150	45	270,20	7341	7341
GRM_NP19	150	50	333,20	7346	7346
GRM_NP20	150	60	328,30	7405	7405

Tabela 6.25: Comparações absolutas entre as heurísticas para as instâncias não proporcionais da classe GRM_NP.

6.4.1

Passo de Pós-Otimização

O algoritmo de pós-otimização poderia ser uma heurística de busca local ou mesmo uma heurística baseada em uma metaheurística para o problema de Steiner. Um algoritmo exato foi utilizado porque o número de terminais é relativamente pequeno em comparação com instâncias testadas na literatura para o problema de Steiner [61], não demandando, assim, um custo computacional alto. Além disso, obtém-se também um parâmetro para testar a eficiência da heurística construtiva Prim escolhida para conectar as p facilidades abertas. A pós-otimização é executada por um algoritmo do tipo *branch and ascent* [61].

As Tabelas 6.26 a 6.44 apresentam a melhor solução encontrada pelos algoritmos aplicando-se o *branch and ascent* à solução obtida em cada uma das cinco execuções das heurísticas limitadas ao tempo tg . As heurísticas GRASPf_RC e VNSf_RC acrescidas do procedimento de pós-otimização serão denominadas, respectivamente, de GRASPf_RC_PO e VNSf_RC_PO. As heurísticas apresentaram os mesmos resultados para as classes de instâncias GRM_P ($w = 2$, $w = 5$ e $w = 10$) e, por isso, os resultados serão mostrados em uma mesma tabela. A coluna F_P indica o valor da função objetivo relacionado ao custo de atendimento dos usuários realizado pelas p facilidades

abertas referente a melhor solução encontrada pelos algoritmos; A coluna F_T indica o valor da função objetivo relacionado ao custo de interconexão das facilidades abertas referente a melhor solução encontrada pelos algoritmos; F_{OT} representa o valor da árvore de Steiner obtida pelo algoritmo exato no passo de pós-otimização; Tempo (s) indica o tempo gasto em segundos na execução do algoritmo exato e F_{PO} representa o valor da melhor solução encontrada pelas heurísticas, levando-se em consideração, os possíveis ganhos obtidos com a pós-otimização. Os valores em negrito nas colunas F_T e F_{OT} representam uma melhoria na árvore de Steiner obtida pela pós-otimização e, conseqüentemente, uma melhoria na melhor solução encontrada pelos algoritmos (coluna F_{PO}).

A coluna Tempo (s) indica que o algoritmo exato do passo de pós-otimização executa em menos de um segundo para todas as instâncias testadas. A escolha de Prim se mostrou bastante acertada para integrar o algoritmo de busca local utilizado pelas heurísticas, pois na grande maioria dos problemas, a heurística construtiva conseguiu encontrar soluções com valor ótimo obtido pelo algoritmo *branch and ascent*.

Adicionando-se o passo de pós-otimização, as heurísticas encontraram a solução ótima nas instâncias ORM_P27 e GRM_P1 ($w = 2$). Nas instâncias ORM_P6, GRM_P2, GRM_P10, GRM_P11 e GRM_P12 ($w = 2$), em que a solução ótima é conhecida e as heurísticas GRASPf_RC e VNSf_RC não conseguiram encontrar o ótimo, houve uma melhoria somente na instância GRM_P10, com o desvio relativo percentual caindo de 0,54% para 0,46%. Nessas cinco instâncias, GRASPf_RC_PO e VNSf_RC_PO ficaram, no máximo, a 0,48% do valor ótimo na instância GRM_P12.

Nos problemas em que não se conhece o ótimo, GRASPf_RC_PO conseguiu melhorar a melhor solução encontrada por GRASPf_RC em 13 instâncias proporcionais: ORM_P23, ORM_P28, ORM_P29, ORM_P33, ORM_P34 e ORM_P37 ($w = 2$), ORM_P7, ORM_P27 e ORM_P36 ($w = 5$), ORM_P32 e ORM_P39 ($w = 2$ e $w = 5$). Nas instâncias não proporcionais, ocorreram 16 melhorias: ORM_P8, ORM_P14, ORM_P16, ORM_P18, ORM_P19, ORM_P22, ORM_P23, ORM_P27, ORM_P28, ORM_P29, ORM_P33, ORM_P36, ORM_P37 e ORM_P40 ($w = 2$), GRM_P1 e GRM_P10 ($w = 2$). Já o VNSf_RC_PO conseguiu melhorar a melhor solução encontrada pelo VNSf_RC em 12 instâncias proporcionais: ORM_P23, ORM_P28, ORM_P29, ORM_P33 e ORM_P37 ($w = 2$), ORM_P7, ORM_P27 e ORM_P36 ($w = 5$), ORM_P32 e ORM_P39 ($w = 2$ e $w = 5$). Nas instâncias não proporcionais, ocorreram 16 melhorias: ORM_P14, ORM_P16, ORM_P18, ORM_P19, ORM_P22, ORM_P23, ORM_P27, ORM_P28, ORM_P29, ORM_P33, ORM_P34, ORM_P36, ORM_P37 e ORM_P40 ($w = 2$), GRM_P1 e GRM_P10 ($w = 2$). Com a pós-otimização, das 189 (resp. 60) instâncias proporcionais (resp. não

proporcionais) testadas, GRASPf_RC_PO e VNSf_RC_PO encontraram os mesmos resultados em 174 (resp. 53) instâncias. GRASPf_RC foi superior a VNSf_RC em dez (resp. duas) instâncias e VNSf_RC foi melhor do que GRASPf_RC em cinco (resp. cinco) instâncias.

