

5

VNS com Filtro e Reconexão por Caminhos

A metaheurística VNS (*Variable Neighborhood Search*) foi proposta por Mladenović e Hansen [40] e possui como idéia básica a mudança de vizinhanças realizada da seguinte maneira: aplica-se um algoritmo de busca local, explorando vizinhanças próximas da solução corrente até que nenhuma solução melhor seja encontrada. Na tentativa de fugir de ótimos locais, aumenta-se a vizinhança, buscando-se soluções mais distantes da solução corrente.

As vizinhanças são exploradas em ordem crescente de distância da solução corrente. Para a construção de diferentes estruturas de vizinhanças, define-se uma métrica de distância entre duas soluções quaisquer, criando-se as vizinhanças baseadas nessa métrica [27].

Essa metaheurística foi escolhida devido às recentes aplicações de sucesso apresentadas na literatura [2, 11]. A solução inicial do VNS para o problema das p -medianas conectadas será gerada, como no GRASP com filtro e reconexão por caminhos, pela heurística construtiva descrita na Seção 4.1 com $\alpha = 0,4$ e o algoritmo de busca local utilizado será aquele descrito no Capítulo 3 (BL_Conc).

A implementação de uma metaheurística envolve um balanceamento entre tempo de processamento e qualidade da solução. Com o objetivo de diminuir os tempos de processamento obtidos pelo VNS, a estratégia de filtro apresentada na Seção 4.4 será utilizada entre a busca local pelas bordas com teste da menor aresta e a busca local com teste. Por outro lado, para melhorar a qualidade das soluções encontradas, uma simples modificação no VNS será incorporada, perturbando-se a solução corrente mais de uma vez e escolhendo a melhor solução para aplicação da busca local (parâmetro b) e o procedimento de reconexão por caminhos será utilizado como estratégia de intensificação, como apresentado na Seção 4.5.

Este capítulo está organizado como se segue: a Seção 5.1 apresenta a descrição do algoritmo VNS em sua versão básica; a Seção 5.2 mostra o ambiente de teste, as instâncias e as medidas utilizadas na configuração do VNS; a Seção 5.3 apresenta o ajuste dos parâmetros da metaheurística; a Seção 5.4 descreve os resultados computacionais do VNS com o filtro em relação ao VNS em sua versão básica. A Seção 5.5 mostra a configuração do parâmetro b e a Seção 5.6 apresenta a configuração do procedimento de reconexão por caminhos e

resultados computacionais em relação ao VNS que não o utiliza. A Seção 5.7 descreve a heurística VNS com filtro e reconexão por caminhos proposta para o problema e, por último, a Seção 5.8 apresenta as conclusões e considerações finais do capítulo.

5.1

Algoritmo VNS Básico

Sejam duas soluções quaisquer formadas pelos conjuntos S e S' compostos por p facilidades abertas e por suas respectivas árvores de Steiner. Define-se a distância entre as duas soluções igual a k , se e somente se, S e S' diferem em k facilidades. Assim, a função de distância simétrica ρ pode ser definida como:

$$\rho(S, S') = |S \setminus S'| = |S' \setminus S|, \forall S, S'. \quad (5-1)$$

As estruturas de vizinhança utilizadas são induzidas por ρ , isto é, k facilidades pertencentes à solução corrente são substituídas por k facilidades da sua vizinhança ($k \leq p$). Seja N_k , $k = 1, \dots, k_{max}$, $k_{max} \leq p$, o conjunto de estruturas de vizinhanças selecionadas e $N_k(S)$ a vizinhança da solução S . Mais formalmente, $S' \in N_k(S) \Leftrightarrow \rho(S, S') = k$.

A Figura 5.1 apresenta o algoritmo VNS em sua versão básica para o problema das p -medianas conectadas.

```

Procedimento VNS_Básico( $\alpha$ , semente)
1.  $F^* \leftarrow +\infty$ ;
2.  $S \leftarrow$  Fase_Construção( $\alpha$ , semente);
3. Enquanto (critério de parada) Faça
4.    $k \leftarrow 1$ ;
5.   Enquanto ( $k \leq k_{max}$ ) Faça
6.     Gerar uma solução aleatória  $S' \in N_k(S)$ ;
7.      $S' \leftarrow$  BL_Conc ( $S'$ );
8.     Se ( $F(S') < F^*$ ) Então
9.        $F^* \leftarrow F(S')$ ;
10.       $S^* \leftarrow S'$ ;
11.       $S \leftarrow S'$ ;
12.       $k \leftarrow 1$ ;
13.     Fim-Se
14.     Senão
15.        $k \leftarrow k + 1$ ;
16.     Fim-Senão
17.   Fim-Enquanto
18. Fim-Enquanto
19. Retorne  $S^*$ ;
Fim

```

Figura 5.1: Algoritmo VNS básico.

A solução inicial é gerada na linha 2 como descrito na Seção 4.1. Executa-se o VNS (laço das linhas 3 a 18) enquanto o critério de parada não for satisfeito. O critério de parada pode ser o tempo de processamento, o número de iterações ou mesmo o número de iterações sem melhoria no valor da função objetivo. O algoritmo inicia explorando a vizinhança que possui a menor distância em relação à solução corrente (linha 4). Enquanto a maior estrutura de vizinhança não for alcançada (parâmetro k_{max}) no laço das linhas 5 a 17, os seguintes passos são executados. Na linha 6, gera-se aleatoriamente uma solução $S' \in N_k(S)$ na k -ésima vizinhança de S . Uma perturbação aleatória é utilizada com a finalidade de evitar ciclagem, o que poderia ocorrer se qualquer regra determinística fosse utilizada. A solução S' é então utilizada como solução inicial para o algoritmo de busca local (BL_Conc), na linha 7. A execução das linhas 6 e 7 constitui uma iteração no VNS. Se o valor da função objetivo do ótimo local é melhor do que o valor da função objetivo da melhor solução encontrada até o momento (linha 8), atualiza-se a melhor solução, a solução corrente e retorna-se à vizinhança N_1 (linhas 9 a 12). Caso contrário, aumenta-se a distância entre a solução corrente S e a nova solução S' que será gerada aleatoriamente (linha 15). A melhor solução encontrada pelo algoritmo é retornada na linha 19.

5.2

Ambiente de Teste, Instâncias e Medidas Utilizadas

A heurística VNS foi implementada em C com o parâmetro de otimização -O3 e executada 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 os testes da estratégia de filtro, ajuste do parâmetro b e configuração do procedimento de reconexão por caminhos, utilizou-se um sub-conjunto das instâncias proporcionais apresentadas na Seção 3.3.4, totalizando 36 problemas: ORM_P1 a ORM_P18 e GRM_P1 a GRM_P18 ($w = 5$). Para a avaliação da qualidade das soluções e dos tempos de processamento, as medidas relativas, descritas no Capítulo 3, foram utilizadas.

Para o ajuste do parâmetro k_{max} , utilizou-se um sub-conjunto maior de instâncias proporcionais apresentadas na Seção 3.3.4, totalizando-se 120 problemas: GRM_P1 a GRM_P20 e ORM_P1 a ORM_P20 ($w = 2$, $w = 5$ e $w = 10$). Para a avaliação da qualidade das soluções e dos tempos de processamento, considera-se, respectivamente, para cada instância, a média em termos de qualidade e tempo obtida pelo algoritmo em um determinado número de execuções.

Para todos os testes e ajuste de parâmetros, foram realizadas cinco execuções com diferentes sementes aleatórias para cada par instância-algoritmo.

