

## Lista de Exercícios 1: Soluções

### Exercício 1 (Vizinhanças, 3 pt)

Para simplificar vamos introduzir um pouco de notação. Para uma permutação  $\pi = (\pi_1, \dots, \pi_n)$  define um segmento

$$\pi_{a:b} = \begin{cases} (\pi_a, \pi_{a+1}, \dots, \pi_b), & \text{caso } a \leq b, \\ (\pi_a, \pi_{a-1}, \dots, \pi_b), & \text{caso } a > b, \end{cases}$$

e para permutações  $\pi$  e  $\rho$  seja  $[\phi \rho]$  a sua concatenação.

Define o  $i$ -ésimo vizinho na vizinhança  $k$ -reverse por

$$R_k(\pi, i) = [\pi_{1:i-1} \pi_{i+k-1:i} \pi_{i+k:n}]$$

para  $i \in [n]$ ,  $i+k \leq n+1$ . Em particular, a vizinhança 1-flip é  $F(\pi, i) = R_2(\pi, i)$ . Ainda define a vizinhança 1-shift com parâmetro  $s$

$$S_s(\pi, i) = \begin{cases} [\pi_{1:i-s-1} \pi_i \pi_{i-s:i-1} \pi_{i+1:n}], & \text{caso } s < 0, \\ [\pi_{1:i-1} \pi_{i+1:i+s} \pi_i \pi_{i+s+1:n}], & \text{caso } s > 0, \end{cases}$$

para  $i \in [n]$ ,  $s \in [-(i-1), n-i] \setminus \{0\}$ . (Nota que essas definições produzem vizinhos individuais; em buscas locais a vizinhança de uma permutação  $\pi$  é o conjunto de todos vizinhos, e.g.  $F(\pi) = \{F(\pi, i) \mid i \in [n-1]\}$ .)

- Podemos verificar que  $F(F(\pi, i), i) = \pi$ , logo  $F$  é simétrica. Também é conhecido que toda permutação pode ser representado por transposições adjacentes, e como um vizinho em  $F$  é obtido exatamente por uma transposição adjacente, segue a conectividade, e com isso também que  $F$  é fracamente otimamente conectada. (Uma estratégia direta que demonstrar isso é por indução: para ir de uma permutação  $\pi$  para outro  $\rho$  assume que temos um prefixo  $\sigma$  em comum. Depois mostra que é possível, via no máximo  $n$  transposições obter um prefixo  $\sigma'$  em comum, com  $|\sigma'| = |\sigma| + 1$ .) Porém,  $F$  não é exata, como o exemplo abaixo vai mostrar.
- Similar ao  $F$ , a vizinhança  $R_k$  satisfaz  $R_k(R_k(\pi, i), i) = \pi$ , que demonstra a simetria. Para  $k = 2$  temos  $R_2 = F$  e os argumentos do item a) aplicam. Para  $k$  ímpar porém, um elemento numa posição par/ímpar sempre ficar numa posição par/ímpar, logo  $R_k$  não é conectada nesse caso. Para  $k$  par podemos argumentar usando a distância de Kendall-tau (“bubble sort distance”, ver Notas de aula) e observar que a diferença depois de aplicar o operador  $R_k$  sempre é par (caso o segmento em questão que  $\tau$  inversões antes vai ter  $6 - \tau$  inversões depois, com todo resto igual, logo a diferença é  $6 - 2\tau$ ). Logo neste caso também não é conectada, porque tem permutações com distância ímpar. O exemplo abaixo mostra que  $R_k$  em geral não é exato. O exemplo também tem um único mínimo global, logo também não é fracamente otimamente conectada.
- Essa vizinhança possui como caso particular a vizinhança  $F$ , então o caso a) aplica.

