

Metaheurística GRASP com refinamento por busca local para o Flowshop Permutacional

Alberto F. K. Neto¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

afkneto@inf.ufrgs.br

1. Introdução

Este relatório refere-se ao trabalho de otimização da disciplina de Otimização Combinatória (INF05010), cursada no período de 2019/1. O texto apresenta o problema de otimização considerado e introduz um modelo de Programação Linear Inteira da literatura do problema. Detalhes sobre o desenvolvimento de um método de solução heurístico baseado em GRASP e Busca Local encontram-se disponíveis nas seções indicadas, e o desempenho do método proposto é comparado com os melhores valores de solução atualmente conhecidos para um pequeno conjunto de instâncias de teste.

2. Descrição do problema

O Problema de Flowshop Permutacional (PFSP) é um tema de pesquisa recorrente nos estudos da otimização combinatória. O problema considera um conjunto de M máquinas e N tarefas, em que todas as tarefas devem ser processadas exatamente uma vez em cada uma das máquinas consideradas. Cada tarefa $1 \leq j \leq N$ demora $T_{rj} \geq 0$ unidades de tempo para ser processada cada máquina $1 \leq r \leq M$. Busca-se uma ordem de execução das tarefas que minimize o tempo final de processamento da última máquina considerada. Essa ordem de execução é seguida por todas as máquinas.

[Tseng et al. 2004] propuseram um modelo de programação linear inteira mista para o problema. As variáveis binárias $D_{ik} \in \{0, 1\}$ assumem o valor 1 para indicar se a tarefa i deve ser processada em algum momento anterior ao processamento da tarefa k , com $1 \leq i < k \leq N$. Já as variáveis contínuas $C_{ri} \geq 0$ indicam o horizonte de tempo de processamento que cada tarefa $1 \leq i \leq N$ em cada máquina $1 \leq r \leq M$. Adicionalmente, a variável $C_{\max} \geq 0$ é utilizado no cálculo do tempo final de processamento da última máquina. De posse dessas definições, a seguinte formulação modela o Problema de Flowshop permutacional. Note a existência de um parâmetro P , que é um número suficientemente grande usado como “big-M” na modelagem das restrições lógicas do modelo.

$$\text{Minimize } C_{\max} \quad (1)$$

Sujeito a:

$$C_{1i} \geq T_{1i} \quad 1 \leq i \leq N \quad (2)$$

$$C_{ri} - C_{r-1,i} \geq T_{ri} \quad 2 \leq r \leq M, 1 \leq i \leq N \quad (3)$$

$$C_{ri} - C_{rk} + PD_{ik} \geq T_{ri} \quad 1 \leq r \leq M, 1 \leq i < k \leq N \quad (4)$$

$$C_{ri} - C_{rk} + PD_{ik} \leq P - T_{rk} \quad 1 \leq r \leq M, 1 \leq i < k \leq N \quad (5)$$

$$C_{\max} \geq C_{Mi} \quad 1 \leq i \leq N \quad (6)$$

$$C_{ri} \geq 0 \quad 1 \leq r \leq M, 1 \leq i \leq N \quad (7)$$

$$D_{ik} \in \{0, 1\} \quad 1 \leq i < k \leq N \quad (8)$$

A função objetivo (1) minimiza o tempo de processamento final da última máquina do problema. As restrições (2) e (3) modelam o tempo final de processamento das tarefas na primeira e demais máquinas, respectivamente. As restrições (4–5) garantem uma única ordem de execução das tarefas em todas as máquinas. A restrição (6) calcula o tempo final de processamento da última máquina. Por fim, as restrições (7–8) modelam o domínio das variáveis de decisão do problema.

3. Método de solução com GRASP e Busca Local

Tendo em vista a questão da típica baixa eficiência de métodos exatos em resolver problemas de otimização combinatória discreta, propõe-se o seguinte método de solução heurístico para resolução do problema. O método de solução é implementa uma heurística GRASP para construção de uma solução inicial [Feo et al. 1994], seguida de uma fase de intensificação com busca local. O pseudocódigo dos algoritmos de construção inicial e de busca local são listados em 1 e 2. Na notação a seguir, uma solução é definida como uma lista com a ordem de processamento das tarefas, e pode ser parcial ou completa. Uma visão geral do método de solução está disponível no algoritmo 3.