5.3 Configuração dos Parâmetros do VNS

Os parâmetros a serem ajustados no VNS em sua versão básica são: critério de parada do algoritmo e número total de vizinhanças utilizadas, representado pela variável k_{max} na Figura 5.1.

O critério de parada escolhido para os testes de configuração do VNS foi o número de iterações realizadas. Uma iteração do VNS corresponde à execução de uma perturbação aleatória seguida de uma busca local. Quanto maior o número de iterações, maior a probabilidade de encontrar uma solução melhor. Porém, devido aos tempos de processamento elevados apresentados por BL_Conc nas maiores instâncias no Capítulo 3, o número de iterações foi estipulado em 100, um valor baixo em relação ao encontrado em algumas aplicações de VNS na literatura, como em [26].

O parâmetro que influencia a qualidade das soluções encontradas e os tempos de processamento obtidos pelo algoritmo VNS é o número total de vizinhanças utilizadas $k_{max} \leq p$. Com o objetivo de obter um compromisso entre qualidade e tempo, testes foram realizados com valores de $k_{max} = 4, 8, 10, 15, 20$ e p para o sub-conjunto de instâncias testadas. Nas Tabelas 5.3 a 5.14, os valores em negrito representam o menor valor obtido pelas configurações para cada instância.

Analisando-se a qualidade média das soluções encontradas pelas configurações do VNS (Tabelas 5.3 a 5.8), quanto maior o valor de k_{max} , maior a possibilidade de fuga dos ótimos locais e, conseqüentemente, aumenta-se a probabilidade de encontrar soluções melhores. Quando o valor de k_{max} é menor, aumenta-se a chance de convergência prematura nos ótimos locais. Somando-se o número de instâncias em que os algoritmos encontraram o menor valor entre todos os métodos, incluindo-se os casos de empate no primeiro lugar para as classes testadas (linha *melhor* nas tabelas), as variantes de VNS com $k_{max} = 4, 8, 10, 15, 20$ e p obtiveram os seguintes resultados, respectivamente: 88, 98, 96, 107, 106 e 102, de um total de 120 instâncias.

Já na análise dos tempos médios de processamento obtidos pelas configurações do VNS (Tabelas 5.9 a 5.14), ocorre o inverso. Valores menores de k_{max} obtêm resultados melhores, isto é, quanto menor o valor do parâmetro, mais rápida é a heurística. Somando-se o número de instâncias em que os algoritmos encontraram o menor valor entre todos os métodos, incluindo-se os casos de empate no primeiro lugar para as classes testadas (linha *melhor* nas tabelas), as variantes de VNS com $k_{max} = 4, 8, 10, 15, 20$ e p obtiveram os seguintes resultados, respectivamente: 106, 8, 4, 3, 1 e 0, de um total de 120 instâncias.

A configuração $k_{max} = 4$ obteve os melhores tempos de processamento, porém os piores resultados em termos de qualidade das soluções. Já a con-

figuração $k_{max} = p$ obteve soluções de qualidade comparável às melhores configurações ($k_{max} = 15$ e 20), porém, apresentou os piores tempos de processamento. Analisando-se qualidade e tempo, as configurações que obtiveram os melhores resultados foram $k_{max} = 15$ e 20 . Optou-se por utilizar a configuração $k_{max} = 15$.

5.4

Estratégia de Filtro

A heurística GRASP, proposta no Capítulo 4, utilizou uma estratégia de filtro com o objetivo de acelerar as iterações do algoritmo. Com a mesma finalidade, o VNS utilizará a mesma estratégia entre a busca local pelas bordas com teste da menor aresta e a busca local com teste, explicada na Seção 4.4, pois BL_Conc apresentou tempos de processamento elevados nas maiores instâncias testadas, conforme mostrado no Capítulo 3.

Em geral, a utilização do filtro reduz o tempo de processamento às custas de perdas na qualidade média das soluções encontradas. Para verificar essa hipótese, testes foram realizados comparando-se o VNS sem filtro (VNSb) com o VNS com filtro (VNSf). A Tabela 5.1 apresenta as medidas relativas $drpm$ e cm utilizadas na análise da qualidade das soluções encontradas pelo VNSb e VNSf.

	VNSb	VNSf
$drpm$	-0,0019	0,0019
cm	1,42	1,58

Tabela 5.1: Qualidade relativa na configuração do filtro no VNS.

Analisando-se a qualidade das soluções encontradas nas medidas $drpm$ e cm , como esperado, o VNSf perde em qualidade das soluções comparando-se com o VNSb. Em números absolutos, das 180 execuções realizadas (36 instâncias com cinco execuções cada), VNSf piora a solução encontrada pelo VNSb somente em 13 execuções.

A Tabela 5.2 apresenta as medidas relativas $drpm$, cm e trm utilizadas na análise dos tempos de processamento obtidos pelo VNSb e VNSf. Nessa última medida, o VNSb foi utilizado como algoritmo de referência.

	VNSb	VNSf
$drpm$	39,07	-39,07
cm	1,97	1,03
trm	1,00	0,47

Tabela 5.2: Tempo relativo na configuração do filtro no VNS.

As três medidas relativas mostraram que, ao se utilizar o filtro no VNS, ganhos em tempos de processamento foram obtidos em relação ao VNSb. Na medida trm , o VNSf gastou, em média, 47% do tempo de execução do VNSb.

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
ORM_P1	6443,00	6443,00	6443,00	6443,00	6443,00	6443,00
ORM_P2	5228,00	5228,00	5228,00	5228,00	5228,00	5228,00
ORM_P3	5368,00	5368,00	5368,00	5368,00	5368,00	5368,00
ORM_P4	5127,00	5127,00	5126,20	5127,00	5127,00	5127,00
ORM_P5	3663,00	3663,00	3663,00	3663,00	3663,00	3663,00
ORM_P6	8185,00	8185,00	8185,00	8185,00	8185,00	8185,00
ORM_P7	6217,00	6217,00	6217,00	6217,00	6217,00	6217,00
ORM_P8	5869,20	5866,60	5866,00	5865,00	5865,00	5865,00
ORM_P9	4699,00	4699,00	4699,40	4699,00	4699,40	4699,20
ORM_P10	3457,00	3457,00	3457,00	3457,00	3457,00	3457,00
ORM_P11	7855,00	7855,00	7855,00	7855,00	7855,00	7855,00
ORM_P12	7074,00	7074,00	7074,00	7074,00	7074,00	7074,00
ORM_P13	5539,00	5539,00	5539,00	5539,00	5539,00	5539,00
ORM_P14	4981,40	4981,00	4981,40	4981,00	4981,00	4981,00
ORM_P15	4477,60	4477,00	4477,00	4477,00	4477,00	4477,00
ORM_P16	8292,00	8292,00	8292,00	8292,00	8292,00	8292,00
ORM_P17	7279,00	7279,00	7279,00	7279,00	7279,00	7279,00
ORM_P18	5989,00	5987,60	5989,00	5987,20	5989,00	5990,00
ORM_P19	4787,20	4787,00	4788,00	4787,20	4787,20	4787,80
ORM_P20	4860,80	4860,60	4860,00	4860,00	4860,00	4860,00
<i>melhor</i>	13	16	15	18	16	16