Instância	F_O	F_G^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_P1	6443	6443	5891	552	552	0,00	6443
ORM_P2	5228	5228	4224	1004	1004	0,00	5228
ORM_P3	5368	5368	4276	1092	1092	0,00	5368
ORM_P4	5123	5123	3661	1462	1462	0,01	5123
ORM_P5	—	3663	2269	1394	1394	0,00	3663
ORM_P6	8180	8185	7867	318	318	0,00	8185
ORM_P7	6217	6217	5689	528	528	0,01	6217
ORM_P8	—	5865	4751	1114	1114	0,01	5865
ORM_P9	—	4699	3285	1414	1414	0,01	4699
ORM_P10	—	3457	2193	1264	1264	0,00	3457
ORM_P11	7855	7855	7717	138	138	0,01	7855
ORM_P12	7074	7074	6658	416	416	0,01	7074
ORM_P13	—	5539	4613	926	926	0,02	5539
ORM_P14	—	4981	3447	1534	1534	0,02	4981
ORM_P15	—	4477	2781	1696	1696	0,02	4477
ORM_P16	8292	8292	8172	120	120	0,03	8292
ORM_P17	7279	7279	7009	270	270	0,02	7279
ORM_P18	—	5982	4980	1002	1002	0,03	5982
ORM_P19	—	4787	3223	1564	1564	0,06	4787
ORM_P20	—	4860	2828	2032	2032	0,05	4860
ORM_P21	9272	9272	9138	134	134	0,03	9272
ORM_P22	8839	8839	8579	260	260	0,02	8839
ORM_P23	—	5799	4805	994	992	0,08	5797
ORM_P24	—	4983	3475	1508	1508	0,09	4983
ORM_P25	—	4808	2994	1814	1814	0,10	4808
ORM_P26	10022	10022	9924	98	98	0,05	10022
ORM_P27	8553	8555	8329	226	224	0,11	8553
ORM_P28	—	5599	4653	946	944	0,10	5597
ORM_P29	—	5079	3497	1582	1580	0,12	5077
ORM_P30	—	5133	3131	2002	2002	0,11	5133
ORM_P31	10185	10185	10087	98	98	0,05	10185
ORM_P32	—	9509	9301	208	200	0,16	9501
ORM_P33	—	5921	4835	1086	1082	0,13	5917
ORM_P34	—	5100	3608	1492	1490	0,16	5098
ORM_P35	—	10485	10401	84	84	0,07	10485
ORM_P36	—	10157	9953	204	204	0,08	10157
ORM_P37	—	6404	5222	1182	1180	0,18	6402
ORM_P38	—	11156	11060	96	96	0,18	11156
ORM_P39	—	9591	9423	168	158	0,11	9581
ORM_P40	—	6438	5342	1096	1096	0,22	6438

Tabela 6.26: Resultados da pós-otimização para a heurística GRASPf_RC nas instâncias proporcionais da classe ORM_P ($w = 2$).

Instância	F_O	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_P1	6443	6443	5891	552	552	0,00	6443
ORM_P2	5228	5228	4224	1004	1004	0,00	5228
ORM_P3	5368	5368	4276	1092	1092	0,00	5368
ORM_P4	5123	5123	3661	1462	1462	0,01	5123
ORM_P5	—	3663	2269	1394	1394	0,00	3663
ORM_P6	8180	8185	7867	318	318	0,00	8185
ORM_P7	6217	6217	5689	528	528	0,01	6217
ORM_P8	—	5865	4751	1114	1114	0,01	5865
ORM_P9	—	4699	3285	1414	1414	0,01	4699
ORM_P10	—	3457	2193	1264	1264	0,00	3457
ORM_P11	7855	7855	7717	138	138	0,01	7855
ORM_P12	7074	7074	6658	416	416	0,01	7074
ORM_P13	—	5539	4613	926	926	0,02	5539
ORM_P14	—	4981	3441	1540	1540	0,02	4981
ORM_P15	—	4477	2781	1696	1696	0,02	4477
ORM_P16	8292	8292	8172	120	120	0,03	8292
ORM_P17	7279	7279	7009	270	270	0,02	7279
ORM_P18	—	5984	4980	1004	1004	0,03	5984
ORM_P19	—	4787	3235	1552	1552	0,06	4787
ORM_P20	—	4860	2828	2032	2032	0,05	4860
ORM_P21	9272	9272	9138	134	134	0,03	9272
ORM_P22	8839	8839	8579	260	260	0,02	8839
ORM_P23	—	5799	4797	1002	1000	0,08	5797
ORM_P24	—	4982	3512	1470	1470	0,09	4982
ORM_P25	—	4808	2994	1814	1814	0,10	4808
ORM_P26	10022	10022	9924	98	98	0,05	10022
ORM_P27	8553	8555	8329	226	224	0,11	8553
ORM_P28	—	5598	4652	946	944	0,09	5596
ORM_P29	—	5079	3491	1588	1586	0,13	5077
ORM_P30	—	5133	3153	1980	1980	0,11	5133
ORM_P31	10185	10185	10087	98	98	0,05	10185
ORM_P32	—	9509	9301	208	200	0,16	9501
ORM_P33	—	5919	4825	1094	1090	0,21	5915
ORM_P34	—	5099	3583	1516	1516	0,17	5099
ORM_P35	—	10485	10401	84	84	0,07	10485
ORM_P36	—	10157	9953	204	204	0,08	10157
ORM_P37	—	6402	5186	1216	1214	0,32	6400
ORM_P38	—	11156	11060	96	96	0,18	11156
ORM_P39	—	9591	9423	168	158	0,11	9581
ORM_P40	—	6437	5339	1098	1098	0,22	6437

Tabela 6.27: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias proporcionais da classe ORM_P ($w = 2$).