Algorithm 1: Construção de solução inicial com GRASP.

```

1 Procedure GRASP ( $N, M, T, \alpha$ )
2    $pend \leftarrow$  lista com valores  $1, 2, \dots, N$ 
3    $s \leftarrow$  lista vazia;  $z \leftarrow 0$ 
4   while  $pend$  não está vazia do
5      $RCL \leftarrow$  lista vazia
6     for  $j \in pend$  do
7        $\bar{z}_j \leftarrow$  custo da solução parcial  $s$  com adição da tarefa  $j$ 
8       adicione a tupla  $(j, \bar{z}_j)$  em  $RCL$ 
9     ordene  $RCL$  em ordem não crescente de  $\bar{z}$ 
10     $tam \leftarrow$  tamanho da lista  $RCL$ 
11     $tp \leftarrow$  escolhe aleatoriamente um índice de  $[1, \max\{1, \alpha \cdot tam\}]$ 
12    atualize a solução  $s$  e custo  $z$  com os dados da tupla  $RCL_{tp}$ 
13    remova a tarefa referente a  $tp$  de  $pend$ 
14 return  $s$ 

```

O algoritmo GRASP inicial com uma solução vazia, de custo 0, e incrementalmente adiciona tarefas na ordem de processamento das máquinas. Inicialmente, todas as tarefas são marcadas como pendentes (lista *pend*). A cada iteração do laço principal (linhas 4–13), calcula-se o custo de inserção de cada tarefa pendente na solução parcial s . Esses valores de custo são adicionados à lista *RCL* de tarefas candidatas a entrar na solução. Faz-se a ordenação dessa lista em ordem não crescente de custo de solução, e escolhe-se aleatoriamente uma das $\alpha\%$ tarefas iniciais da lista de candidatos. Essa tarefa entra na solução parcial s , e o custo z é atualizado de acordo. Finalmente, a tarefa é removida da lista de pendentes e a próxima iteração inicia. Essa implementação de GRASP faz a seleção com α pelos índices da lista de candidatos.

Algorithm 2: Algoritmo de Busca Local iterada com trocas aleatória.

```

1 Procedure Swap2LS ( $s^*$ ,  $numVezes$ )
2    $z^* \leftarrow$  custo da solução atual
3   for  $i \leftarrow 1$  até  $numVezes$  do
4     selecione tarefas  $j_1 \neq j_2$  aleatoriamente, com distribuição uniforme
5      $\bar{s} \leftarrow$  troque a ordem de processamento de  $j_1 \leftrightarrow j_2$  em  $s^*$ 
6      $\bar{z} \leftarrow$  avalie o custo da solução  $\bar{s}$ 
7     if  $\bar{z} < z^*$  then
8        $s^* \leftarrow \bar{s}$ 
9        $z^* \leftarrow \bar{z}$ 
10 return  $s^*$ 

```

Após a construção de uma solução inicial com GRASP, inicia-se a fase de melhoramento da solução com o algoritmo de busca local iterado 2. A busca local faz diversas tentativas de troca da ordem de processamento de duas tarefas em posições distintas, e sempre aceita a troca na ordem das tarefas caso seja vantajosa (estratégia de “primeira melhora”). De posse de ambos os algoritmos, é possível definir o método de solução completo como em 3. A heurística começa com a construção de uma solução inicial utilizando-se o procedimento do algoritmo 1. Em seguida, o algoritmo de Busca Local é aplicado no laço das linhas (3–6) da heurística (3). A Busca Local iterada é aplicada usando-se três critérios distintos: até $\lceil N/100 \rceil$ tentativas de troca, depois $\lceil iter/1000 \rceil$ tentativas de troca, e então um número aleatório de trocas entre 1 e N .

