

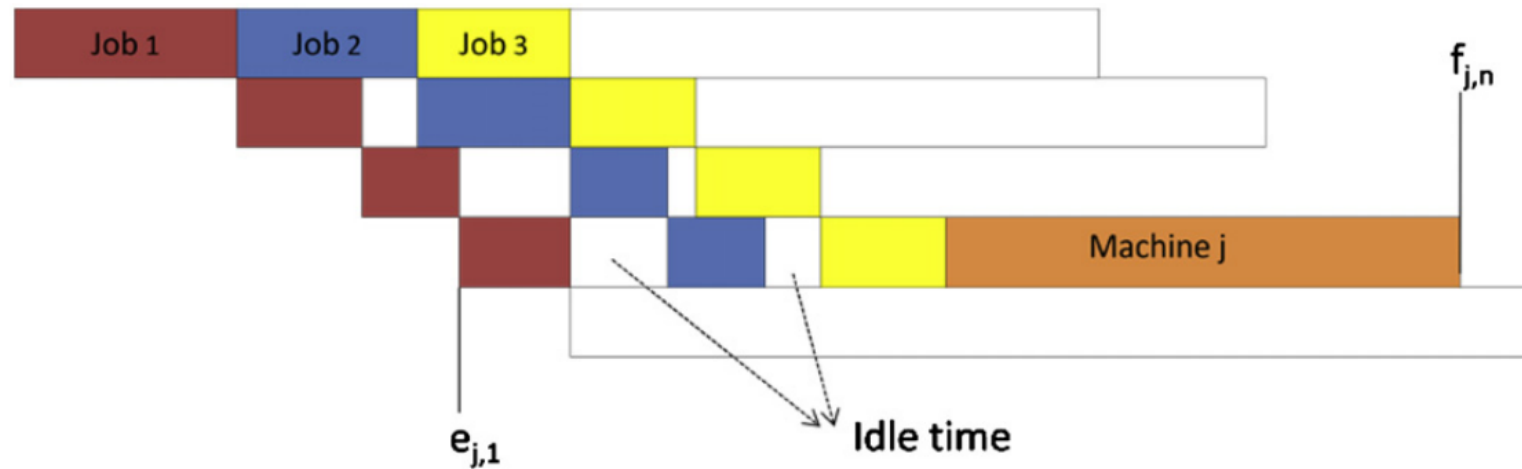
Comparing three-step heuristics for the permutation flow shop problem

Imma Ribas, Ramon Companys, Xavier Tort-Martorell

Computers & Operations Research (2010)

Contextualização

- Permutation flow shop



- Objetivo: Reduzir o tempo total de produção (Makespan)

Estrutura da heurística proposta

- A heurística proposta é dividida em **três etapas**;
- As duas primeiras são **variações** da heurística **NEH**.
- Terceira é a etapa onde ocorre a busca, foco da apresentação.
- São apresentadas várias alternativas para cada etapa, e no fim é desenvolvida uma análise estatística dos resultados para avaliar se a diferença entre as heurísticas é relevante.

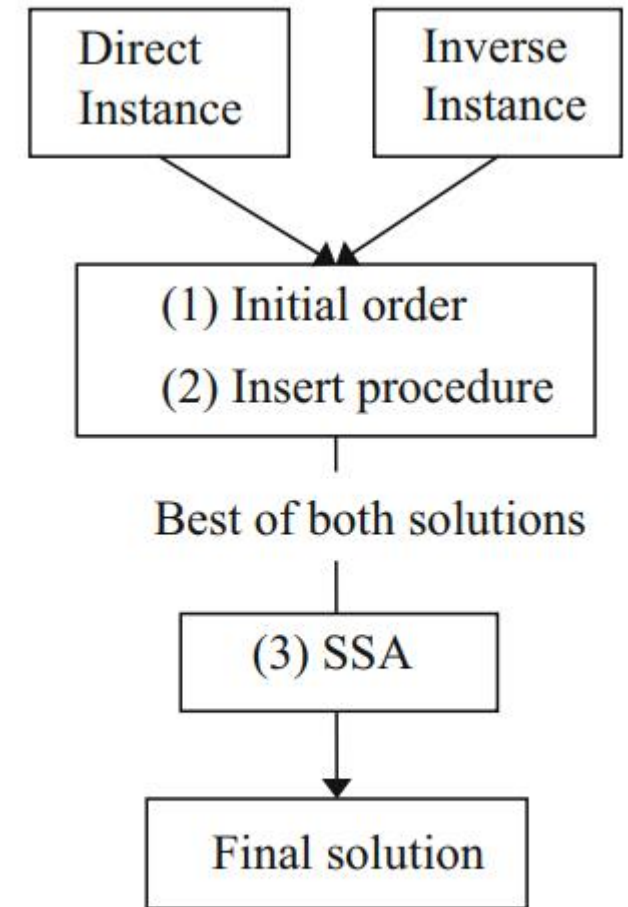


Fig. 1. Three-step procedure.

Heurística NEH

Jobs processing times matrix:

J1	3	7	4	(14)
J2	6	2	3	(11)
J3	9	7	3	(19)
J4	8	6	2	(16)
J5	9	7	4	(20)

Ordenados pela soma dos tempos de processamento

Essa sequência define a ordem que os serviços serão inseridos na programação.

First sort (jobs): 5 3 4 1 2 3 4 1 2 4 1 2 1 2 2

Second part (sequences):

5 (20)	3 5 (29)	4 5 3 (36)	1 5 3 4 x (37)	2 1 5 3 4 (43)
	5 3 x (28)	5 4 3 (36)	5 1 3 4 (38)	1 2 5 3 4 (43)
		5 3 4 x (34)	5 3 1 4 (40)	1 5 2 3 4 (43)
			5 3 4 1 (43)	1 5 3 2 4 (43)
				1 5 3 4 2 x (40)

x : best partial sequence (with partial total time)

Cada serviço é inserido na sua melhor posição.