Instância	F_O	F_G^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_P1	7184	7184	6059	1125	1125	0,01	7184
ORM_P2	6572	6572	4992	1580	1580	0,00	6572
ORM_P3	6776	6776	5021	1755	1755	0,00	6776
ORM_P4	—	6944	4549	2395	2395	0,00	6944
ORM_P5	—	5292	2907	2385	2385	0,00	5292
ORM_P6	—	8662	7872	790	790	0,00	8662
ORM_P7	—	6977	5822	1155	1130	0,00	6952
ORM_P8	—	7203	5148	2055	2055	0,01	7203
ORM_P9	—	6436	4221	2215	2215	0,00	6436
ORM_P10	—	5004	3054	1950	1950	0,01	5004
ORM_P11	8062	8062	7717	345	345	0,01	8062
ORM_P12	—	7666	6796	870	870	0,01	7666
ORM_P13	—	6693	5238	1455	1455	0,01	6693
ORM_P14	—	6632	4392	2240	2240	0,00	6632
ORM_P15	—	6363	3708	2655	2655	0,02	6363
ORM_P16	8458	8458	8188	270	270	0,02	8458
ORM_P17	—	7676	7021	655	655	0,01	7676
ORM_P18	—	7259	5444	1815	1815	0,02	7259
ORM_P19	—	6537	4362	2175	2175	0,03	6537
ORM_P20	—	7256	3951	3305	3305	0,04	7256
ORM_P21	9473	9473	9138	335	335	0,04	9473
ORM_P22	—	9219	8594	625	625	0,03	9219
ORM_P23	—	7020	5325	1695	1695	0,05	7020
ORM_P24	—	6648	4418	2230	2230	0,06	6648
ORM_P25	—	6897	4037	2860	2860	0,07	6897
ORM_P26	10169	10169	9924	245	245	0,00	10169
ORM_P27	—	8893	8338	555	550	0,05	8888
ORM_P28	—	6791	5146	1645	1645	0,07	6791
ORM_P29	—	6739	4694	2045	2045	0,09	6739
ORM_P30	—	7485	4035	3450	3450	0,12	7485
ORM_P31	—	10332	10087	245	245	0,04	10332
ORM_P32	—	9821	9306	515	495	0,16	9801
ORM_P33	—	7224	5364	1860	1860	0,10	7224
ORM_P34	—	6771	4721	2050	2050	0,13	6771
ORM_P35	—	10610	10405	205	205	0,06	10610
ORM_P36	—	10452	9972	480	475	0,09	10447
ORM_P37	—	7897	5772	2125	2125	0,14	7897
ORM_P38	—	11300	11060	240	240	0,18	11300
ORM_P39	—	9839	9429	410	405	0,11	9834
ORM_P40	—	7836	5891	1945	1945	0,16	7836

Tabela 6.28: Resultados da pós-otimização para a heurística GRASPf_RC nas instâncias proporcionais da classe ORM_P ($w = 5$).

Instância	F_O	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_P1	7184	7184	6059	1125	1125	0,01	7184
ORM_P2	6572	6572	4992	1580	1580	0,00	6572
ORM_P3	6776	6776	5021	1755	1755	0,00	6776
ORM_P4	—	6944	4549	2395	2395	0,00	6944
ORM_P5	—	5292	2907	2385	2385	0,00	5292
ORM_P6	—	8662	7872	790	790	0,00	8662
ORM_P7	—	6977	5822	1155	1130	0,00	6952
ORM_P8	—	7203	5148	2055	2055	0,01	7203
ORM_P9	—	6436	4221	2215	2215	0,00	6436
ORM_P10	—	5004	3054	1950	1950	0,01	5004
ORM_P11	8062	8062	7717	345	345	0,01	8062
ORM_P12	—	7666	6796	870	870	0,01	7666
ORM_P13	—	6693	5238	1455	1455	0,01	6693
ORM_P14	—	6632	4392	2240	2240	0,00	6632
ORM_P15	—	6363	3708	2655	2655	0,02	6363
ORM_P16	8458	8458	8188	270	270	0,02	8458
ORM_P17	—	7676	7021	655	655	0,01	7676
ORM_P18	—	7259	5444	1815	1815	0,02	7259
ORM_P19	—	6537	4362	2175	2175	0,03	6537
ORM_P20	—	7258	3893	3365	3365	0,06	7258
ORM_P21	9473	9473	9138	335	335	0,04	9473
ORM_P22	—	9219	8594	625	625	0,03	9219
ORM_P23	—	7020	5325	1695	1695	0,05	7020
ORM_P24	—	6648	4418	2230	2230	0,06	6648
ORM_P25	—	6897	4037	2860	2860	0,07	6897
ORM_P26	10169	10169	9924	245	245	0,00	10169
ORM_P27	—	8893	8333	560	555	0,06	8888
ORM_P28	—	6791	5146	1645	1645	0,07	6791
ORM_P29	—	6739	4694	2045	2045	0,09	6739
ORM_P30	—	7486	4016	3470	3470	0,12	7486
ORM_P31	—	10332	10087	245	245	0,04	10332
ORM_P32	—	9821	9306	515	495	0,16	9801
ORM_P33	—	7224	5374	1850	1850	0,10	7224
ORM_P34	—	6771	4711	2060	2060	0,13	6771
ORM_P35	—	10610	10405	205	205	0,06	10610
ORM_P36	—	10452	9972	480	475	0,09	10447
ORM_P37	—	7897	5812	2085	2085	0,14	7897
ORM_P38	—	11300	11060	240	240	0,18	11300
ORM_P39	—	9839	9429	410	405	0,11	9834
ORM_P40	—	7836	5891	1945	1945	0,16	7836