Para demonstrar que todas vizinhanças acima não são exatas, temos que exibir um mínimo local deles, que não é um mínimo global. Isso foi feito experimentalmente: considera a matriz

$$\begin{pmatrix} 10 & 5 & 9 & 10 \\ 7 & 8 & 7 & 8 \\ 10 & 9 & 4 & 3 \\ 8 & 8 & 6 & 6 \end{pmatrix}$$

Tabela 1: Resultados 1 ||  $\sum U_j$ .

$n$	FI			BI			Ex.
	it.	t(s)	r.d.(%)	it.	t(s)	r.d.(%)	
100	20.5	0.0	15.4	20.1	0.0	15.8	0.00
200	35.4	0.0	17.3	36.5	0.0	16.6	0.00
300	63.0	0.0	15.1	61.8	0.2	15.6	0.00
400	97.1	0.1	13.8	93.4	0.6	14.2	0.00
500	190.5	0.6	6.5	201.4	2.1	6.1	0.00
---	81.3	0.1	13.6	82.6	0.6	13.7	0.00

de valor 42 e a matriz permutada, com permutação  $\pi = (3\ 4\ 1\ 2)$

$$\begin{pmatrix} 4 & 3 & 10 & 9 \\ 6 & 6 & 8 & 8 \\ 9 & 10 & 10 & 5 \\ 7 & 8 & 7 & 8 \end{pmatrix}$$

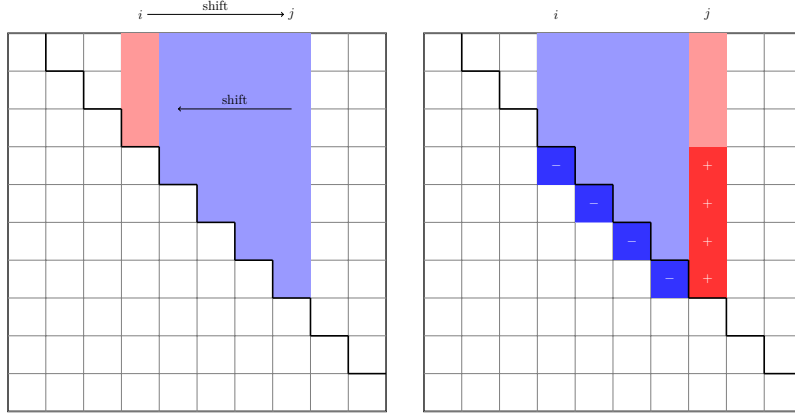
de valor 43. É simples de verificar a matriz permutada é mínimo local estrito das vizinhanças  $R_2, R_3, R_4$  acima, logo nenhuma é exata. Como o LOP é NP-completo, logo não é surpreendente as vizinhanças acima não serem exatas, porque eles são polinomiais. Uma vizinhança exata para uma delas implicaria que PLS = NP.

**Exercício 2 (Busca local para um problema polinomial, 3 pt)**

- a) Tabela 1 tem todos resultados. Podemos ver que tanto primeira quanto melhor melhora tem aproximadamente um desvio relativo médio da solução ótima de 13.5%. Não tem grande diferenças entre as duas estratégias, porque a busca local diminui o número de tarefas atrasadas por no máximo um, então o progresso nas duas buscas é similar. O tempo de execução é curto, e maior para melhor melhora, como era para esperar, e aumenta significativamente com o número de iterações.
- b) O algoritmo exato tem um tempo negligível e soluções aproximadamente 13.5% melhores comparado com as buscas locais.

**Exercício 3 (Busca local para um problema NP-completo, 4 pt)**

Vamos começar discutir um cálculo eficiente do novo valor da função objetivo sobre um *shift* do elemento na posição  $i$  da permutação  $\pi$  para posição  $j$ . Assume que a matriz  $m$  é a matriz permutada atual. Assume primeiramente que  $i < j$ . Então as colunas  $k \in [i + 1, j]$  se movem para esquerda e perdem por isso um elemento. Como no mesmo tempo nas colunas o elemento de linha  $i$  vai para linha  $j$  eles perdem é o elemento  $m_{ik}$ . A coluna  $i$  ganha  $j - i$  elementos, pelo mesmo argumento anterior são os elementos  $m_{ki}$  com  $k \in [i + 1, j]$ . Uma ilustração da situação é