Variações da primeira etapa

Ordenação inicial das tarefas de acordo com 4 variantes:

- KK: ordenação em ordem decrescente de $c_i = \min(a_i; b_i)$

sendo:

$$a_i = \sum_{j=1}^m ((m-1) \cdot (m-2)/2 + m-j) \cdot p_{ji}$$

$$b_i = \sum_{j=1}^m ((m-1) \cdot (m-2)/2 + j-1) \cdot p_{ji}$$

- LPT: ordenação em ordem decrescente de $P_i = \sum_{j=1}^m p_{ji}$
- NM: ordenação em ordem decrescente de $\bar{P}_i = P_i - \max_h \{BT_{hi}\}$

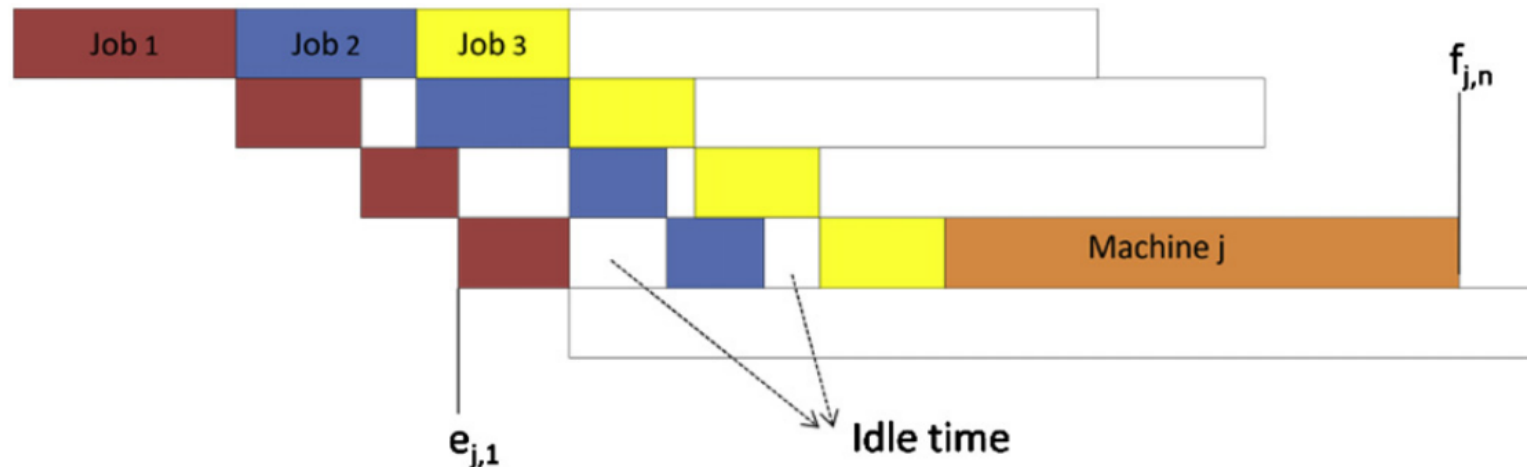
sendo BT_{hi} o limite inferior do tempo de espera entre o término da tarefa i para o início da tarefa h em todas as máquinas consideradas

- RA: uma sequência inicial é gerada randomicamente

Variações da segunda etapa

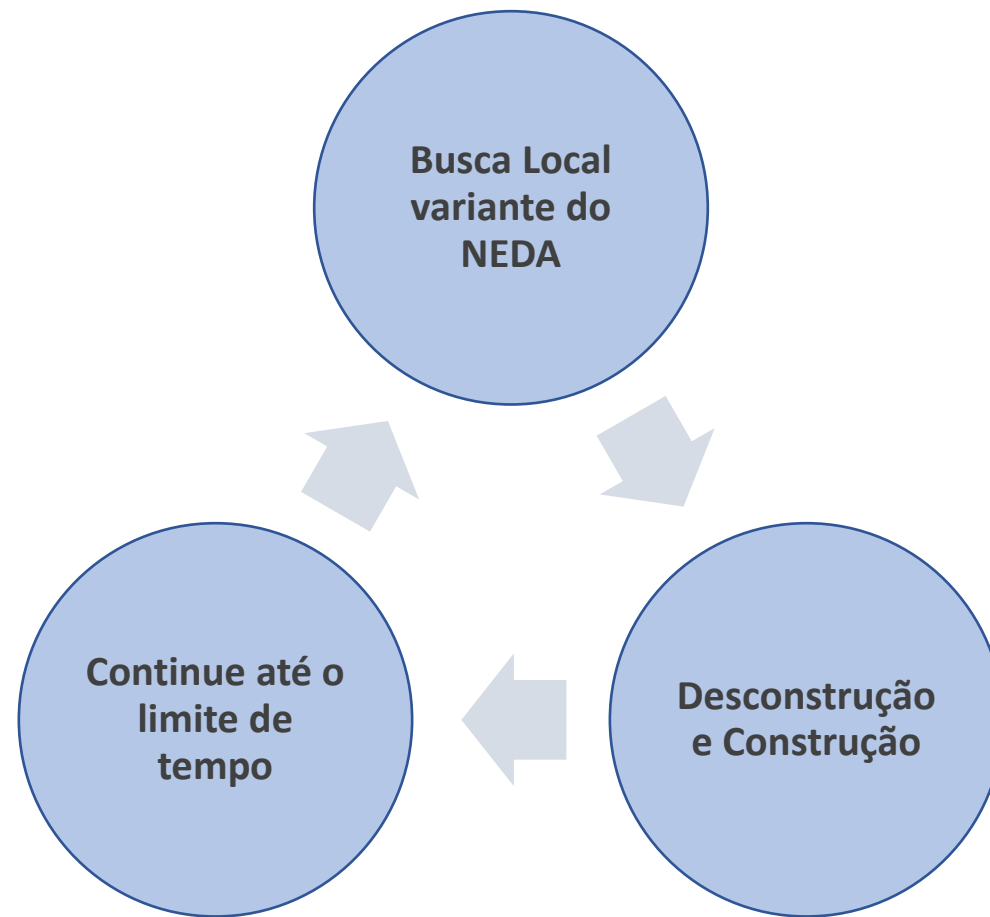
Na segunda é idêntica ao NEH, com a inclusão de duas técnicas de desempate.

TIT: Reduz o tempo parado (Total idle time);