Tabela 6.29: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias proporcionais da classe ORM_P ($w = 5$).

Instância	F_O	F_G^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_P1	8146	8146	6846	1300	1300	0,00	8146
ORM_P2	7706	7706	6606	1100	1100	0,00	7706
ORM_P3	8265	8265	5765	2500	2500	0,00	8265
ORM_P4	—	9135	5115	4020	4020	0,00	9135
ORM_P5	—	7485	3345	4140	4140	0,00	7485
ORM_P6	—	9387	8037	1350	1350	0,00	9387
ORM_P7	—	7878	6468	1410	1410	0,00	7878
ORM_P8	—	8632	6692	1940	1940	0,00	8632
ORM_P9	—	8174	5344	2830	2830	0,01	8174
ORM_P10	—	6894	3204	3690	3690	0,01	6894
ORM_P11	8383	8383	7753	630	630	0,00	8383
ORM_P12	—	8426	7046	1380	1380	0,01	8426
ORM_P13	—	7679	6249	1430	1430	0,00	7679
ORM_P14	—	8505	5165	3340	3340	0,01	8505
ORM_P15	—	8864	4024	4840	4840	0,02	8864
ORM_P16	8728	8728	8188	540	540	0,01	8728
ORM_P17	—	8317	7127	1190	1190	0,01	8317
ORM_P18	—	8547	6567	1980	1980	0,02	8547
ORM_P19	—	8313	5013	3300	3300	0,03	8313
ORM_P20	—	10386	4316	6070	6070	0,04	10386
ORM_P21	9808	9808	9138	670	670	0,04	9808
ORM_P22	—	9776	8706	1070	1070	0,02	9776
ORM_P23	—	8359	6179	2180	2180	0,04	8359
ORM_P24	—	8548	5148	3400	3400	0,00	8548
ORM_P25	—	9610	4400	5210	5210	0,07	9610
ORM_P26	—	10414	9924	490	490	0,06	10414
ORM_P27	—	9317	8557	760	760	0,03	9317
ORM_P28	—	8077	6037	2040	2040	0,04	8077
ORM_P29	—	8552	5252	3300	3300	0,07	8552
ORM_P30	—	10735	4445	6290	6290	0,11	10735
ORM_P31	—	10556	10126	430	430	0,04	10556
ORM_P32	—	10304	9434	870	870	0,05	10304
ORM_P33	—	8666	6366	2300	2300	0,05	8666
ORM_P34	—	8551	5201	3350	3350	0,12	8551
ORM_P35	—	10811	10491	320	320	0,05	10811
ORM_P36	—	10898	10028	870	870	0,06	10898
ORM_P37	—	9575	6845	2730	2730	0,12	9575
ORM_P38	—	11525	11105	420	420	0,08	11525
ORM_P39	—	10220	9480	740	740	0,08	10220
ORM_P40	—	9380	6840	2540	2540	0,12	9380

Tabela 6.30: Resultados da pós-otimização para a heurística GRASPF_RC nas instâncias proporcionais da classe ORM_P ($w = 10$).