Tabela 5.3: Configuração de k_{max} : qualidade média das instâncias ORM_P1 a ORM_P20 ($w = 2$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
ORM_P1	7184,00	7184,00	7184,00	7184,00	7184,00	7184,00
ORM_P2	6572,00	6572,00	6572,00	6572,00	6572,00	6572,00
ORM_P3	6776,00	6776,00	6776,00	6776,00	6776,00	6776,00
ORM_P4	6944,00	6944,00	6944,00	6944,00	6944,00	6944,00
ORM_P5	5292,00	5292,00	5292,00	5292,00	5292,00	5292,00
ORM_P6	8662,00	8662,00	8662,00	8662,00	8662,00	8662,00
ORM_P7	6984,80	6982,60	6980,80	6980,80	6980,80	6980,80
ORM_P8	7209,20	7205,00	7203,00	7203,00	7203,00	7203,00
ORM_P9	6436,00	6436,00	6436,00	6436,00	6436,00	6436,00
ORM_P10	5004,40	5004,00	5004,00	5004,00	5004,00	5004,00
ORM_P11	8062,00	8062,00	8062,00	8062,00	8062,00	8062,00
ORM_P12	7666,00	7666,00	7666,00	7666,00	7666,00	7666,00
ORM_P13	6693,00	6693,00	6693,00	6693,00	6693,00	6693,00
ORM_P14	6633,60	6633,60	6632,40	6632,00	6632,00	6632,00
ORM_P15	6369,60	6365,20	6364,60	6364,00	6363,80	6363,60
ORM_P16	8458,00	8458,00	8458,00	8458,00	8458,00	8458,00
ORM_P17	7676,00	7676,00	7676,00	7676,00	7676,00	7676,00
ORM_P18	7259,00	7259,00	7259,00	7259,00	7259,00	7259,00
ORM_P19	6538,80	6538,20	6538,60	6537,20	6537,00	6537,80
ORM_P20	7282,00	7275,40	7277,60	7269,00	7273,80	7271,80
<i>melhor</i>	13	14	16	18	18	18

Tabela 5.4: Configuração de k_{max} : qualidade média das instâncias ORM_P1 a ORM_P20 ($w = 5$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
ORM_P1	8146,00	8146,00	8146,00	8146,00	8146,00	8146,00
ORM_P2	7706,00	7706,00	7706,00	7706,00	7706,00	7706,00
ORM_P3	8265,00	8265,00	8265,00	8265,00	8265,00	8265,00
ORM_P4	9135,00	9135,00	9135,00	9135,00	9135,00	9135,00
ORM_P5	7498,60	7485,00	7485,00	7485,00	7485,00	7485,00
ORM_P6	9387,00	9387,00	9387,00	9387,00	9387,00	9387,00
ORM_P7	7879,60	7879,00	7878,00	7878,00	7878,00	7878,00
ORM_P8	8632,00	8632,00	8632,00	8632,00	8632,00	8632,00
ORM_P9	8174,00	8174,00	8175,40	8174,00	8174,00	8174,00
ORM_P10	6900,80	6905,20	6898,80	6895,20	6896,20	6895,40
ORM_P11	8383,00	8383,00	8383,00	8383,00	8383,00	8383,00
ORM_P12	8426,00	8426,00	8426,00	8426,00	8426,00	8426,00
ORM_P13	7679,00	7679,00	7679,00	7679,00	7679,00	7679,00
ORM_P14	8507,80	8506,40	8505,60	8506,60	8505,20	8505,40
ORM_P15	8922,60	8901,80	8885,40	8906,00	8876,80	8870,60
ORM_P16	8728,00	8728,00	8728,00	8728,00	8728,00	8728,00
ORM_P17	8317,00	8317,00	8317,00	8317,00	8317,00	8317,00
ORM_P18	8547,00	8547,00	8547,00	8547,00	8547,00	8547,00
ORM_P19	8328,00	8329,80	8319,00	8316,60	8322,60	8325,00
ORM_P20	10504,40	10499,80	10497,60	10477,60	10457,60	10449,40
<i>melhor</i>	13	14	14	17	16	17

Tabela 5.5: Configuração de k_{max} : qualidade média das instâncias ORM_P1 a ORM_P20 ($w = 10$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
GRM_P1	6403,00	6403,00	6403,00	6403,00	6403,00	6403,00
GRM_P2	5489,80	5489,00	5489,00	5489,00	5489,00	5489,00
GRM_P3	5157,00	5157,00	5157,00	5157,00	5157,00	5157,00
GRM_P4	5246,00	5246,00	5246,00	5246,00	5246,00	5246,00
GRM_P5	5403,00	5403,00	5403,00	5403,00	5403,00	5403,00
GRM_P6	5580,00	5580,00	5580,00	5580,00	5580,00	5580,00
GRM_P7	5766,00	5766,00	5766,00	5766,00	5766,00	5766,00
GRM_P8	5961,00	5961,00	5961,00	5961,00	5961,00	5961,00
GRM_P9	6364,00	6364,00	6364,00	6364,00	6364,00	6364,00
GRM_P10	11475,00	11475,00	11475,00	11475,00	11475,00	11475,00
GRM_P11	9862,00	9862,00	9862,00	9862,00	9862,00	9862,00
GRM_P12	8910,00	8910,00	8910,00	8910,00	8910,00	8910,00
GRM_P13	8352,00	8352,00	8352,00	8352,00	8352,00	8352,00
GRM_P14	8181,60	8186,80	8187,80	8184,00	8178,60	8179,60
GRM_P15	8137,20	8133,00	8133,00	8133,00	8133,00	8134,60
GRM_P16	8243,80	8244,40	8244,20	8244,00	8244,00	8244,20
GRM_P17	8372,00	8372,00	8372,00	8372,00	8372,00	8372,00
GRM_P18	8530,00	8530,00	8530,00	8530,00	8530,00	8530,00
GRM_P19	8701,00	8701,00	8701,00	8701,00	8701,00	8701,00
GRM_P20	9080,00	9080,00	9080,00	9080,00	9080,00	9080,00
<i>melhor</i>	17	18	18	18	19	17

Tabela 5.6: Configuração de k_{max} : qualidade média das instâncias GRM_P ($w = 2$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
GRM_P1	7036,00	7036,00	7036,00	7036,00	7036,00	7036,00
GRM_P2	6662,00	6662,00	6662,00	6662,00	6662,00	6662,00
GRM_P3	6923,00	6923,00	6923,00	6923,00	6923,00	6923,00
GRM_P4	7602,00	7602,00	7602,00	7602,00	7602,00	7602,00
GRM_P5	8379,00	8379,00	8379,00	8379,00	8379,00	8379,00
GRM_P6	9178,00	9178,00	9178,00	9178,00	9178,00	9178,00
GRM_P7	9984,00	9984,00	9984,00	9984,00	9984,00	9984,00
GRM_P8	10802,00	10802,00	10802,00	10802,00	10802,00	10802,00
GRM_P9	12447,00	12447,00	12447,00	12447,00	12447,00	12447,00
GRM_P10	12202,00	12202,00	12202,00	12202,00	12202,00	12202,00
GRM_P11	11086,00	11086,00	11086,00	11086,00	11086,00	11086,00
GRM_P12	10776,60	10769,40	10767,00	10767,00	10767,00	10767,00
GRM_P13	10904,40	10902,00	10902,80	10902,00	10902,80	10902,80
GRM_P14	11293,00	11293,00	11294,60	11293,80	11293,00	11293,80
GRM_P15	11876,60	11874,40	11876,40	11876,20	11874,40	11874,60
GRM_P16	12542,60	12539,00	12541,40	12541,40	12539,00	12539,00
GRM_P17	13278,80	13277,40	13277,40	13276,00	13277,40	13277,40
GRM_P18	14036,00	14036,00	14036,00	14036,00	14036,00	14036,00
GRM_P19	14814,00	14814,00	14814,00	14814,00	14814,00	14814
GRM_P20	16422,00	16422,00	16422,00	16422,00	16422,00	16422,00
<i>melhor</i>	15	18	15	17	18	16