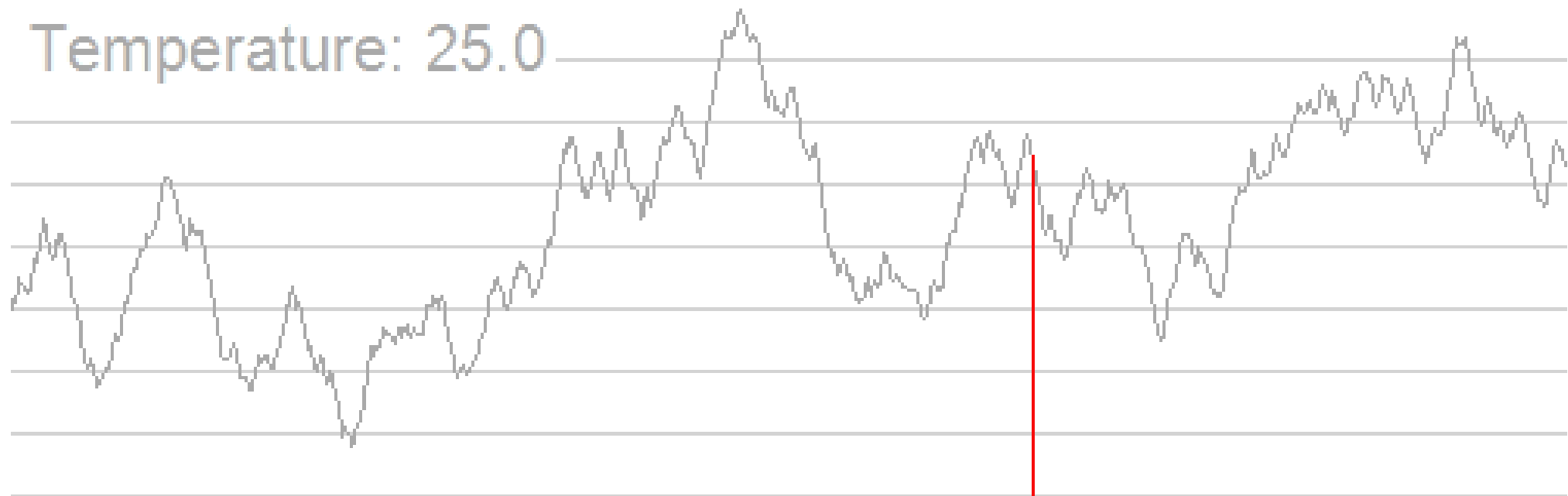


KK1: a posição escolhida depende dos fatores A_i e B_i .

3º Passo – Soft Simulated Annealing



Simulated annealing



https://en.wikipedia.org/wiki/File:Hill_Climbing_with_Simulated_Annealing.gif

3º Passo – Soft Simulated Annealing

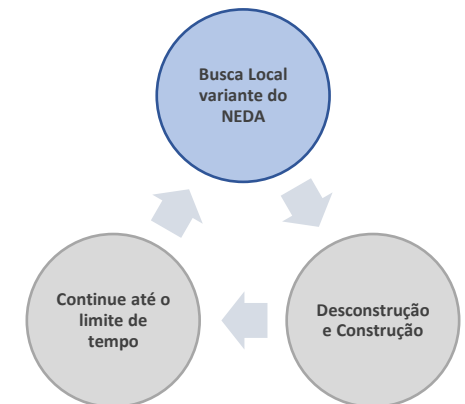
- NEDA: Algoritmo de Descida Não Exaustivo

Tenta melhorar a solução (π) trocando quaisquer duas posições na sequência. Pode gerar até $n.(n-1)/2$ vizinhos. Se uma nova permutação melhorar o valor da função objetivo, ela se torna a nova solução atual e o processo continua até que todas as posições tenham sido permutadas e a melhora já não está acontecendo. Neste procedimento, **a vizinhança é sempre explorada na mesma ordem**.

Jobs processing times matrix:					
J1	3	7	4	(14)	
J2	6	2	3	(11)	
J3	9	7	3	(19)	
J4	8	6	2	(16)	
J5	9	7	4	(20)	
First sort (jobs): 5 3 4 1 2 3 4 1 2 4 1 2 1 2 2					
Second part (sequences):	5 (20)	3 5 (29)	4 5 3 (36)	1 5 3 4 x (37)	2 1 5 3 4 (43)
		5 3 x (28)	5 4 3 (36)	5 1 3 4 (38)	1 2 5 3 4 (43)
			5 3 4 x (34)	5 3 1 4 (40)	1 5 2 3 4 (43)
				5 3 4 1 (43)	1 5 3 2 4 (43)
x : best partial sequence (with partial total time)				1 5 3 4 2 x (40)	

Solução de partida:

15342



3º Passo – Soft Simulated Annealing

- NEDA: Algoritmo de Descida Não Exaustivo

vizinhança: ALLPAIRS

15342 -> 51342 | 35142 | 45312 | 25341

3º Passo – Soft Simulated Annealing

- NEDA: Algoritmo de Descida Não Exaustivo

vizinhança: ALLPAIRS

15342 -> 51342 | 35142 | 45312 | 25341

↓
25341 -> 52341 | 35241 | 45321 | 15342 | 23541 | 24351

3º Passo – Soft Simulated Annealing

- NEDA: Algoritmo de Descida Não Exaustivo

vizinhança: ALLPAIRS

15342 -> 51342 | 35142 | 45312 | 25341

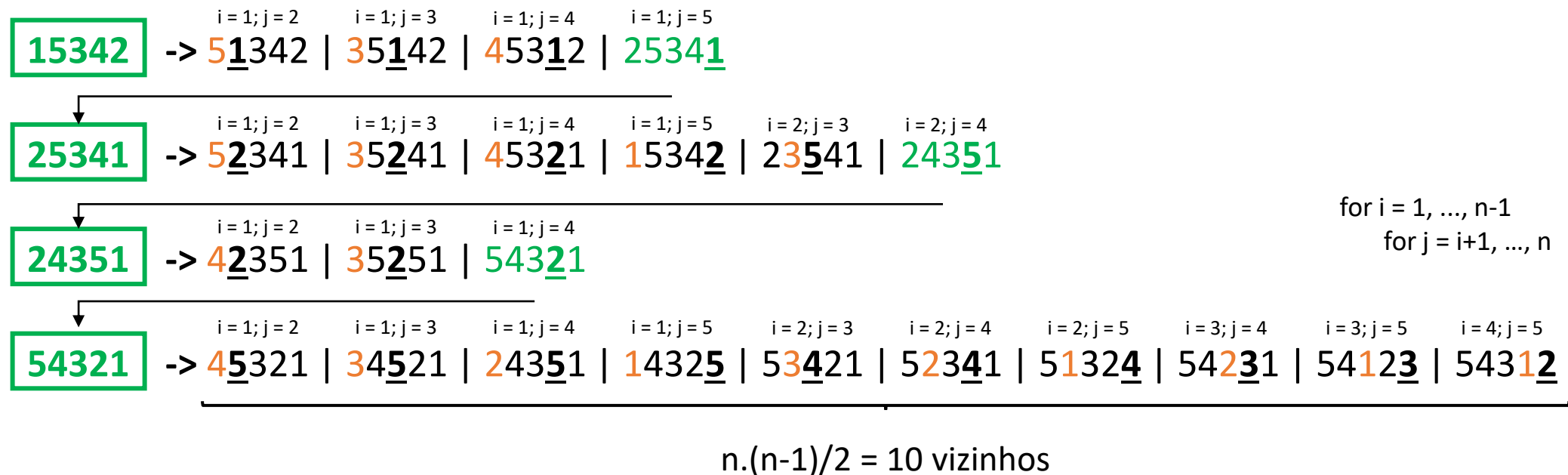
↓
25341 -> 52341 | 35241 | 45321 | 15342 | 23541 | 24351