Instância	F_O	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_P1	8146	8146	6846	1300	1300	0,00	8146
ORM_P2	7706	7706	6606	1100	1100	0,00	7706
ORM_P3	8265	8265	5765	2500	2500	0,00	8265
ORM_P4	—	9135	5115	4020	4020	0,00	9135
ORM_P5	—	7485	3345	4140	4140	0,00	7485
ORM_P6	—	9387	8037	1350	1350	0,00	9387
ORM_P7	—	7878	6468	1410	1410	0,00	7878
ORM_P8	—	8632	6692	1940	1940	0,00	8632
ORM_P9	—	8174	5344	2830	2830	0,01	8174
ORM_P10	—	6894	3204	3690	3690	0,01	6894
ORM_P11	8383	8383	7753	630	630	0,00	8383
ORM_P12	—	8426	7046	1380	1380	0,01	8426
ORM_P13	—	7679	6249	1430	1430	0,00	7679
ORM_P14	—	8505	5165	3340	3340	0,01	8505
ORM_P15	—	8864	4024	4840	4840	0,02	8864
ORM_P16	8728	8728	8188	540	540	0,01	8728
ORM_P17	—	8317	7127	1190	1190	0,01	8317
ORM_P18	—	8547	6567	1980	1980	0,02	8547
ORM_P19	—	8313	5013	3300	3300	0,03	8313
ORM_P20	—	10397	4387	6010	6010	0,07	10397
ORM_P21	9808	9808	9138	670	670	0,04	9808
ORM_P22	—	9776	8706	1070	1070	0,02	9776
ORM_P23	—	8359	6159	2200	2200	0,04	8359
ORM_P24	—	8548	5148	3400	3400	0,00	8548
ORM_P25	—	9610	4400	5210	5210	0,08	9610
ORM_P26	—	10414	9924	490	490	0,06	10414
ORM_P27	—	9317	8557	760	760	0,03	9317
ORM_P28	—	8077	6037	2040	2040	0,04	8077
ORM_P29	—	8552	5252	3300	3300	0,07	8552
ORM_P30	—	10735	4445	6290	6290	0,13	10735
ORM_P31	—	10556	10126	430	430	0,04	10556
ORM_P32	—	10304	9434	870	870	0,05	10304
ORM_P33	—	8666	6366	2300	2300	0,05	8666
ORM_P34	—	8551	5201	3350	3350	0,12	8551
ORM_P35	—	10811	10491	320	320	0,05	10811
ORM_P36	—	10898	10028	870	870	0,06	10898
ORM_P37	—	9575	6825	2750	2750	0,12	9575
ORM_P38	—	11525	11105	420	420	0,08	11525
ORM_P39	—	10220	9480	740	740	0,08	10220
ORM_P40	—	9380	6840	2540	2540	0,12	9380

Tabela 6.31: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias proporcionais da classe ORM_P ($w = 10$).

Instância	F_O	F_G^*/F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
GRM_P1	6391	6403	5703	700	688	0,09	6391
GRM_P2	5475	5489	4707	782	782	0,02	5489
GRM_P3	5157	5157	3971	1186	1186	0,01	5157
GRM_P4	5246	5246	3648	1598	1598	0,03	5246
GRM_P5	—	5403	3391	2012	2012	0,02	5403
GRM_P6	—	5580	3152	2428	2428	0,01	5580
GRM_P7	—	5766	2898	2868	2868	0,02	5766
GRM_P8	—	5961	2663	3298	3298	0,02	5961
GRM_P9	—	6364	2260	4104	4104	0,02	6364
GRM_P10	11413	11475	10935	540	530	0,06	11465
GRM_P11	9823	9862	9046	816	816	0,07	9862
GRM_P12	8867	8910	7664	1246	1246	0,07	8910
GRM_P13	—	8352	6636	1716	1716	0,07	8352
GRM_P14	—	8174	6070	2104	2104	0,08	8174
GRM_P15	—	8133	5623	2510	2510	0,08	8133
GRM_P16	—	8243	5325	2918	2918	0,08	8243
GRM_P17	—	8372	5042	3330	3330	0,08	8372
GRM_P18	—	8530	4816	3714	3714	0,06	8530
GRM_P19	—	8701	4581	4120	4120	0,05	8701
GRM_P20	—	9080	4164	4916	4916	0,06	9080

Tabela 6.32: Resultados da pós-otimização para as heurísticas GRASPf_RC e VNSf_RC nas instâncias proporcionais da classe GRM_P ($w = 2$).

6.5

Considerações Finais

Nesse capítulo, as heurísticas GRASP e VNS com filtro e reconexão por caminhos (GRASPf_RC e VNSf_RC, respectivamente) foram comparadas de três formas diferentes.

A qualidade das soluções encontradas pelas heurísticas para as 249 instâncias testadas (189 proporcionais e 60 não proporcionais) foram analisadas em três tempos de processamento distintos: tp , ti e tg . Utilizando-se na avaliação as medidas $drpm$, cm e $melhor$, em geral, o GRASPf_RC se mostrou superior ao VNSf_RC nas instâncias proporcionais. Já nas instâncias não proporcionais, o VNSf_RC apresentou resultados melhores. A diferença nas medidas torna-se menor quando aumenta-se o tempo de processamento, com as heurísticas convergindo praticamente para as mesmas soluções nas instâncias proporcionais e não proporcionais nos maiores tempos de processamento (ti e tg).

Em seguida, gráficos do tempo gasto para atingir o valor alvo foram utilizados na comparação. Dos problemas analisados, GRASPf_RC apresentou resultados melhores do que VNSf_RC nas instâncias proporcionais com fator w maior ($w = 5$ e $w = 10$) e nas instâncias não proporcionais GRM_NP.