Tabela 5.7: Configuração de k_{max} : qualidade média das instâncias GRM_P ($w = 5$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
GRM_P1	7884,00	7884,00	7884,00	7884,00	7884,00	7884,00
GRM_P2	8580,00	8580,00	8580,00	8580,00	8580,00	8580,00
GRM_P3	9793,00	9793,00	9793,00	9793,00	9793,00	9793,00
GRM_P4	11487,00	11487,00	11487,00	11487,00	11487,00	11487,00
GRM_P5	13273,00	13273,00	13273,00	13273,00	13273,00	13273,00
GRM_P6	15079,00	15079,00	15079,00	15079,00	15079,00	15079,00
GRM_P7	16909,00	16909,00	16909,00	16909,00	16909,00	16909,00
GRM_P8	18754,00	18754,00	18754,00	18754,00	18754,00	18754,00
GRM_P9	22510,40	22507,00	22507,00	22507,00	22507,00	22507,00
GRM_P10	13083,00	13083,00	13083,00	13083,00	13083,00	13083,00
GRM_P11	13040,00	13040,00	13040,00	13040,00	13040,00	13040,00
GRM_P12	13845,00	13845,00	13845,00	13845,00	13845,00	13845,00
GRM_P13	14989,00	14985,80	14986,60	14985,80	14985,80	14985,80
GRM_P14	16355,00	16355,00	16355,00	16355,00	16355,00	16355,00
GRM_P15	17919,00	17919,00	17919,00	17919,00	17919,00	17919,00
GRM_P16	19579,40	19579,60	19579,40	19579,60	19579,60	19579,60
GRM_P17	21318,80	21317,60	21318,20	21317,00	21317,00	21317,60
GRM_P18	23095,00	23095,00	23095,00	23095,00	23095,00	23095,00
GORM_P19	24895,00	24895,00	24895,00	24895,00	24895,00	24895,00
GRM_P20	28566,00	28566,00	28566,00	28566,00	28566,00	28566,00
<i>melhor</i>	17	18	18	19	19	18

Tabela 5.8: Configuração de k_{max} : qualidade média das instâncias GRM_P ($w = 10$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
ORM_P1	0,42	0,57	0,56	0,57	0,57	0,58
ORM_P2	0,81	1,17	1,13	1,13	1,13	1,16
ORM_P3	0,88	1,25	1,26	1,25	1,26	1,27
ORM_P4	3,44	4,62	4,62	4,96	5,11	5,25
ORM_P5	8,99	13,49	14,14	16,04	16,89	19,00
ORM_P6	1,16	1,51	1,51	1,53	1,50	1,56
ORM_P7	2,29	3,28	3,33	3,35	3,33	3,48
ORM_P8	9,11	13,17	12,95	13,69	13,43	14,06
ORM_P9	38,80	52,13	52,49	54,71	57,93	60,96
ORM_P10	109,20	130,58	137,51	143,25	151,24	168,19
ORM_P11	1,48	1,68	1,64	1,65	1,66	1,68
ORM_P12	3,63	4,10	4,19	4,19	4,20	4,29
ORM_P13	45,07	50,21	50,22	52,45	53,75	58,38
ORM_P14	178,57	190,15	197,15	202,34	232,35	246,72
ORM_P15	495,19	545,82	556,65	601,72	631,07	830,17
ORM_P16	2,77	3,02	3,00	3,01	2,95	3,12
ORM_P17	4,42	5,40	5,84	5,84	5,82	6,06
ORM_P18	54,82	63,12	67,98	70,59	70,74	86,74
ORM_P19	379,33	455,96	423,01	500,28	496,54	536,38
ORM_P20	1.446,04	1.397,37	1.456,86	1.579,04	1.641,58	1.891,17
<i>melhor</i>	19	1	0	0	0	0

Tabela 5.9: Configuração de k_{max} : tempo médio de execução das instâncias ORM_P1 a ORM_P20 ($w = 2$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
ORM_P1	0,77	0,80	0,80	0,79	0,81	0,80
ORM_P2	2,32	2,76	2,92	2,83	2,84	2,85
ORM_P3	3,33	3,28	3,36	3,31	3,32	3,35
ORM_P4	5,01	5,28	5,37	5,56	5,65	5,64
ORM_P5	9,06	9,62	9,36	9,57	10,05	9,92
ORM_P6	1,75	1,81	1,80	1,80	1,81	1,80
ORM_P7	7,49	7,92	7,75	7,75	7,78	7,76
ORM_P8	25,03	25,42	26,35	27,14	28,21	28,19
ORM_P9	86,81	104,32	104,96	111,63	112,38	123,23
ORM_P10	99,21	103,56	104,32	107,52	110,89	124,87
ORM_P11	1,77	1,82	1,83	1,82	1,82	1,83
ORM_P12	6,46	6,94	7,17	7,14	7,24	7,30
ORM_P13	107,73	114,27	120,67	121,12	125,43	132,41
ORM_P14	226,32	249,90	262,28	278,83	277,06	337,02
ORM_P15	510,44	520,47	499,81	523,92	525,86	587,38
ORM_P16	2,65	2,86	2,87	2,87	2,87	2,87
ORM_P17	5,60	6,43	6,75	6,78	6,79	6,82
ORM_P18	260,52	272,41	275,17	296,80	325,43	350,96
ORM_P19	1.003,81	1.042,71	1.097,13	1.121,39	1.158,49	1.180,73
ORM_P20	2.045,04	1.975,54	2.076,64	2.045,24	2.082,70	2.144,66
<i>melhor</i>	17	2	1	0	0	0

Tabela 5.10: Configuração de k_{max} : tempo médio de execução das instâncias ORM_P1 a ORM_P20 ($w = 5$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
ORM_P1	1,24	1,49	1,48	1,46	1,86	1,54
ORM_P2	2,01	2,58	2,58	2,53	2,88	2,69
ORM_P3	1,89	2,48	2,52	2,50	2,97	2,66
ORM_P4	4,78	6,16	6,25	6,52	7,62	7,45
ORM_P5	9,32	12,07	12,26	12,32	12,67	12,92
ORM_P6	3,80	4,83	4,79	4,73	4,96	5,02
ORM_P7	10,56	13,57	13,72	13,55	14,99	14,28
ORM_P8	27,33	32,74	33,16	33,24	36,20	34,96
ORM_P9	64,09	72,76	72,94	74,91	87,64	82,24
ORM_P10	125,93	131,51	135,93	130,52	149,11	151,44
ORM_P11	3,62	4,02	4,02	3,93	4,05	4,30
ORM_P12	15,68	18,27	18,36	18,03	17,10	20,00
ORM_P13	116,16	116,84	122,48	121,31	116,68	135,75
ORM_P14	362,66	409,96	403,55	450,89	443,08	566,81
ORM_P15	654,51	644,83	646,33	638,13	658,91	766,57
ORM_P16	3,68	3,98	3,98	3,73	4,01	4,25
ORM_P17	16,97	18,05	19,08	17,83	18,86	20,10
ORM_P18	317,97	326,43	337,52	366,36	366,75	426,73
ORM_P19	1.048,48	1.091,12	1.238,73	1.141,21	1.334,11	1.340,77
ORM_P20	1.886,63	2.055,35	2.211,74	2.160,39	2.470,19	2.323,34
<i>melhor</i>	19	0	0	1	0	0