↓
24351 -> 42351 | 35251 | 54321

3º Passo – Soft Simulated Annealing

- NEDA: Algoritmo de Descida Não Exaustivo

vizinhança: ALLPAIRS



Passando por todas as permutações e não encontrando nenhuma que melhore o objetivo, o algoritmo finaliza e a solução final é a última obtida (no exemplo, a **54321**)

3º Passo – Soft Simulated Annealing

- NEDA: Algoritmo de Descida Não Exaustivo

vizinhança: ALLPAIRS

1ª variação

for i = 1, ..., n-1
for j = i+1, ..., n

i = 1; j = 2

i = 1; j = 3

i = 1; j = 4

i = 1; j = 5

15342
 π

-> 51342 | 35142 | 45312 | 25341

π

$C_{\max}(25341) < C_{\max}(15342)$

3º Passo – Soft Simulated Annealing

- NEDA: Algoritmo de Descida Não Exaustivo

vizinhança: ALLPAIRS

1ª variação

π 15342 \rightarrow $\overset{i=1;j=2}{5\underline{1}342}$ | $\overset{i=1;j=3}{35\underline{1}42}$ | $\overset{i=1;j=4}{453\underline{1}2}$ | $\overset{i=1;j=5}{\textcolor{green}{2}534\underline{1}}$ | $\overset{i=2;j=3}{23\underline{5}41}$ | $\overset{i=2;j=4}{243\underline{5}1}$ | $\overset{i=2;j=5}{\textcolor{green}{2}134\underline{5}}$
 π π π

$$C_{\max}(25341) < C_{\max}(15342)$$

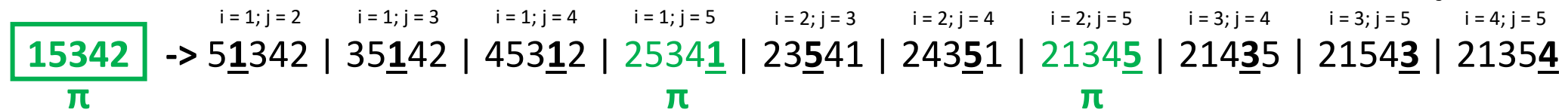
$$C_{\max}(21345) < C_{\max}(25341)$$

3º Passo – Soft Simulated Annealing

- NEDA: Algoritmo de Descida Não Exaustivo

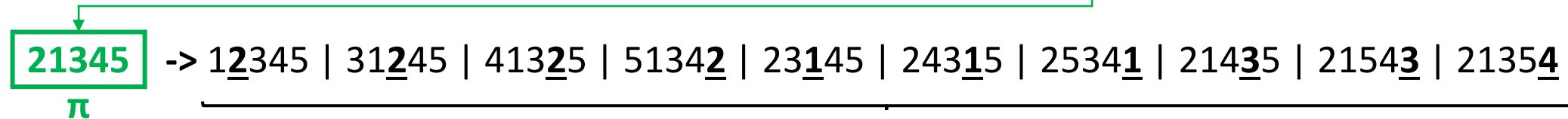
vizinhança: ALLPAIRS

1ª variação



$$C_{\max}(25341) < C_{\max}(15342)$$

$$C_{\max}(21345) < C_{\max}(25341)$$



$$n.(n-1)/2 = 10 \text{ vizinhos}$$

Passando por todas as permutações e não encontrando nenhuma outra solução que melhore o objetivo, o algoritmo finaliza e a solução final é a última obtida (no exemplo, a **21345**)

fim do algoritmo

3º Passo – Soft Simulated Annealing

- Variante do NEDA para o Soft Simulated Annealing (SSA)

Vetor auxiliar, **revolver**, que permite a **exploração aleatória** da vizinhança.

Codifica as posições das vizinhanças da solução corrente e as novas posições é a que serão usadas na aplicação do NEDA.

Se toda a vizinhança da solução corrente for explorada sem melhoria, o processo reinicia.

Soluções **empatadas** são aceitas sob uma **probabilidade α** , evitando que a busca fique presa em um ótimo local, fazendo com que o algoritmo caminhe sobre platôs.

A etapa de melhoria termina quando os **empates** atingem um número pré-definido **β** ou se não houver mudanças na solução corrente.

3º Passo – Soft Simulated Annealing

- Variante do NEDA para o Soft Simulated Annealing (SSA)

Solução de partida π : **15342**

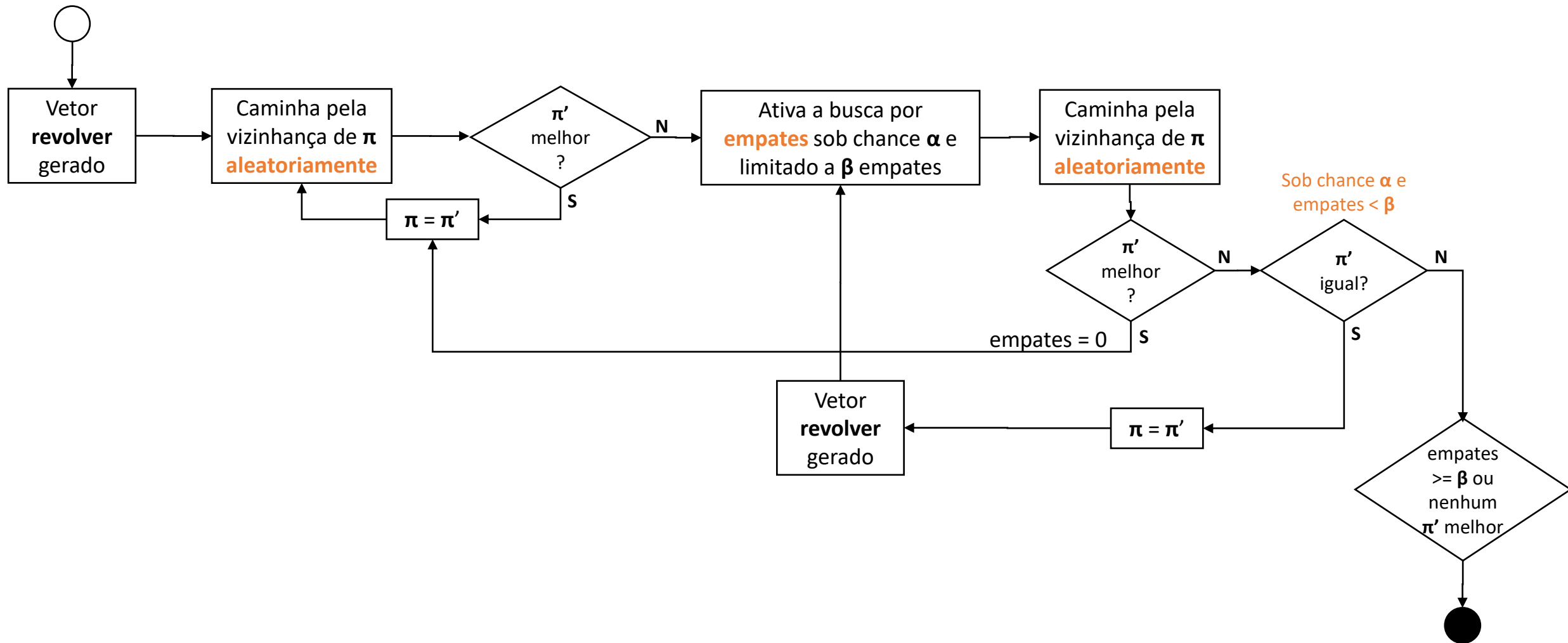
Posições originais: 12345 (i, j)