Instância	F_O	F_G^*/F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
GRM_P1	7036	7036	6181	855	855	0,01	7036
GRM_P2	6662	6662	4707	1955	1955	0,02	6662
GRM_P3	6923	6923	4018	2905	2905	0,01	6923
GRM_P4	7602	7602	3707	3895	3895	0,02	7602
GRM_P5	—	8379	3464	4915	4915	0,02	8379
GRM_P6	—	9178	3263	5915	5915	0,01	9178
GRM_P7	—	9984	3059	6925	6925	0,02	9984
GRM_P8	—	10802	2832	7970	7970	0,01	10802
GRM_P9	—	12447	2357	10090	10090	0,03	12447
GRM_P10	12202	12202	11297	905	905	0,04	12202
GRM_P11	11086	11086	9046	2040	2040	0,07	11086
GRM_P12	10767	10767	7672	3095	3095	0,06	10767
GRM_P13	—	10902	6742	4160	4160	0,05	10902
GRM_P14	—	11293	6118	5175	5175	0,06	11293
GRM_P15	—	11874	5734	6140	6140	0,06	11874
GRM_P16	—	12539	5479	7060	7060	0,05	12539
GRM_P17	—	13276	5181	8095	8095	0,06	13276
GRM_P18	—	14036	4901	9135	9135	0,06	14036
GRM_P19	—	14814	4674	10140	10140	0,06	14814
GRM_P20	—	16422	4222	12200	12200	0,07	16422

Tabela 6.33: Resultados da pós-otimização para as heurísticas GRASPf_RC e VNSf_RC nas instâncias proporcionais da classe GRM_P ($w = 5$).

Instância	F_O	F_G^*/F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
GRM_P1	7884	7884	6224	1660	1660	0,01	7884
GRM_P2	8580	8580	4790	3790	3790	0,02	8580
GRM_P3	9793	9793	4093	5700	5700	0,01	9793
GRM_P4	11487	11487	3757	7730	7730	0,02	11487
GRM_P5	—	13273	3513	9760	9760	0,02	13273
GRM_P6	—	15079	3309	11770	11770	0,01	15079
GRM_P7	—	16909	3079	13830	13830	0,01	16909
GRM_P8	—	18754	2854	15900	15900	0,02	18754
GRM_P9	—	22507	2397	20110	20110	0,03	22507
GRM_P10	13083	13083	11423	1660	1660	0,02	13083
GRM_P11	13040	13040	9190	3850	3850	0,03	13040
GRM_P12	13845	13845	7775	6070	6070	0,06	13845
GRM_P13	—	14985	7025	7960	7960	0,05	14985
GRM_P14	—	16355	6395	9960	9960	0,04	16355
GRM_P15	—	17919	5919	12000	12000	0,06	17919
GRM_P16	—	19579	5549	14030	14030	0,05	19579
GRM_P17	—	21317	5297	16020	16020	0,05	21317
GRM_P18	—	23095	5035	18060	18060	0,05	23095
GRM_P19	—	24895	4775	20120	20120	0,06	24895
GRM_P20	—	28566	4316	24250	24250	0,06	28566

Tabela 6.34: Resultados da pós-otimização para as heurísticas GRASPf_RC e VNSf_RC nas instâncias proporcionais da classe GRM_P ($w = 10$).

Instância	F_O	F_G^*/F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
SLM_P1	—	4851	3065	1786	1786	0,19	4851
SLM_P2	—	5317	3291	2026	2026	0,23	5317
SLM_P3	—	5409	3281	2128	2128	0,32	5409

Tabela 6.35: Resultados da pós-otimização para a heurística GRASPf_RC nas instâncias proporcionais da classe SLM_P ($w = 2$).

Instância	F_O	F_G^*/F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
SLM_P1	—	4851	3051	1800	1800	0,19	4851
SLM_P2	—	5317	3297	2020	2020	0,23	5317
SLM_P3	—	5409	3325	2084	2084	0,32	5409

Tabela 6.36: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias proporcionais da classe SLM_P ($w = 2$).

Instância	F_O	F_G^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
SLM_P1	—	6961	3986	2975	2975	0,15	6961
SLM_P2	—	7840	4205	3635	3635	0,23	7840
SLM_P3	—	7880	4315	3565	3565	0,28	7880

Tabela 6.37: Resultados da pós-otimização para a heurística GRASPf_RC nas instâncias proporcionais da classe SLM_P ($w = 5$).

Instância	F_O	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
SLM_P1	—	6961	3986	2975	2975	0,16	6961
SLM_P2	—	7841	4196	3645	3645	0,23	7841
SLM_P3	—	7885	4325	3560	3560	0,29	7885

Tabela 6.38: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias proporcionais da classe SLM_P ($w = 5$).

Instância	F_O	F_G^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
SLM_P1	—	9767	4297	5470	5470	0,16	9767
SLM_P2	—	11304	4574	6730	6730	0,19	11304
SLM_P3	—	11273	4683	6590	6590	0,30	11273

Tabela 6.39: Resultados da pós-otimização para a heurística GRASPf_RC nas instâncias proporcionais da classe SLM_P ($w = 10$).

Instância	F_O	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
SLM_P1	—	9788	4268	5520	5520	0,16	9788
SLM_P2	—	11319	4569	6750	6750	0,19	11319
SLM_P3	—	11294	4654	6640	6640	0,30	11294

Tabela 6.40: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias não proporcionais da classe SLM_P ($w = 10$).