Algorithm 3: Algoritmo completo da heurística GRASP com Busca Local.

```

1 Procedure GRASP_LS ( $N$ ,  $M$ ,  $T$ ,  $\alpha$ )
2    $s \leftarrow$  GRASP( $N$ ,  $M$ ,  $T$ ,  $\alpha$ )
3   for  $iter \leftarrow 1$  até  $MAX\_ITER$  do
4     Swap2LS( $s$ ,  $\lceil N/100 \rceil$ )
5     Swap2LS( $s$ ,  $\lceil iter/1000 \rceil$ )
6     Swap2LS( $s$ , randomInt(1, $N$ ))
7 return  $s$ 

```

Como consideração final, todas as seleções aleatórias se deram por distribuição uniforme.

4. Resultados computacionais

Os testes computacionais da heurística e da formulação matemática foram conduzidos nas instâncias de teste indicadas na definição do trabalho da disciplinas. Utilizou-se um

computador Intel 3612QM @ 2.10GHz, dispondo-se de 8 GB de memória principal. A heurística foi implementada em Python 3.7.4, e o modelo foi resolvido por meio do GLPK 4.65. O ambiente de testes foi o Arch Linux de 64 bits, com kernel linux-5.3.8.

A tabela 1 apresenta os resultados obtidos ao resolver-se a formulação matemática (1–8). Como critério de parada, utilizou-se um limite de tempo de 1 hora de execução, ou a prova da solução ótima pelo solver. O GLPK foi capaz de encontrar soluções inteiras apenas para as instâncias de menor porte, de até 60 tarefas e 5 máquinas. Para as demais instâncias, o resolvidor não foi capaz de encontrar nem o valor de relaxação ótima do modelo. Apenas a instância VFR10_15_1 teve sua solução ótima encontrada, após 1244,7 segundos de execução do resolvidor. As demais instâncias tiveram sua resolução interrompida por conta do limite de tempo estabelecido. Cabe-se ressaltar que foram habilitados os procedimentos de geração de cortes e procedimentos avançados para escolha da variável de *branching* do solver com as opções da linha de comando `--presol --bestp --mir`. Mais informações estão disponíveis na documentação do GNU GLPK.

Instância	BKS	Valor relaxação	Obj. solução inteira	GAP _{BKS} (%)
VFR10_15_1	1307	880,0	1307	0,0
VFR10_10_3	1592	687,0	1873	56,9
VFR_20_20_1	2270	1391,0	2573	42,6
VFR60_5_10	3663	382,0	3878	89,3
VFR100_60_1	9395	TL	–	∞
VFR500_40_1	28548	TL	–	∞
VFR500_60_3	31125	TL	–	∞
VFR600_20_1	31433	TL	–	∞
VFR700_20_10	36417	TL	–	∞

Tabela 1. Resultado obtido por meio do GLPK.

Os testes com a heurística GRASP e Busca Local foram conduzidos considerando-se os valores de $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Para cada valor de α e instância, executou-se 10 replicações do experimento, utilizando-se sementes de aleatoriedades distintas. Considerando a melhor média de valor das soluções, constatou-se que os resultados mais promissores são obtidos com a configuração de $\alpha = 0.0$. Os resultados individuais para cada instância de teste estão listados na tabela 2. Por linha, apresenta-se a instância testada e o BKS reportado na literatura, seguidos do valor de função objetivo médio e desvio percentual (em relação ao BKS) da solução construtiva encontrada pelo GRASP. Nas colunas seguintes, são apresentados os valor médio e desvio percentual (também em relação ao BKS), da melhor solução encontrada pelo GRASP com Busca Local, bem como o tempo de processamento total da heurística. A fim de manter o tempo de execução da heurística em até 5 minutos, estabeleceu-se um limite de repetições $MAX_ITER = 2140$. Esse valor foi estabelecido ao testar quantas iterações a heurística conseguiu realizar na maior instância de testes dentro desse limite de tempo.

Observou-se um desvio médio percentual de 16,9% das soluções iniciais com GRASP, em relação ao BKS da literatura, contra o desvio médio percentual de 4,9% da abordagem GRASP com Busca Local. Conclui-se que a Busca Local foi efetiva em melhorar as soluções encontradas pelo GRASP, estreitando a faixa de desvios médios de 5,08 ~ 27,29% para 0,15 ~ 7,76%.