Vetor **revolver**: **24351** (i_{rev} , j_{rev})

		$i = 1; j = 2$	$i = 1; j = 3$	$i = 1; j = 4$	$i = 1; j = 5$	$i = 2; j = 3$	$i = 2; j = 4$	$i = 2; j = 5$	$i = 3; j = 4$	$i = 3; j = 5$	$i = 4; j = 5$
NEDA	15342	-> 5 <u>1</u> 342	35 <u>1</u> 42	453 <u>1</u> 2	<u>2</u> 534 <u>1</u>	13 <u>5</u> 42	143 <u>5</u> 2	1234 <u>5</u>	154 <u>3</u> 2	<u>1</u> 524 <u>3</u>	<u>1</u> 532 <u>4</u>
Variante NEDA	15342	-> 143 <u>5</u> 2	13 <u>5</u> 42	1234 <u>5</u>	<u>5</u> 1342	15 <u>4</u> 32	<u>1</u> 532 <u>4</u>	<u>4</u> 5312	<u>1</u> 524 <u>3</u>	<u>3</u> 5142	<u>2</u> 534 <u>1</u>

O NEDA adaptado gera $n.(n-1)/2$ permutações iguais às do NEDA original: o que muda é a ordem na qual estas são exploradas pelo algoritmo.

3º Passo – Soft Simulated Annealing



3º Passo – Soft Simulated Annealing

- Desconstrução e construção

Após a etapa de busca local, a solução π atual passa por uma fase de desconstrução e construção.

Na fase de *desconstrução*, d tarefas são removidas **aleatoriamente** de π e armazenadas em π'' uma atrás da outra.

Então, na etapa de *construção*, as d tarefas vão retornando à solução, uma a uma, usando o procedimento de inserção da **segunda etapa**.

A solução resultante π^* é a melhor solução obtida pela busca local.

Caso o limite de tempo não seja atingido, o processo continua.



Análise dos resultados

- O parâmetro avaliado é sempre:

Menor = Melhor

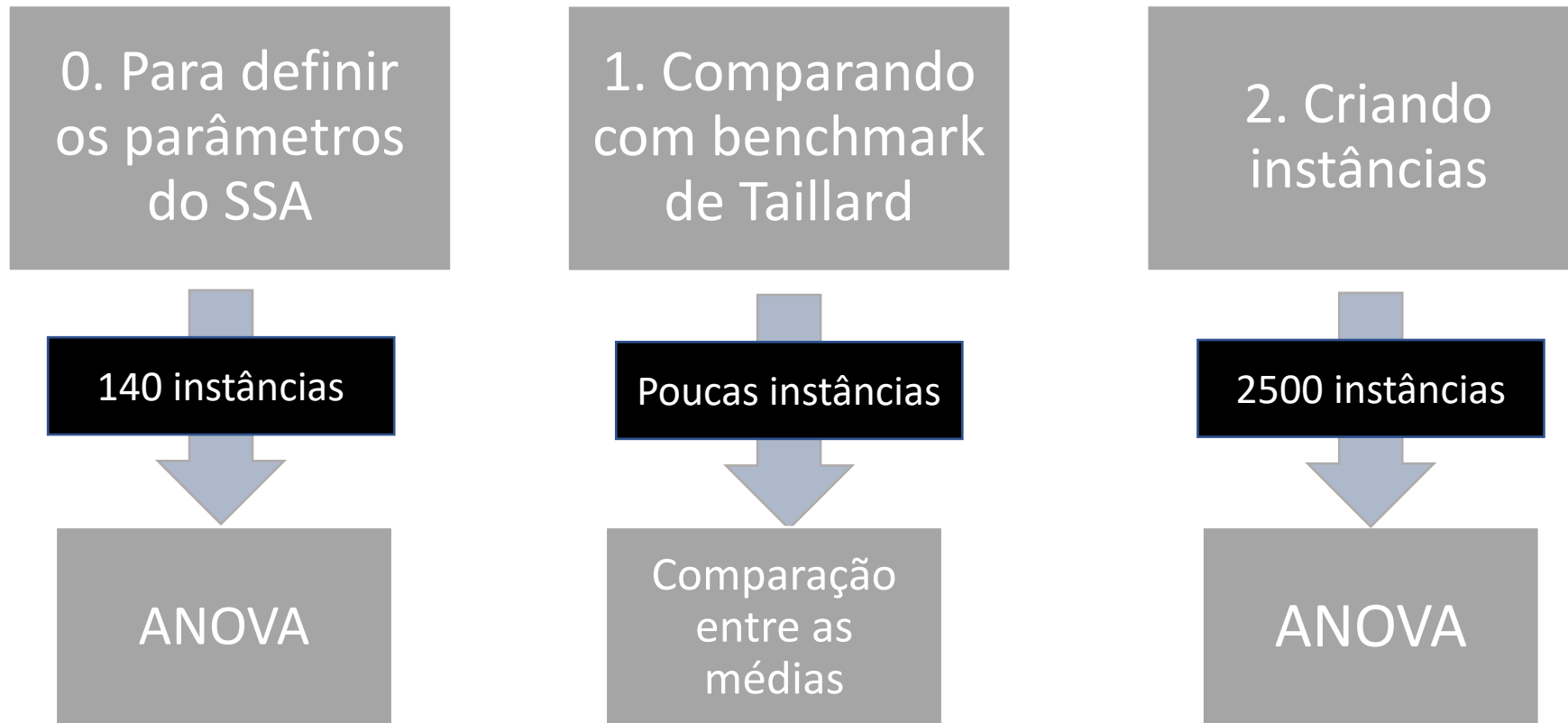
$$I_{hs} = \frac{Heur_{hs} - Best_s}{Best_s} \times 100$$

Melhor valor conhecido
para a instância

- Heurísticas com natureza estocástica foram resolvidas 3 vezes e foi adotado o valor médio.
- Foi definido como limite de computação o tempo de **n . m . 50ms**

Análise dos resultados – visão geral

Foram desenvolvidas três análises:



Análise dos resultados – Parte 0 - Introdução

- A parte 0 é a definição dos parâmetros do SSA:
 - α , 2 levels: 0.5, 0.75;
 - β , 2 levels: $n \cdot (n-1)/2$, $n \cdot (n-1)$;
 - d , 2 levels: 4, 6.
- Foram geradas 5 instâncias para cada combinação entre serviços e máquinas:

$n = \{20, 50, 80, 110, 140, 170, 200\}$ and $m = \{5, 10, 15, 20\}$.