Instância	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_NP1	6447	5893	554	554	0,00	6447
ORM_NP2	5196	4128	1068	1068	0,00	5196
ORM_NP3	5235	4368	867	867	0,00	5235
ORM_NP4	4935	3458	1477	1477	0,01	4935
ORM_NP5	3377	1900	1477	1477	0,01	3377
ORM_NP6	8098	7824	274	274	0,01	8098
ORM_NP7	6153	5641	512	512	0,01	6153
ORM_NP8	5629	4591	1038	1034	0,02	5625
ORM_NP9	4364	3107	1257	1257	0,01	4364
ORM_NP10	3362	1999	1363	1363	0,01	3362
ORM_NP11	7832	7717	115	115	0,00	7832
ORM_NP12	6979	6658	321	321	0,02	6979
ORM_NP13	5342	4485	857	857	0,02	5342
ORM_NP14	4781	3432	1349	1348	0,04	4780
ORM_NP15	4120	2440	1680	1680	0,04	4120
ORM_NP16	8293	8172	121	115	0,02	8287
ORM_NP17	7246	7010	236	236	0,03	7246
ORM_NP18	5838	4962	876	868	0,04	5830
ORM_NP19	4546	3179	1367	1366	0,06	4545
ORM_NP20	4428	2650	1778	1778	0,05	4428
ORM_NP21	9238	9138	100	100	0,02	9238
ORM_NP22	8800	8583	217	214	0,03	8797
ORM_NP23	5629	4712	917	916	0,08	5628
ORM_NP24	4678	3364	1314	1314	0,09	4678
ORM_NP25	4538	2749	1789	1789	0,11	4538
ORM_NP26	9996	9917	79	79	0,03	9996
ORM_NP27	8514	8307	207	205	0,09	8512
ORM_NP28	5443	4650	793	790	0,10	5440
ORM_NP29	4801	3407	1394	1393	0,14	4800
ORM_NP30	4700	2898	1802	1802	0,15	4700
ORM_NP31	10158	10086	72	72	0,06	10158
ORM_NP32	9474	9297	177	177	0,10	9474
ORM_NP33	5723	4806	917	915	0,17	5721
ORM_NP34	4759	3446	1313	1313	0,18	4759
ORM_NP35	10463	10400	63	63	0,06	10463
ORM_NP36	10116	9934	182	178	0,14	10112
ORM_NP37	6190	5168	1022	1014	0,20	6182
ORM_NP38	11129	11060	69	69	0,11	11129
ORM_NP39	9551	9423	128	128	0,19	9551
ORM_NP40	6213	5293	920	919	0,67	6212

Tabela 6.41: Resultados da pós-otimização para a heurística GRASPf_RC nas instâncias não proporcionais da classe ORM_NP ($w = 2$).

Instância	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
ORM_NP1	6447	5893	554	554	0,00	6447
ORM_NP2	5196	4128	1068	1068	0,00	5196
ORM_NP3	5235	4368	867	867	0,00	5235
ORM_NP4	4930	3381	1549	1549	0,01	4930
ORM_NP5	3377	1900	1477	1477	0,01	3377
ORM_NP6	8098	7824	274	274	0,01	8098
ORM_NP7	6153	5641	512	512	0,01	6153
ORM_NP8	5623	4533	1090	1090	0,02	5623
ORM_NP9	4364	3107	1257	1257	0,01	4364
ORM_NP10	3362	1999	1363	1363	0,01	3362
ORM_NP11	7832	7717	115	115	0,00	7832
ORM_NP12	6979	6658	321	321	0,02	6979
ORM_NP13	5342	4485	857	857	0,02	5342
ORM_NP14	4781	3432	1349	1348	0,04	4780
ORM_NP15	4120	2409	1711	1711	0,05	4120
ORM_NP16	8293	8172	121	115	0,02	8287
ORM_NP17	7246	7010	236	236	0,03	7246
ORM_NP18	5836	4948	888	880	0,04	5828
ORM_NP19	4546	3188	1358	1357	0,07	4545
ORM_NP20	4428	2650	1778	1778	0,07	4428
ORM_NP21	9238	9138	100	100	0,02	9238
ORM_NP22	8800	8583	217	214	0,03	8797
ORM_NP23	5628	4698	930	929	0,08	5627
ORM_NP24	4678	3371	1307	1307	0,08	4678
ORM_NP25	4538	2749	1789	1789	0,11	4538
ORM_NP26	9996	9917	79	79	0,03	9996
ORM_NP27	8514	8307	207	205	0,09	8512
ORM_NP28	5439	4634	805	803	0,17	5437
ORM_NP29	4801	3423	1378	1377	0,12	4800
ORM_NP30	4700	2898	1802	1802	0,15	4700
ORM_NP31	10158	10086	72	72	0,06	10158
ORM_NP32	9474	9297	177	177	0,10	9474
ORM_NP33	5718	4803	915	913	0,21	5716
ORM_NP34	4759	3465	1294	1293	0,18	4758
ORM_NP35	10463	10400	63	63	0,06	10463
ORM_NP36	10116	9934	182	178	0,14	10112
ORM_NP37	6190	5159	1031	1023	3,46	6182
ORM_NP38	11129	11060	69	69	0,11	11129
ORM_NP39	9551	9423	128	128	0,19	9551
ORM_NP40	6213	5295	918	917	0,64	6212

Tabela 6.42: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias não proporcionais da classe ORM_NP ($w = 2$).