Em resumo a função objetivo é diferente por

$$\delta^+ = \sum_{k \in [i+1, j]} m_{ki} - \sum_{k \in [i+1, j]} m_{ik}.$$

Por um argumento similar, caso  $i > j$  obtemos

$$\delta^- = \sum_{k \in [j, i-1]} m_{ik} - \sum_{k \in [j, i-1]} m_{ki}.$$

Isso custa  $O(|i - j|) = O(n)$  por shift. Como se trata de somas de linhas e colunas, podemos introduzir duas matrizes com as somas parciais

$$r_{ij} = \sum_{k \in [j]} m_{ik}$$

$$c_{ij} = \sum_{k \in [i]} m_{kj}$$

Com isso podemos simplificar para

$$\delta^+ = (c_{ji} - c_{ii}) - (r_{ij} - r_{ii})$$

$$\delta^- = (r_{i, i-1} - r_{i, j-1}) - (c_{i-1, i} - c_{j-1, i})$$

e agora uma avaliação de um *shift* é possível em tempo  $O(1)$ .

- Os resultados se encontram nas tabelas 2 e 3. Em geral podemos ver que a primeira melhora produz melhores resultados: temos um desvio relativo do melhor valor conhecido de 1.60%, o melhor de 1.28% em aproximadamente 5 s, enquanto a melhor melhora termina em média com 1.75%, o melhor de 1.40% e precisa aprox. 25 s.
- O teste foi alterado para  $p \in [0, 5]/100$ , porque em testes valores altos de  $p$  se mostraram inferiores, cada uma com um tempo limite de 90s, mas com somente 10 replicações, para limitar o tempo de execução. A Tabela 4 tem os resultados, resumidos por valor de  $p$ . Podemos ver que a BLMR piora com probabilidades  $p$  maiores, e o melhor valor, de fato é  $p = 0$ . Neste caso tem um desvio relativo médio de 1.57%, e um melhor valor de 1.40%. Então os valores médios são um pouco melhor que da primeira melhora, mas o melhor valor é igual. A diferença nesse caso não vem dos movimentos aleatórios, porque  $p = 0$ , mas da estratégia de busca diferente: a BLMR seleciona entre os movimentos que melhoram um aleatoriamente.

Tabela 2: Resultados para busca primeira melhora.