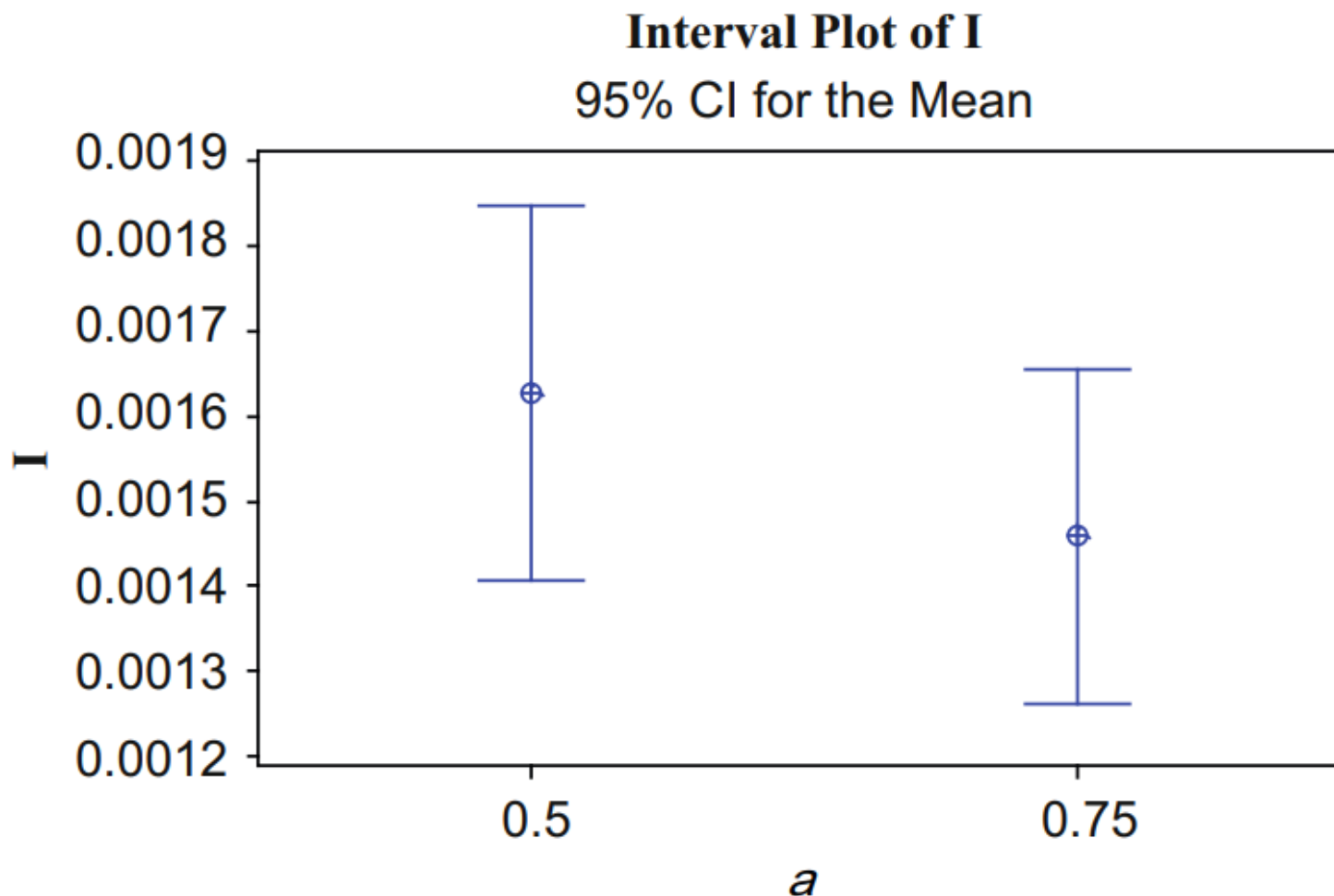
Análise dos resultados – Parte 0 - Resultados

Table 2

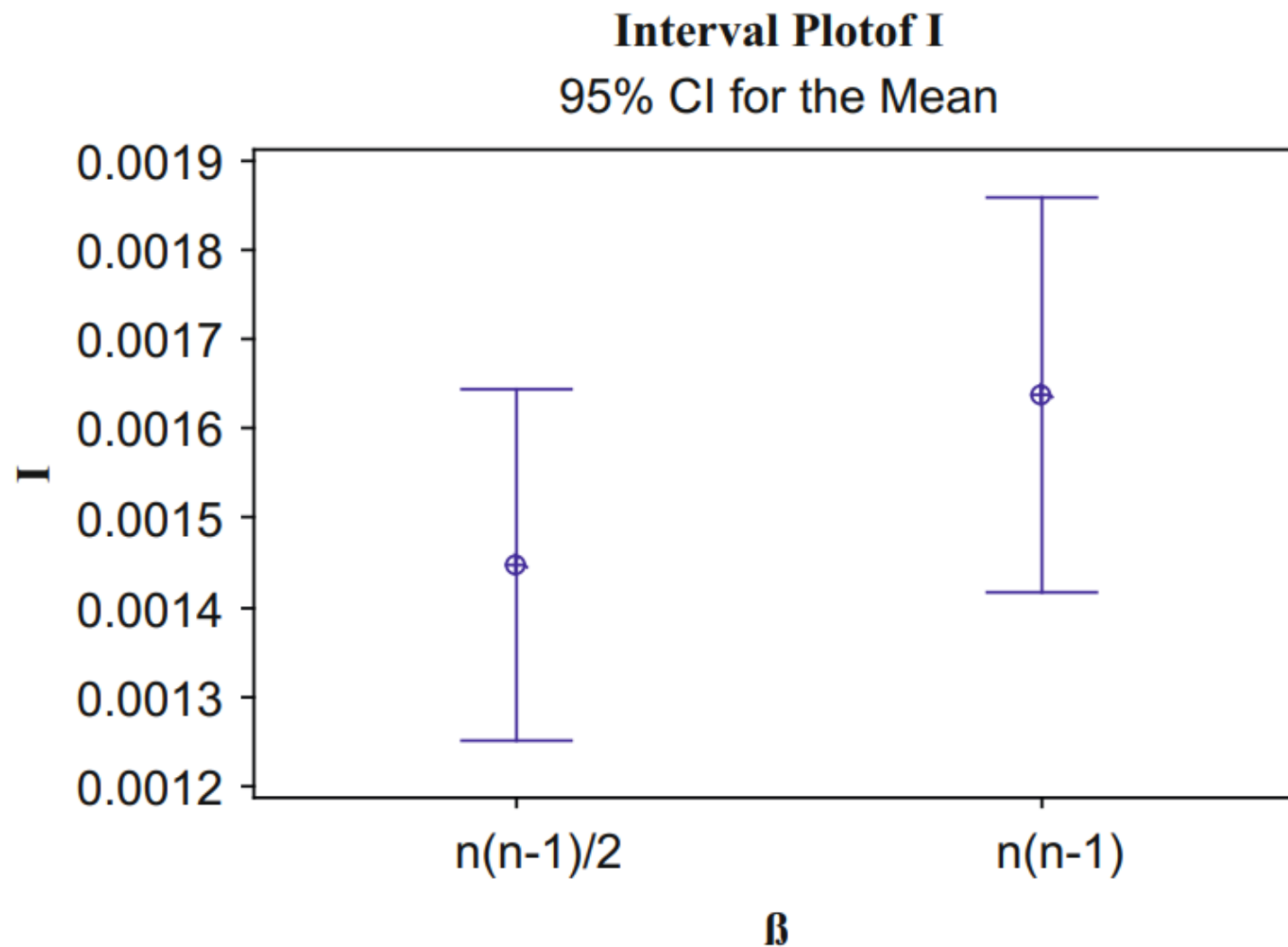
ANOVA table for the experiment on tuning the parameters of SSA.