Instância	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
GRM_NP1	6156	5703	453	434	0,03	6137
GRM_NP2	5185	4498	687	687	0,02	5185
GRM_NP3	4825	3986	839	839	0,03	4825
GRM_NP4	4720	3708	1012	1012	0,02	4720
GRM_NP5	4675	3418	1257	1257	0,01	4675
GRM_NP6	4675	3161	1514	1514	0,02	4675
GRM_NP7	4685	2915	1770	1770	0,03	4685
GRM_NP8	4717	2688	2029	2029	0,02	4717
GRM_NP9	4806	2233	2573	2573	0,03	4806
GRM_NP10	11334	10841	493	477	0,09	11318
GRM_NP11	9710	8752	958	958	0,12	9710
GRM_NP12	8751	7640	1111	1111	0,11	8751
GRM_NP13	8135	6801	1334	1334	0,09	8135
GRM_NP14	7719	6181	1538	1538	0,06	7719
GRM_NP15	7527	5748	1779	1779	0,10	7527
GRM_NP16	7427	5413	2014	2014	0,06	7427
GRM_NP17	7362	5123	2239	2239	0,08	7362
GRM_NP18	7341	4865	2476	2476	0,08	7341
GRM_NP19	7346	4632	2714	2714	0,07	7346
GRM_NP20	7405	4123	3282	3282	0,08	7405

Tabela 6.43: Resultados da pós-otimização para a heurística GRASPf_RC nas instâncias não proporcionais da classe GRM_NP ($w = 2$).

Instância	F_V^*	F_P	F_T	F_{OT}	Tempo (s)	F_{PO}
GRM_NP1	6156	5703	453	434	0,03	6137
GRM_NP2	5185	4498	687	687	0,02	5185
GRM_NP3	4825	3986	839	839	0,03	4825
GRM_NP4	4720	3708	1012	1012	0,02	4720
GRM_NP5	4675	3418	1257	1257	0,01	4675
GRM_NP6	4675	3161	1514	1514	0,02	4675
GRM_NP7	4685	2915	1770	1770	0,03	4685
GRM_NP8	4717	2688	2029	2029	0,02	4717
GRM_NP9	4806	2233	2573	2573	0,03	4806
GRM_NP10	11334	10841	493	477	0,09	11318
GRM_NP11	9710	8752	958	958	0,12	9710
GRM_NP12	8759	7613	1146	1146	0,07	8759
GRM_NP13	8135	6801	1334	1334	0,09	8135
GRM_NP14	7719	6181	1538	1538	0,06	7719
GRM_NP15	7527	5748	1779	1779	0,10	7527
GRM_NP16	7431	5416	2015	72015	0,08	7431
GRM_NP17	7360	5126	2234	2234	0,06	7360
GRM_NP18	7341	4865	2476	2476	0,08	7341
GRM_NP19	7346	4632	2714	2714	0,07	7346
GRM_NP20	7405	4123	3282	3282	0,08	7405

Tabela 6.44: Resultados da pós-otimização para a heurística VNSf_RC nas instâncias não proporcionais da classe GRM_NP ($w = 2$).

Já o algoritmo VNSf_RC apresentou resultados melhores do que GRASPf_RC nas instâncias proporcionais com $w = 2$ e nas instâncias não proporcionais ORM_NP.

Por último, 49 soluções ótimas para problemas de pequeno porte foram obtidas e os resultados encontrados pelas heurísticas no tempo tg foram comparados com essas instâncias. Para os problemas que não possuem o ótimo, a melhor solução obtida pelos algoritmos no tempo tg foi a melhor solução encontrada. Levando-se em consideração os casos de empate no primeiro lugar, dos 249 problemas, GRASPf_RC apresentou a melhor solução em 234 instâncias e VNSf_RC apresentou a melhor solução em 236 instâncias. Em relação aos problemas com o ótimo conhecido, as heurísticas apresentaram os mesmos resultados, encontrando a solução ótima em 42 problemas, ficando, no máximo, a 0,54% do ótimo nas sete instâncias onde o mesmo não foi alcançado. Investigações adicionais mostraram que, em algumas instâncias, as facilidades abertas encontradas pelo algoritmo exato coincidem com as facilidades abertas encontradas pelas heurísticas. Assim, a árvore de Steiner obtida pelos algoritmos aproximados não é ótima, o que sugeriu um passo adicional de pós-otimização.

O passo de pós-otimização consistiu na execução de um algoritmo ótimo para o problema de Steiner, tendo como vértices teminais as facilidades abertas da melhor solução encontrada pelo GRASPf_RC ou pelo VNSf_RC após todas as iterações terem sido realizadas nas cinco execuções no tempo tg . O algoritmo exato encontrou a árvore de Steiner ótima da melhor solução encontrada pelas heurísticas em menos de um segundo para todas as instâncias testadas. Os resultados da pós-otimização mostraram a eficiência da heurística Prim que obteve a grande maioria das solução ótimas. Acrescentando o passo de pós-otimização, GRASPf_RC e VNSf_RC encontraram duas novas soluções ótimas, totalizando 44 problemas e, ficando, no máximo a 0,48% do valor ótimo nas cinco instâncias onde o mesmo não foi alcançado. Nas instâncias em que não se conhece o ótimo, GRASP e VNS, acrescidas da pós-otimização, conseguiram melhorar a melhor solução encontrada pelos algoritmos sem o procedimento em 29 e 28 instâncias, respectivamente. Levando-se em consideração os casos de empate no primeiro lugar e incluindo a pós-otimização, dos 249 problemas, GRASPf_RC apresentou a melhor solução em 239 instâncias e VNSf_RC apresentou a melhor solução em 237 instâncias.