Inst.	$\bar{t}$	$\sigma_t$	$\bar{i}$	$\sigma_i$	$\bar{\Delta}$	$\sigma_{\Delta}$	r.d.
be75eec	4.8	0.6	13,745	401	1.64	0.14	1.31
be75np	4.9	0.6	13,938	374	1.71	0.14	1.35
be75oi	4.7	0.6	13,579	419	1.67	0.17	1.26
be75tot	4.9	0.7	14,176	450	1.57	0.12	1.23
stabu70	4.9	0.7	13,952	382	1.60	0.14	1.28
stabu74	4.8	0.6	14,001	345	1.65	0.12	1.28
stabu75	4.9	0.6	14,007	379	1.72	0.13	1.35
t59b11xx	4.6	0.6	12,067	308	1.67	0.13	1.41
t59d11xx	4.7	0.6	12,382	304	1.40	0.10	1.12
t59f11xx	4.6	0.6	12,299	314	1.53	0.13	1.15
t59i11xx	4.7	0.6	12,541	342	1.61	0.13	1.24
t59n11xx	4.7	0.6	12,182	330	1.65	0.13	1.40
t65b11xx	4.8	0.5	14,016	407	1.57	0.14	1.26
t65d11xx	4.8	0.6	13,502	317	1.52	0.13	1.27
t65f11xx	4.9	0.7	13,890	366	1.52	0.13	1.17
t65i11xx	4.9	0.7	13,844	390	1.65	0.13	1.31
t65l11xx	4.1	0.4	10,133	284	2.02	0.18	1.67
t65n11xx	4.8	0.6	13,005	351	1.65	0.14	1.29
t65w11xx	5.1	0.7	14,362	386	1.50	0.11	1.15
t69r11xx	4.8	0.7	12,917	290	1.71	0.14	1.36
t70b11xx	4.8	0.7	13,381	372	1.59	0.12	1.40
t70d11xx	4.9	0.6	14,091	408	1.48	0.11	1.26
t70d11xxb	5.0	0.7	14,256	383	1.52	0.13	1.17
t70f11xx	4.9	0.5	13,723	364	1.43	0.10	1.18
t70i11xx	4.9	0.5	13,995	360	1.56	0.12	1.22
t70k11xx	4.9	0.7	13,457	360	1.53	0.12	1.23
t70l11xx	4.3	0.6	10,741	304	2.02	0.15	1.70
t70n11xx	4.7	0.5	13,290	374	1.58	0.14	1.32
t70u11xx	4.5	0.6	11,716	297	1.52	0.12	1.14
t70w11xx	5.1	0.7	14,355	422	1.52	0.12	1.22
t70x11xx	5.0	0.7	14,413	365	1.46	0.11	1.15
t74d11xx	4.9	0.7	14,252	390	1.51	0.12	1.21
t75d11xx	4.9	0.6	14,243	348	1.47	0.12	1.21
t75e11xx	5.1	0.6	14,290	419	1.60	0.15	1.26
t75i11xx	4.9	0.6	13,986	340	1.56	0.12	1.24
t75k11xx	4.7	0.6	12,912	363	1.62	0.13	1.23
t75n11xx	4.7	0.5	13,169	360	1.73	0.14	1.40
t75u11xx	4.8	0.7	13,240	357	1.54	0.13	1.33
tiw56n54	4.8	0.6	13,421	352	1.59	0.11	1.31
tiw56n58	4.9	0.7	13,557	348	1.51	0.13	1.19
tiw56n62	4.9	0.7	13,699	365	1.53	0.12	1.17
tiw56n66	4.9	0.6	13,764	361	1.52	0.11	1.21
tiw56n67	4.9	0.5	14,001	361	1.65	0.13	1.39
tiw56n72	4.9	0.7	13,966	369	1.62	0.14	1.21
tiw56r54	4.9	0.6	13,513	307	1.63	0.11	1.36
tiw56r58	4.9	0.7	13,554	359	1.49	0.12	1.20
tiw56r66	4.9	0.6	13,700	356	1.59	0.13	1.28
tiw56r67	4.9	0.6	13,830	362	1.65	0.12	1.25
tiw56r72	4.9	0.7	13,890	338	1.62	0.13	1.29
usa79	5.0	0.7	14,255	385	1.82	0.15	1.52
—	4.8	0.6	13,464	360	1.60	0.13	1.28

Tabela 3: Resultados para busca melhor melhora.