Source	DF	SS	MS	F	P
Main effects					
n	6	0.000417868	0.000069645	15.74	0.000
m	3	0.001477890	0.000492630	111.33	0.000
α	1	0.000008049	0.000008049	1.82	0.178
β	1	0.000010056	0.000010056	2.27	0.132
d	1	0.000009788	0.000009788	2.21	0.137
Interactions					
$n*m$	18	0.000397221	0.000022068	4.99	0.000
$n*\alpha$	6	0.000033291	0.000005549	1.25	0.276
$n*\beta$	6	0.000026449	0.000004408	1.00	0.426
$n*d$	6	0.000014854	0.000002476	0.56	0.763
$m*\alpha$	3	0.000013737	0.000004579	1.03	0.376
$m*\beta$	3	0.000007268	0.000002423	0.55	0.650
$m*d$	3	0.000007393	0.000002464	0.56	0.644
$\alpha*\beta$	1	0.000000000	0.000000000	0.00	0.997
$\alpha*d$	1	0.000001581	0.000001581	0.36	0.550
$\beta*d$	1	0.000000004	0.000000004	0.00	0.977
Error	1059	0.004686137	0.000004425		
Total	1119	0.007111585			

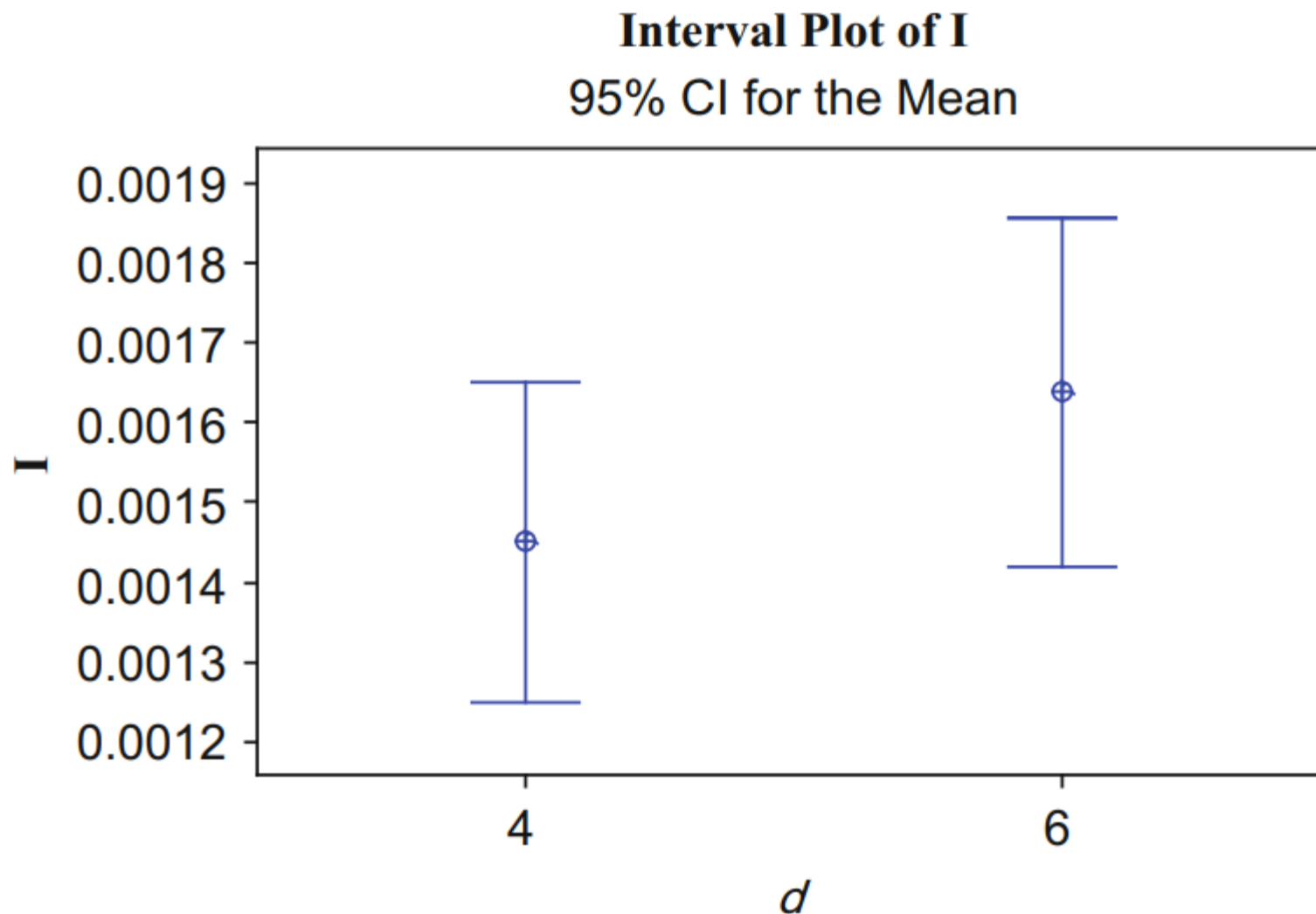
Análise dos resultados – Parte 0 - Resultados



Análise dos resultados – Parte 0 - Resultados



Análise dos resultados – Parte 0 - Resultados



Análise dos resultados – Parte 0 - Conclusão

- Os intervalos de confiança à 95% não foram significativamente diferentes, mas para as outras etapas **foram adotados os parâmetros que produziram a menor média.**

Análise dos resultados – Parte 1 - Introdução

- Os resultados são comparados contra os resultados produzidos no artigo de benchmark de Taillard.
- São considerados diversas outras heurísticas da literatura.