Instância	BKS	Sol. GRASP		Sol. GRASP+BL		
		F.O.	Desvio (%)	F.O.	Desvio (%)	Tempo (seg.)
VFR10_15_1	1307	1424 \pm 0	8,95	1339, 6 \pm 18, 319	2,49	1, 5
VFR20_10_3	1592	2017 \pm 0	26,70	1687, 5 \pm 29, 304	6	2, 1
VFR20_20_1	2270	2715 \pm 0	19,60	2360, 1 \pm 33, 478	3,97	3, 9
VFR60_5_10	3663	3849 \pm 0	5,08	3668, 4 \pm 7, 291	0,15	3, 2
VFR60_10_3	3423	4357 \pm 0	27,29	3632, 6 \pm 62, 45	6,12	6, 0
VFR100_60_1	9395	11247 \pm 0	19,71	10008, 8 \pm 47, 123	6,53	57, 7
VFR500_40_1	28548	33119 \pm 0	16,01	30640, 6 \pm 67, 832	7,33	200, 4
VFR500_60_3	31125	36930 \pm 0	18,65	33539, 6 \pm 106, 966	7,76	298, 5
VFR600_20_1	31433	35473 \pm 0	12,85	32904, 4 \pm 69, 306	4,68	118, 4
VFR700_20_10	36417	40916 \pm 0	12,35	37857, 4 \pm 114, 996	3,96	140, 6

Tabela 2. Resultados médios da heurística para $\alpha = 0$.

5. Conclusões

O Flowshop Permutacional é um problema de otimização combinatória relevante, e métodos eficientes para solução de problema são de interesse acadêmico e prático. Ao que se pode observar, métodos de resolução exata com programação matemática são eficazes para resolver o PFSP, mas demandam muito tempo para obtenção de soluções minimamente factíveis. Dessa forma, propôs-se uma abordagem heurística baseada em solução construtiva inicial com GRASP, seguida de um refinamento com busca local randomizada. Os experimentos computacionais demonstraram a eficiência da heurística em encontrar soluções de alta qualidade do problema, e foi possível verificar que tanto o GRASP quanto a Busca Local foram capazes de contribuir para melhoramento das soluções.

Referências

- Feo, T. A., Resende, M. G., and Smith, S. H. (1994). A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42(5):860–878.
- Tseng, F. T., Stafford Jr, E. F., and Gupta, J. N. (2004). An empirical analysis of integer programming formulations for the permutation flowshop. *Omega*, 32(4):285–293.