Tabela 5.11: Configuração de k_{max} : tempo médio de execução das instâncias ORM_P1 a ORM_P20 ($w = 10$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
GRM_P1	1,17	1,18	1,19	1,45	1,46	1,19
GRM_P2	2,56	3,02	3,16	3,90	3,96	3,19
GRM_P3	8,17	8,94	9,19	12,12	12,25	9,81
GRM_P4	21,92	31,32	31,84	39,46	40,41	35,02
GRM_P5	50,01	63,91	65,79	71,42	74,41	74,04
GRM_P6	57,63	70,46	70,69	79,16	84,94	91,86
GRM_P7	83,34	99,43	103,22	109,18	114,23	113,48
GRM_P8	79,74	82,28	88,11	95,80	100,14	97,98
GRM_P9	62,34	64,34	64,99	70,14	69,71	71,25
GRM_P10	3,42	3,63	3,67	3,90	3,86	3,75
GRM_P11	10,14	10,90	11,79	12,70	12,48	12,16
GRM_P12	13,73	17,03	16,57	19,41	18,80	18,34
GRM_P13	23,53	26,38	28,42	31,81	34,05	32,73
GRM_P14	51,19	60,66	60,12	68,82	68,22	67,71
GRM_P15	94,98	112,04	118,80	130,28	133,16	134,73
GRM_P16	187,04	230,55	218,84	253,00	273,75	283,70
GRM_P17	276,79	300,78	303,94	327,17	337,85	371,94
GRM_P18	308,50	323,77	322,37	353,72	358,43	417,53
GRM_P19	311,23	379,34	400,37	433,24	446,63	434,05
GRM_P20	365,20	430,35	448,85	468,14	467,97	473,94
<i>melhor</i>	20	0	0	0	0	0

Tabela 5.12: Configuração de k_{max} : tempo médio de execução das instâncias GRM_P ($w = 2$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
GRM_P1	2,69	2,64	2,63	2,63	2,64	2,69
GRM_P2	7,90	8,82	9,06	9,08	9,07	8,95
GRM_P3	17,66	20,81	22,02	24,24	24,25	23,97
GRM_P4	27,14	31,46	30,61	34,49	34,24	34,64
GRM_P5	27,73	29,17	29,56	30,42	30,64	31,18
GRM_P6	32,20	32,19	31,93	32,69	33,40	34,11
GRM_P7	34,38	34,59	35,02	34,90	35,32	36,30
GRM_P8	38,78	39,57	39,31	39,03	39,88	39,76
GRM_P9	47,05	47,77	48,13	49,16	48,23	49,93
GRM_P10	5,56	5,55	5,56	5,56	5,53	5,62
GRM_P11	18,58	18,96	19,10	19,07	19,02	19,40
GRM_P12	39,58	46,74	50,53	53,01	53,10	53,78
GRM_P13	90,32	86,96	97,13	99,23	96,69	97,86
GRM_P14	109,55	133,48	137,92	142,88	141,86	147,30
GRM_P15	167,63	183,96	189,24	201,31	205,75	208,59
GRM_P16	187,71	214,19	231,60	234,91	230,44	257,93
GRM_P17	209,09	222,07	220,40	236,09	250,54	265,92
GRM_P18	207,29	237,79	237,73	246,59	279,43	288,64
GRM_P19	412,28	464,12	485,28	504,09	541,83	518,99
GRM_P20	345,17	380,27	374,55	412,22	454,20	446,87
<i>melhor</i>	16	1	2	1	1	0

Tabela 5.13: Configuração de k_{max} : tempo médio de execução das instâncias GRM_P ($w = 5$).

	$k_{max} = 4$	$k_{max} = 8$	$k_{max} = 10$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = p$
Instância	Média	Média	Média	Média	Média	Média
GRM_P1	3,64	3,60	4,67	3,60	4,26	3,65
GRM_P2	9,09	10,52	14,20	10,83	12,98	11,20
GRM_P3	13,51	16,64	24,01	20,22	23,77	23,28
GRM_P4	19,96	21,05	27,04	24,04	27,78	28,07
GRM_P5	28,07	28,62	32,03	30,50	30,37	31,08
GRM_P6	33,59	36,01	37,76	37,63	35,97	37,00
GRM_P7	39,15	39,58	43,24	44,08	41,77	43,45
GRM_P8	45,71	45,79	48,49	49,88	47,24	49,03
GRM_P9	55,58	54,38	60,07	58,01	56,91	59,40
GRM_P10	9,29	9,67	10,07	10,08	9,67	10,10
GRM_P11	36,53	39,33	40,34	40,75	39,16	42,13
GRM_P12	67,63	66,83	70,67	77,67	73,63	80,89
GRM_P13	101,76	109,55	108,34	113,30	117,56	128,80
GRM_P14	127,43	147,91	163,08	165,08	162,85	179,80
GRM_P15	161,96	176,36	192,55	186,63	172,74	202,50
GRM_P16	182,97	198,85	208,61	217,94	217,68	271,43
GRM_P17	221,97	242,82	249,37	278,29	263,31	291,36
GRM_P18	293,68	327,98	323,76	342,50	346,65	365,00
GRM_P19	294,98	296,93	289,13	317,18	290,65	335,16
GRM_P20	337,02	332,47	342,99	349,87	343,46	355,16
<i>melhor</i>	15	4	1	1	0	0

Tabela 5.14: Configuração de k_{max} : tempo médio de execução das instâncias GRM_P ($w = 10$).

5.5

Configuração do Parâmetro b

Na linha 6 do algoritmo VNS em sua versão básica (Figura 5.1), uma perturbação aleatória é realizada na vizinhança de ordem k da solução corrente. Uma possível modificação seria encontrar a melhor solução perturbada entre b soluções geradas aleatoriamente na k -ésima vizinhança da solução corrente, aplicando, em seguida, a busca local a essa solução.

Para mostrar a influência da quantidade de vezes que a perturbação é executada (parâmetro b) na qualidade das soluções e nos tempos de processamento, testes computacionais foram realizados comparando-se o VNS com filtro com $b = 1, 2, 4, 8$ e 10 . O valor $b = 1$ corresponde à configuração padrão utilizada no algoritmo da Figura 5.1.

Para a análise da qualidade das soluções, as medidas relativas $drpm$, cm e $melhor$ foram utilizadas, como mostra a Tabela 5.15.

	b				
	1	2	4	8	10
$drpm$	0,000234	-0,000748	-0,002189	-0,000971	0,003674
cm	2,96	2,94	2,92	2,93	3,25
$melhor$	163	169	167	166	160

Tabela 5.15: Qualidade relativa na configuração de b .

Nas medidas $drpm$ e cm , os melhores resultados foram obtidos por $b = 4$, enquanto na medida $melhor$, a configuração $b = 2$ obteve o melhor resultado. As configurações $b = 1$ e $b = 10$ obtiveram os piores resultados em todas as medidas relativas.

A Tabela 5.16 apresenta as medidas $drpm$, cm e trm utilizadas na análise dos tempos de processamento obtidos pelo VNS com filtro e $b = 1, 2, 4, 8$ e 10 . Nessa última medida, a configuração padrão $b = 1$ foi tomada como algoritmo de referência.

	b				
	1	2	4	8	10
$drpm$	-1,98	-0,83	-0,13	-1,68	4,63
cm	2,78	2,83	3,00	2,86	3,53
trm	1,00	1,02	1,02	1,01	1,07

Tabela 5.16: Tempo relativo na configuração de b .