Inst.	$\bar{t}$	$\sigma_t$	$\bar{i}$	$\sigma_i$	$\bar{\Delta}$	$\sigma_{\Delta}$	r.d.
be75eec	24.3	1.0	1,802	74	1.77	0.15	1.44
be75np	24.1	1.0	1,786	76	1.86	0.15	1.49
be75oi	24.6	0.9	1,824	70	1.80	0.17	1.36
be75tot	24.6	1.1	1,813	82	1.73	0.12	1.48
stabu70	24.2	1.1	1,782	78	1.74	0.14	1.34
stabu74	24.4	1.0	1,798	73	1.74	0.15	1.38
stabu75	24.9	1.6	1,794	76	1.83	0.13	1.54
t59b11xx	23.4	1.1	1,715	75	1.86	0.15	1.47
t59d11xx	23.5	1.1	1,723	79	1.54	0.13	1.30
t59f11xx	23.6	1.0	1,731	72	1.71	0.12	1.39
t59i11xx	23.8	1.0	1,737	68	1.78	0.14	1.36
t59n11xx	23.6	1.1	1,728	77	1.82	0.13	1.47
t65b11xx	24.5	1.0	1,793	73	1.72	0.14	1.37
t65d11xx	24.1	1.1	1,766	83	1.65	0.13	1.35
t65f11xx	24.5	1.1	1,793	79	1.68	0.13	1.30
t65i11xx	24.5	1.1	1,790	78	1.84	0.16	1.48
t65l11xx	22.6	1.1	1,646	79	2.15	0.19	1.52
t65n11xx	24.1	1.3	1,760	93	1.82	0.15	1.46
t65w11xx	24.6	1.0	1,788	74	1.66	0.13	1.29
t69r11xx	24.3	1.0	1,769	74	1.87	0.14	1.49
t70b11xx	24.5	1.1	1,784	81	1.77	0.14	1.41
t70d11xx	24.6	1.1	1,789	77	1.61	0.13	1.26
t70d11xxb	24.8	1.1	1,803	77	1.65	0.12	1.25
t70f11xx	24.5	1.1	1,784	78	1.54	0.13	1.30
t70i11xx	24.6	1.1	1,785	80	1.72	0.15	1.38
t70k11xx	24.4	1.0	1,774	69	1.71	0.14	1.38
t70l11xx	23.2	0.9	1,683	65	2.18	0.18	1.73
t70n11xx	24.2	1.0	1,762	74	1.79	0.13	1.47
t70u11xx	23.4	0.9	1,704	69	1.66	0.13	1.43
t70w11xx	24.9	1.2	1,802	86	1.66	0.11	1.39
t70x11xx	25.0	1.1	1,812	78	1.59	0.13	1.26
t74d11xx	24.9	1.0	1,807	75	1.62	0.12	1.28
t75d11xx	24.7	1.0	1,787	71	1.63	0.11	1.42
t75e11xx	24.9	1.2	1,803	81	1.78	0.15	1.43
t75i11xx	24.6	1.1	1,785	76	1.71	0.13	1.35
t75k11xx	24.2	1.0	1,754	76	1.80	0.14	1.40
t75n11xx	24.4	1.2	1,769	83	1.86	0.15	1.49
t75u11xx	24.4	1.3	1,767	89	1.64	0.12	1.34
tiw56n54	24.5	1.1	1,772	82	1.73	0.13	1.31
tiw56n58	24.7	1.3	1,788	93	1.67	0.14	1.31
tiw56n62	24.7	1.1	1,793	77	1.67	0.13	1.40
tiw56n66	24.5	0.9	1,774	67	1.72	0.14	1.46
tiw56n67	24.9	1.2	1,801	88	1.78	0.14	1.42
tiw56n72	24.9	1.1	1,811	78	1.70	0.14	1.37
tiw56r54	24.3	1.2	1,763	88	1.79	0.14	1.47
tiw56r58	24.7	1.0	1,784	73	1.64	0.13	1.29
tiw56r66	24.6	1.0	1,777	76	1.76	0.13	1.39
tiw56r67	24.8	1.1	1,792	79	1.80	0.13	1.48
tiw56r72	25.1	1.0	1,812	74	1.75	0.15	1.38
usa79	24.9	1.3	1,817	83	1.98	0.16	1.54
—	24.4	1.1	1,776	78	1.75	0.14	1.40

Tabela 4: Resultados para busca local monotona randomizada.

p	$\bar{t}$	$\sigma_t$	$\bar{i}$	$\sigma_i$	$\bar{\Delta}$	$\sigma_{\Delta}$	r.d.
0.00	90.0	0.0	6,948	73	1.57	0.12	1.40
0.01	90.0	0.0	7,107	77	1.59	0.12	1.40
0.02	90.0	0.0	7,262	79	1.63	0.11	1.45
0.03	90.0	0.0	7,395	82	1.69	0.12	1.51
0.04	90.0	0.0	7,494	80	1.78	0.12	1.58
0.05	90.0	0.0	7,598	80	1.89	0.11	1.71