Apêndice A – Média dos resultados computacionais para diversos α

Instância	BKS	Sol. GRASP			Sol. GRASP + Busca Local		
		α	Valor F.O.	GAP _{BKS} (%)	Valor F.O.	GAP _{BKS} (%)	Tempo (seg.)
VFR10_15_1	1307.00	0.00	1424 \pm 0	8.95	1339.6 \pm 18.319	2.49	1.5 \pm 0.04
VFR10_15_1	1307.00	0.20	1431.5 \pm 24.024	9.53	1354.2 \pm 23.011	3.61	1.4 \pm 0.03
VFR10_15_1	1307.00	0.40	1459.6 \pm 39.884	11.68	1364.2 \pm 28.944	4.38	1.5 \pm 0.04
VFR10_15_1	1307.00	0.60	1465.8 \pm 43.827	12.15	1346.1 \pm 42.331	2.99	1.4 \pm 0.03
VFR10_15_1	1307.00	0.80	1470.6 \pm 49.934	12.52	1362.9 \pm 30.205	4.28	1.5 \pm 0.04
VFR10_15_1	1307.00	1.00	1528.7 \pm 75.588	16.96	1342.2 \pm 28.867	2.69	1.5 \pm 0.03
VFR100_60_1	9395.00	0.00	11247 \pm 0	19.71	10008.8 \pm 47.123	6.53	57.7 \pm 0.59
VFR100_60_1	9395.00	0.20	11251.8 \pm 118.302	19.76	10054.5 \pm 70.099	7.02	57.7 \pm 0.42
VFR100_60_1	9395.00	0.40	11243.3 \pm 121.401	19.67	10039.1 \pm 54.017	6.86	57.9 \pm 0.52
VFR100_60_1	9395.00	0.60	11287.2 \pm 131.908	20.14	10040.9 \pm 73.843	6.87	58.5 \pm 0.87
VFR100_60_1	9395.00	0.80	11409.9 \pm 164.966	21.45	10048.8 \pm 69.904	6.96	58 \pm 1
VFR100_60_1	9395.00	1.00	11312.1 \pm 187.334	20.41	10057.8 \pm 55.519	7.05	58.2 \pm 0.99
VFR20_10_3	1592.00	0.00	2017 \pm 0	26.70	1687.5 \pm 29.304	6.00	2.1 \pm 0.05
VFR20_10_3	1592.00	0.20	2030.4 \pm 44.443	27.54	1685.8 \pm 23.223	5.89	2 \pm 0.03
VFR20_10_3	1592.00	0.40	1954.6 \pm 51.036	22.78	1682 \pm 21.417	5.65	2 \pm 0.03
VFR20_10_3	1592.00	0.60	1931 \pm 47.044	21.29	1690.8 \pm 39.6	6.21	2 \pm 0.04
VFR20_10_3	1592.00	0.80	1894.9 \pm 65.665	19.03	1692.3 \pm 32.094	6.30	2 \pm 0.02
VFR20_10_3	1592.00	1.00	2007.5 \pm 64.24	26.10	1682.7 \pm 24.157	5.70	2 \pm 0.04
VFR20_20_1	2270.00	0.00	2715 \pm 0	19.60	2360.1 \pm 33.478	3.97	3.9 \pm 0.07
VFR20_20_1	2270.00	0.20	2759.4 \pm 69.617	21.56	2355.8 \pm 41.214	3.98	3.9 \pm 0.08
VFR20_20_1	2270.00	0.40	2745.8 \pm 80.5	20.96	2350 \pm 25.573	3.52	3.9 \pm 0.08
VFR20_20_1	2270.00	0.60	2706.7 \pm 69.72	19.24	2376.6 \pm 31.178	4.70	3.9 \pm 0.06
VFR20_20_1	2270.00	0.80	2735.3 \pm 44.475	20.50	2362.9 \pm 26.236	4.09	3.8 \pm 0.05
VFR20_20_1	2270.00	1.00	2787.7 \pm 84.592	22.81	2366.9 \pm 38.766	4.27	3.9 \pm 0.07
VFR500_40_1	28548.00	0.00	33119 \pm 0	16.01	30640.6 \pm 67.832	7.33	200.4 \pm 8.47
VFR500_40_1	28548.00	0.20	33572.6 \pm 207.304	17.60	30753.7 \pm 111.634	7.73	200 \pm 4.51
VFR500_40_1	28548.00	0.40	33516.3 \pm 217.696	17.40	30697.4 \pm 107.934	7.53	197.2 \pm 1.52
VFR500_40_1	28548.00	0.60	33720.7 \pm 278.457	18.12	30681.7 \pm 127.513	7.47	198.4 \pm 1.59
VFR500_40_1	28548.00	0.80	33710.1 \pm 176.109	18.08	30688.4 \pm 101.606	7.50	199.6 \pm 3.45