Em todas as medidas, os melhores resultados foram obtidos por $b = 1$. Porém, os resultados foram muito semelhantes em todas as configurações, mostrando que a operação de perturbação é relativamente barata em comparação ao tempo de processamento total do algoritmo. A maior perturbação testada ($b = 10$) torna, em média, o algoritmo 7% mais lento do que a configuração $b = 1$ para as instâncias analisadas.

Os resultados das tabelas mostram que, perturbar a solução mais de uma vez na k -ésima vizinhança da solução corrente e executar a busca local na melhor solução encontrada pode levar a resultados promissores. Os resultados obtidos pelas configurações $b = 2$, $b = 4$ e $b = 8$ são muito semelhantes em termos de qualidade e tempo. Optou-se por utilizar $b = 2$.

5.6 VNS e Reconexão por Caminhos

O procedimento de reconexão por caminhos foi também utilizado no VNS como estratégia de intensificação conectando cada ótimo local produzido pela busca local com um elemento de um conjunto de soluções de elite armazenados durante as iterações do VNS (ver Seção 4.5).

5.6.1 Configuração do Tamanho do Conjunto de Soluções de Elite

O parâmetro a ser ajustado no procedimento de reconexão por caminhos é o tamanho máximo do conjunto de soluções de elite (*MaxElite*). As seguintes configurações foram analisadas: VNS com filtro sem reconexão por caminhos (VNSf) e VNS com filtro e reconexão por caminhos com *MaxElite* = 5, 10, 15 e 20 (VNSf_RC5, VNSf_RC10, VNSf_RC15 e VNSf_RC20, respectivamente).

As medidas *drpm* e *cm* serão utilizadas para analisar a qualidade das soluções encontradas, como mostra a Tabela 5.17.

	<i>drpm</i>	<i>cm</i>
VNSf	0,001637	3,25
VNSf_RC5	-0,001101	2,82
VNSf_RC10	-0,001499	2,97
VNSf_RC15	-0,0002	2,94
VNSf_RC20	0,001165	3,01

Tabela 5.17: Qualidade relativa na configuração de *MaxElite* no VNS.

Analisando-se a Tabela 5.17, nas medidas *drpm* e *cm*, o melhor resultado foi obtido pelo VNSf_RC10 e VNSf_RC5, respectivamente. Em termos absolutos, dos 180 testes realizados (36 instâncias com cinco execuções cada), VNSf_RC5, VNSf_RC10, VNSf_RC15 e VNSf_RC20 melhoraram a qualidade da solução encontrada pelo VNSf em 13, 14, 12 e 12 testes, respectivamente.

Para a análise dos tempos de processamento, os valores obtidos pelos algoritmos nas medidas *drpm*, *cm* e *trm* são mostrados na Tabela 5.18. Na medida *trm*, o VNSf é o algoritmo de referência.

Analisando-se as medidas na Tabela 5.18, os melhores resultados foram obtidos pelo VNSf, pois não existe a necessidade do controle e gerenciamento do conjunto de soluções de elite. Os algoritmos com reconexão por caminhos não apresentaram uma diferença significativa entre eles.

	<i>drpm</i>	<i>cm</i>	<i>trm</i>
VNSf	-8,39	1,81	1,00
VNSf_RC5	1,95	3,00	1,13
VNSf_RC10	1,70	3,03	1,13
VNSf_RC15	1,84	3,25	1,13
VNSf_RC20	2,91	3,75	1,14

Tabela 5.18: Tempo relativo na configuração de *MaxElite* no VNS.

Levando-se em consideração as medidas de qualidade e tempo, destacaram-se os algoritmos VNS_RC5 e VNS_RC10, pois apresentaram menores tempos de processamento e qualidade superiores às demais configurações com um maior número de soluções de elite. Optou-se pela configuração *MaxElite* = 10 para armazenar um número maior de soluções.

5.6.2

Comparação entre o VNS com Filtro e o VNS com Filtro e Reconexão por Caminhos

Para mostrar a efetividade do procedimento de reconexão por caminhos no VNS, comparações serão realizadas através de gráficos de distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo, executando-se o algoritmo com e sem o procedimento. As seguintes instâncias com seus respectivos valores alvo entre parênteses serão analisadas: GRM_P11 (11086), GRM_P18 (14036), ORM_P9 (6436) e ORM_P18 (7259), para $w = 5$. Nas figuras, o VNS com filtro com parâmetros $k_{max} = 15$ e $b = 2$ será denominado de VNSf e o VNS com filtro e reconexão por caminhos com os parâmetros $k_{max} = 15$, $b = 2$ e *MaxElite* = 10 será denominado de VNSf_RC.

As Figuras 5.2 e 5.3 mostram que as curvas do VNSf e do VNSf_RC praticamente se equivalem para essas instâncias. No primeiro gráfico, o VNSf chega a ser um pouco melhor do que o VNSf_RC para encontrar o valor alvo 11086 na instância GRM_P11. A probabilidade de encontrar um valor menor ou igual ao alvo em menos de 1,1 segundo é de aproximadamente 89% (resp. 81%) para o VNSf (resp. VNSf_RC). Em instâncias menores, os ganhos com a utilização do procedimento de reconexão por caminhos no VNS não são significativos.

Quando aumenta-se o tamanho da instância, a utilização do procedimento de reconexão por caminhos torna-se uma boa estratégia, como mostram as Figuras 5.4 e 5.5.

Os gráficos ilustram que a probabilidade de encontrar um valor pelo menos tão bom quanto o valor alvo aumenta do VNSf para o VNSf_RC para as duas instâncias testadas. Na Figura 5.4, a probabilidade de encontrar um valor alvo menor ou igual a 7259 em menos do que 40 segundos é de aproximadamente 82% para o VNSf e de aproximadamente 91% para o

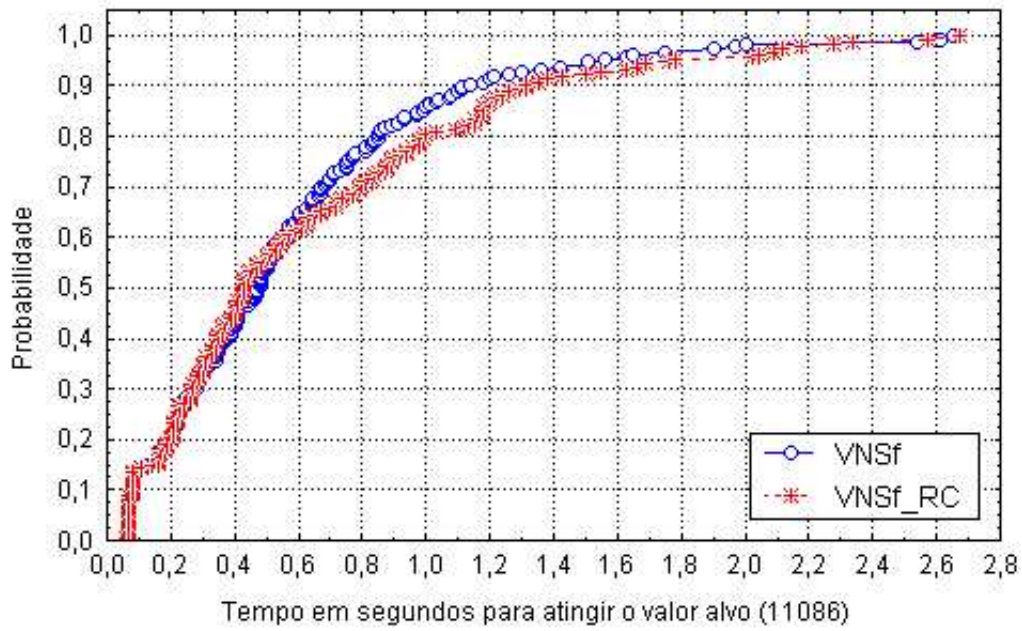


Figura 5.2: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 11086 para a instância GRM_P11 ($w = 5$).

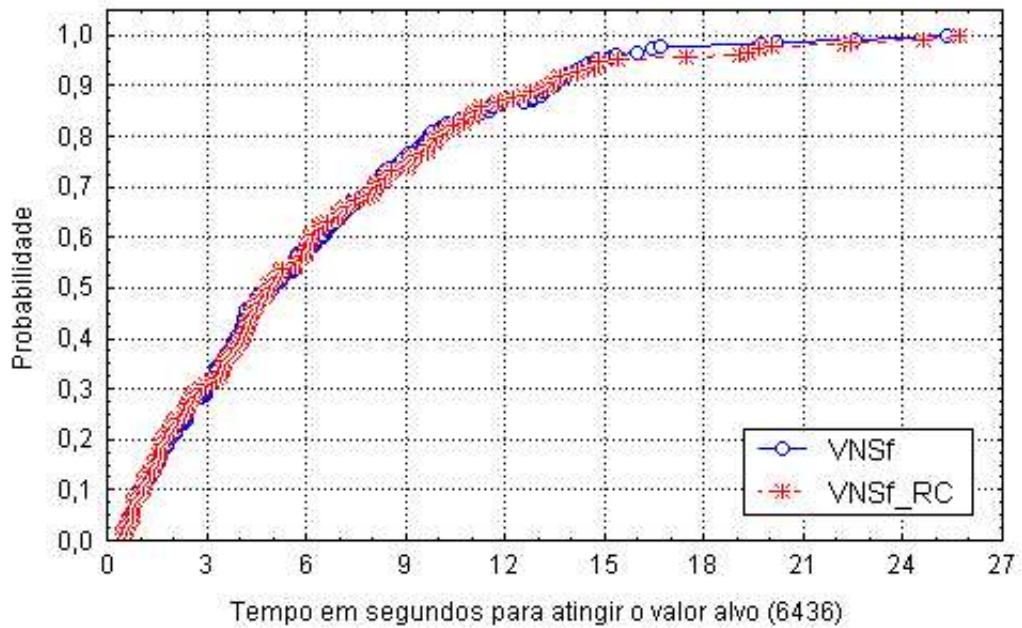


Figura 5.3: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 6436 para a instância ORM_P9 ($w = 5$).

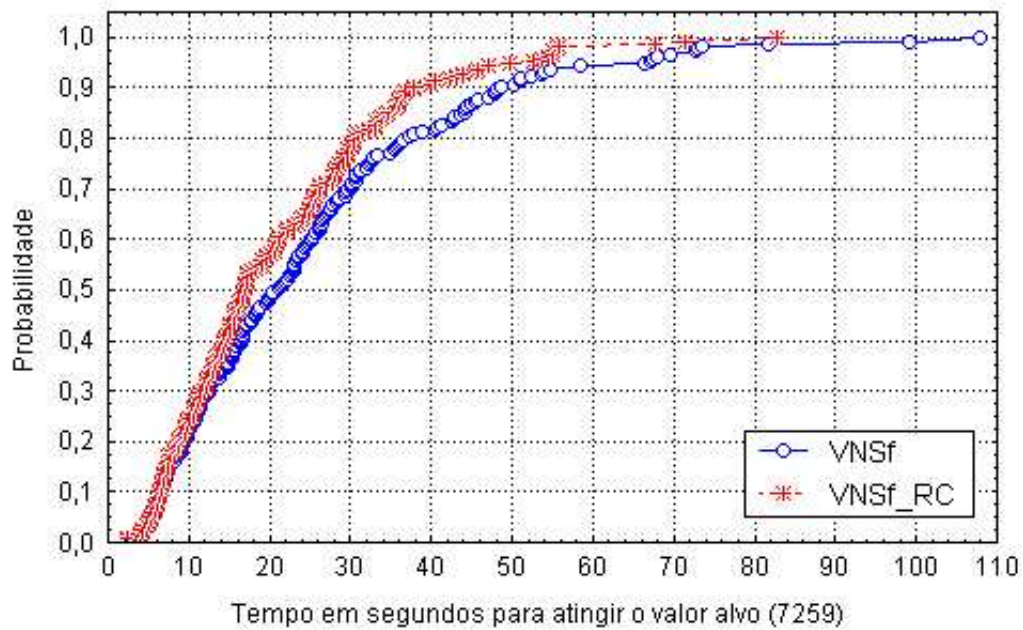


Figura 5.4: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 7259 para a instância ORM_P18 ($w = 5$).

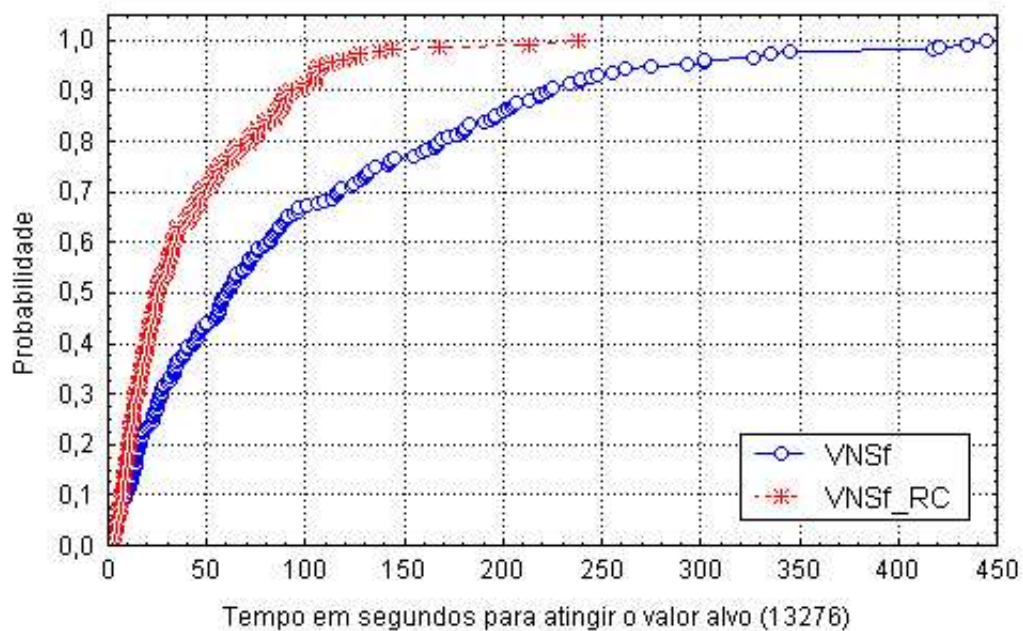


Figura 5.5: Distribuição de probabilidade empírica do tempo gasto para encontrar o valor alvo 13276 para a instância GRM_P17 ($w = 5$).

VNSf_RC. Na Figura 5.5, o ganho do VNSf_RC em relação ao VNSf é mais significativo. A probabilidade de encontrar um valor alvo menor ou igual a 13276 em menos do que 150 segundos é de aproximadamente 77% para o VNSf e de quase 100% para o VNSf_RC.