VFR500_40_1	28548.00	1.00	33522.1 \pm 494.424	17.42	30741.5 \pm 113.56	7.68	200.9 \pm 7.53
VFR500_60_3	31125.00	0.00	36930 \pm 0	18.65	33539.6 \pm 106.966	7.76	298.5 \pm 4.31
VFR500_60_3	31125.00	0.20	36741.8 \pm 257.954	18.05	33624.6 \pm 167.947	8.03	300.7 \pm 3.79
VFR500_60_3	31125.00	0.40	36508.5 \pm 314.718	17.30	33535.1 \pm 81.036	7.74	299.2 \pm 3.89
VFR500_60_3	31125.00	0.60	36596.7 \pm 410.012	17.58	33576.6 \pm 71.104	7.88	300.6 \pm 3.38
VFR500_60_3	31125.00	0.80	36482.3 \pm 288.909	17.21	33490.7 \pm 96.158	7.60	298.3 \pm 3.3
VFR500_60_3	31125.00	1.00	36327 \pm 354.164	16.71	33530.5 \pm 65.58	7.73	298.7 \pm 2.61
VFR60_10_3	3423.00	0.00	4357 \pm 0	27.29	3632.6 \pm 62.45	6.12	6 \pm 0.06
VFR60_10_3	3423.00	0.20	4367.2 \pm 83.639	27.58	3637.4 \pm 67.612	6.26	6 \pm 0.14
VFR60_10_3	3423.00	0.40	4334.5 \pm 93.77	26.63	3630.7 \pm 55.041	6.07	6 \pm 0.08
VFR60_10_3	3423.00	0.60	4305 \pm 84.374	25.77	3608.3 \pm 50.557	5.41	5.9 \pm 0.11
VFR60_10_3	3423.00	0.80	4430.9 \pm 112.114	29.44	3603.6 \pm 72.537	5.28	6 \pm 0.08
VFR60_10_3	3423.00	1.00	4390.9 \pm 136.315	28.28	3626.3 \pm 54.214	5.94	6 \pm 0.09
VFR60_5_10	3663.00	0.00	3849 \pm 0	5.08	3668.4 \pm 7.291	0.15	3.2 \pm 0.09
VFR60_5_10	3663.00	0.20	3833.2 \pm 22.25	4.65	3667.9 \pm 5.971	0.13	3.2 \pm 0.13
VFR60_5_10	3663.00	0.40	3847.5 \pm 24.699	5.04	3672.2 \pm 8.574	0.25	3.1 \pm 0.05
VFR60_5_10	3663.00	0.60	3887.3 \pm 33.002	6.12	3674.4 \pm 8.03	0.31	3.2 \pm 0.06
VFR60_5_10	3663.00	0.80	3917.5 \pm 60.99	6.95	3668.6 \pm 7.152	0.15	3.2 \pm 0.03
VFR60_5_10	3663.00	1.00	3914 \pm 87.271	6.85	3665.6 \pm 1.897	0.07	3.1 \pm 0.05
VFR600_20_1	31433.00	0.00	35473 \pm 0	12.85	32904.4 \pm 69.306	4.68	118.4 \pm 1.86
VFR600_20_1	31433.00	0.20	35828.1 \pm 208.495	13.98	32930 \pm 65.09	4.76	121.1 \pm 5.56
VFR600_20_1	31433.00	0.40	35865.6 \pm 235.46	14.10	32999.7 \pm 123.094	4.98	119.3 \pm 1.99
VFR600_20_1	31433.00	0.60	36042.2 \pm 370.703	14.66	32982.4 \pm 68.39	4.93	119.2 \pm 1.82
VFR600_20_1	31433.00	0.80	36070.7 \pm 419.604	14.75	32932.5 \pm 134.142	4.77	123.1 \pm 9.14
VFR600_20_1	31433.00	1.00	35964.1 \pm 425.271	14.42	32990.1 \pm 97.588	4.95	122.6 \pm 7.68
VFR700_20_10	36417.00	0.00	40916 \pm 0	12.35	37857.4 \pm 114.996	3.96	140.6 \pm 2.03
VFR700_20_10	36417.00	0.20	40858.5 \pm 222.473	12.20	37792.3 \pm 93.295	3.78	140 \pm 3.16
VFR700_20_10	36417.00	0.40	41003.3 \pm 407.564	12.59	37865.9 \pm 79.689	3.98	139 \pm 2.11
VFR700_20_10	36417.00	0.60	41196.8 \pm 355.395	13.13	37798.9 \pm 87.46	3.79	142.6 \pm 9.19
VFR700_20_10	36417.00	0.80	41036.6 \pm 281.03	12.69	37882.2 \pm 110.235	4.02	140.3 \pm 3.43
VFR700_20_10	36417.00	1.00	41214.5 \pm 385.111	13.17	37807.6 \pm 124.189	3.82	139.8 \pm 2.51