5.7

Algoritmo VNS com Filtro e Reconexão por Caminhos

A heurística VNS com filtro e reconexão por caminhos é mostrada na Figura 5.6.

O conjunto de soluções de elite P com tamanho máximo igual a dez está inicialmente vazio (linha 1). A solução inicial S do VNS é gerada na linha 3. Executa-se o laço das linhas 4 a 43 enquanto o critério de parada não for satisfeito. O critério de parada escolhido para a configuração do VNS foi o número de iterações realizadas ($niter = 100$). O algoritmo inicia explorando a vizinhança que possui a menor distância em relação à solução corrente (linha 5). Enquanto a maior estrutura de vizinhança não for alcançada ($k_{max} = 15$) no laço das linhas 6 a 42, os seguintes passos são executados. Primeiramente, a solução S é perturbada aleatoriamente $b = 2$ vezes na sua k -ésima vizinhança (linhas 7 a 13) e a melhor solução perturbada serve como entrada para a busca local mais rápida na linha 14. Na linha 15 ocorre o teste da estratégia de filtro e caso ambas as condições sejam satisfeitas, executa-se a busca local mais lenta no ótimo local encontrado pela primeira busca (linha 16). O ótimo local encontrado pela segunda busca é um candidato a inserção no conjunto de soluções de elite e caso satisfaça as condições de qualidade e diversidade, torna-se um novo elemento de P (linha 22). Caso pelo menos uma das condições não sejam satisfeitas na linha 15, o ótimo local da primeira busca também pode ser inserido em P (linha 22). Na linha 23, o procedimento de reconexão por caminhos começa a atuar a partir da segunda iteração. Primeiramente, uma solução é escolhida do conjunto de soluções de elite privilegiando-se elementos com uma maior diferença simétrica em relação ao ótimo local candidato (linha 24). Na linha 25, determina-se qual será a solução inicial e a solução alvo entre o ótimo local candidato e o elemento de P . A melhor solução será a solução inicial e a pior será a solução alvo. Assim, de acordo com esse teste, executa-se reconexão por caminhos entre a solução inicial e a solução alvo nas linhas 26 ou 29. Caso o procedimento obtenha sucesso, retorna-se a melhor solução encontrada no caminho; caso contrário, retorna-se a melhor solução entre S_1 e S_2 . Durante cada passo, se o procedimento encontra uma solução melhor do que a melhor solução encontrada até o momento, esta também torna-se uma candidata a inserção em P . Se o valor da função objetivo do procedimento de reconexão por caminhos for melhor do que o valor da função objetivo da melhor solução encontrada até o momento (linha 31), atualiza-se a melhor

```

Procedimento VNS_Filtro_Reconexão_Caminhos( $\alpha$ , semente)
1.  $P \leftarrow \emptyset$ ;
2.  $F^* \leftarrow +\infty$ ;
3.  $S \leftarrow \text{Fase\_Construção}(\alpha, \textit{semente})$ ;
4. Para  $it = 1, \dots, \textit{niter}$  Faça
5.      $k \leftarrow 1$ ;
6.     Enquanto ( $k \leq k_{max}$ ) Faça
7.          $F' \leftarrow +\infty$ ;
8.         Para  $j = 1, \dots, b$  Faça
9.             Gerar uma solução aleatória  $S'' \in N_k(S)$ ;
10.            Se ( $F(S'') < F'$ ) Então
11.                 $F' \leftarrow F(S'')$ ,  $S' \leftarrow S''$ ;
12.            Fim-Se
13.        Fim-Para
14.         $S'_1 \leftarrow \text{BL\_BTMA}(S')$ ;
15.        Se ( $S'_1$  não foi visitado previamente) e
            ( $F(S'_1) < (1 + \lambda) \times F^*$ ) Então
16.             $S' \leftarrow \text{BL\_Teste}(S'_1)$ ;
17.        Fim-Se
18.        Senão
19.             $S' \leftarrow S'_1$ ;
20.        Fim-Senão
21.         $S_1 \leftarrow S'$ ;
22.        Atualizar o conjunto de soluções de elite  $P$  com  $S_1$ ;
23.        Se ( $it \geq 2$ ) Então
24.            Escolher uma solução  $S_2 \in P$ ;
25.            Se ( $F(S_1) \leq F(S_2)$ ) Então
26.                 $S'' \leftarrow \text{Reconexão\_por\_Caminhos}(S_1, S_2)$ ;
27.            Fim-Se
28.            Senão
29.                 $S'' \leftarrow \text{Reconexão\_por\_Caminhos}(S_2, S_1)$ ;
30.            Fim-Senão
31.            Se ( $F(S'') < F^*$ ) Então
32.                 $F^* \leftarrow F(S'')$ ,  $S^* \leftarrow S''$ ;
33.             $S \leftarrow S''$ ;
34.             $k \leftarrow 1$ ;
35.        Fim-Se
36.        Senão
37.             $k \leftarrow k + 1$ ;
38.        Fim-Senão
39.        Senão
40.             $F^* \leftarrow F(S')$ ,  $S^* \leftarrow S'$ ;
41.        Fim-Senão
42.    Fim-Enquanto
43. Fim-Para
44. Retorne  $S^*$ ;
Fim

```

Figura 5.6: Algoritmo VNS com filtro e reconexão por caminhos.

solução, a solução corrente e retorna-se à vizinhança N_1 (linhas 32 a 34). Caso contrário, aumenta-se a distância entre a solução corrente S e as soluções geradas aleatoriamente no processo de perturbação (linha 37). Na primeira iteração, atualiza-se a melhor solução nas linhas 39 a 41. A melhor solução encontrada pelo algoritmo é retornada na linha 44.

5.8

Considerações Finais

Este capítulo propôs a heurística VNS com filtro e reconexão por caminhos para o problema das p -medianas conectadas.

A solução inicial do VNS é gerada pela heurística construtiva do GRASP com filtro e reconexão por caminhos, apresentada na Seção 4.1, e o algoritmo de busca local utilizado foi descrito no Capítulo 3.

Com o objetivo de diminuir os tempos de processamento apresentados pelo VNS em sua versão básica, uma estratégia de filtro foi incorporada ao algoritmo entre a busca local pelas bordas com teste da menor aresta e a busca local com teste. Resultados computacionais comparando-se o VNS com e sem o filtro mostraram ganhos em termos de tempo de processamento com uma pequena perda na qualidade das soluções encontradas.

Uma simples modificação que se mostrou efetiva no algoritmo foi perturbar aleatoriamente a solução corrente um número determinado de vezes (parâmetro b) e utilizar a melhor solução perturbada como entrada para o algoritmo de busca local. Testes computacionais foram realizados com valores de $b = 1, 2, 4, 8$ e 10 . Como essa é uma operação relativamente barata, os tempos de processamento mantiveram-se praticamente idênticos à configuração padrão ($b = 1$) com melhoras na qualidade das soluções encontradas em quase todas as configurações. De acordo com os testes realizados, optou-se por utilizar $b = 2$.

O procedimento de reconexão por caminhos foi utilizado do mesmo modo ao apresentado no Capítulo 4. Comparações realizadas entre o VNS com filtro com e sem o procedimento através de gráficos de distribuição de probabilidade empírica do tempo gasto para atingir o valor alvo mostraram que, em instâncias menores, o VNS não se beneficia do procedimento. Porém, quando aumenta-se o tamanho da instância, a utilização do procedimento é uma boa estratégia, conduzindo a resultados melhores do que o VNS que não o utiliza.