Análise dos resultados – Parte 1 - Resultados

“d-” significa que foi o resultado obtido aplicando apenas a formulação direta. Sem o “d-” aplicando ambas e escolhendo a melhor.

Apenas variações dos critérios de desempate do NEH.

Table 3
Average of index I_{hs} on Taillard's instances.

	d-NEH0	NEH0	d-NEH1	NEH1	d-NEH2	NEH2	d-NEH3	NEH3	d-NEHR	NEHR
20 × 5	3.30	2.56	2.69	2.65	2.73	2.57	2.48	2.48	2.52	2.33
20 × 10	4.60	4.26	4.35	4.35	4.31	4.31	4.13	4.13	4.32	3.87
20 × 20	3.73	3.36	3.68	3.68	3.41	3.38	3.70	3.70	3.54	3.29
50 × 5	0.73	0.58	0.87	0.74	0.59	0.59	0.73	0.60	0.60	0.47
50 × 10	5.07	4.90	5.08	5.03	4.87	4.60	4.80	4.68	4.83	4.35
50 × 20	6.66	6.01	6.51	6.33	6.42	6.18	6.25	5.96	5.77	5.56
100 × 5	0.53	0.40	0.48	0.41	0.40	0.40	0.49	0.41	0.35	0.34
100 × 10	2.21	1.97	2.10	1.92	1.77	1.64	1.96	1.57	2.08	1.68
100 × 20	5.34	5.19	5.28	5.05	5.28	5.05	5.01	4.67	5.44	5.09
200 × 10	1.26	1.14	1.19	1.08	1.17	1.10	1.01	0.90	1.02	0.94
200 × 20	4.41	4.31	4.41	4.21	4.23	4.03	3.88	3.69	4.17	4.04
500 × 20	2.06	2.01	1.98	1.93	2.03	1.88	1.71	1.62	1.96	1.81
Average	3.33	3.06	3.22	3.11	3.10	2.98	3.01	2.87	3.05	2.81

Resultados da versão com critério de desempate proposto

Análise dos resultados – Parte 1 - Resultados

Table 4

Average of index I_{hs} obtained by the heuristics on Taillard's instances.

	d- KKER	KKER	d- NEHR	NEHR	d- NMER	NMER	d- RAER	RAER
20 × 5	2.46	2.32	2.52	2.33	2.71	2.37	3.12	2.57
20 × 10	4.97	4.11	4.32	3.87	4.13	3.15	5.94	4.18
20 × 20	3.47	3.25	3.54	3.29	3.98	3.42	4.57	3.61
50 × 5	0.74	0.49	0.60	0.47	0.88	0.71	1.67	1.22
50 × 10	5.08	4.35	4.83	4.35	5.03	4.63	5.59	5.59
50 × 20	5.98	5.69	5.77	5.56	5.93	5.68	6.68	6.14
100 × 5	0.36	0.35	0.35	0.34	0.47	0.32	0.92	0.55
100 × 10	1.82	1.78	2.08	1.68	2.14	1.73	2.65	2.24
100 × 20	5.37	4.70	5.44	5.09	5.41	5.00	5.71	5.32
200 × 10	1.11	0.97	1.02	0.94	1.07	0.91	1.69	1.56
200 × 20	4.18	3.93	4.17	4.04	3.99	3.86	4.57	4.25
500 × 20	1.87	1.77	1.96	1.81	1.92	1.82	2.41	2.21
Average	3.12	2.81	3.05	2.81	3.14	2.80	3.79	3.29

A diferença das
quatro variações da
primeira etapa

RAER foi o pior,
usando sequências
iniciais aleatórias.

Análise dos resultados – Parte 1 - Resultados

Table 5

Average of index I_{hs} obtained by the heuristics on Taillard's benchmark.

	IGA	KKER+	NEHR+	NMER+	RAER+
20 × 5	0.073	0.041	0.059	0.041	0.041
20 × 10	0.030	0.119	0.119	0.079	0.120
20 × 20	0.067	0.102	0.174	0.094	0.167
50 × 5	0.016	0.025	0.048	0.031	0.035
50 × 10	0.755	1.148	0.938	1.152	0.969
50 × 20	1.367	2.428	2.178	2.376	2.006
100 × 5	0.064	0.041	0.052	0.040	0.051
100 × 10	0.417	0.461	0.470	0.446	0.473
100 × 20	1.802	2.279	2.087	2.343	2.117
200 × 10	0.462	0.351	0.408	0.351	0.437
200 × 20	2.222	1.913	1.983	1.912	1.935
500 × 20	1.273	1.115	1.114	1.020	1.192
Average	0.712	0.835	0.802	0.823	0.795

Agora sim com as três etapas.

IGA foi o algoritmo mais eficiente que encontraram para comparar.

Análise dos resultados – Parte 2 - Introdução

- As heurísticas são comparadas em problemas gerados.
- **O objetivo das instâncias pequenas são comparadas contra valores obtidos por um método exato.**
- 2500 instâncias.
- Utiliza ANOVA;
- **Busca entender se há relação entre a qualidade das respostas de uma heurística contra outra, e de uma heurística e parâmetros do problema** (ex.: número de máquinas ou tarefas).

Análise dos resultados – Parte 2 - Introdução

- Para ANOVA ser aplicável, algumas premissas precisam ser verdadeiras:
- Normalidade
- Homocedasticidade
- Independência dos resíduos

Análise dos resultados – Parte 2 - Introdução

- Para ANOVA ser aplicável, algumas premissas precisam ser verdadeiras:

Essa premissa não se aplica, mas ela não é crítica quando se tem muitos dados, o que é o caso nessa etapa.

- ~~• Normalidade~~
- Homocedasticidade
- Independência dos resíduos

Análise dos resultados – Parte 2 - Resultados

Table 6

Three way ANOVA table for *Algorithm*, *n* and *m*.

Source	DF	SS	MS	F	P
<i>Algorithm</i>	4	13.7092	3.4273	29.51	0.000
<i>n</i>	3	5.4989	1.8330	15.78	0.000
<i>m</i>	3	171.5620	57.1873	492.48	0.000
<i>Algorithm</i> * <i>n</i>	12	0.6144	0.0512	0.44	0.947
<i>Algorithm</i> * <i>m</i>	12	12.9249	1.0771	9.28	0.000
<i>n</i> * <i>m</i>	9	9.3521	1.0391	8.95	0.000
Error	7956	923.8591	0.1161		
Total	7999	1137.5206			

Análise dos resultados – Parte 2 – Instâncias pequenas

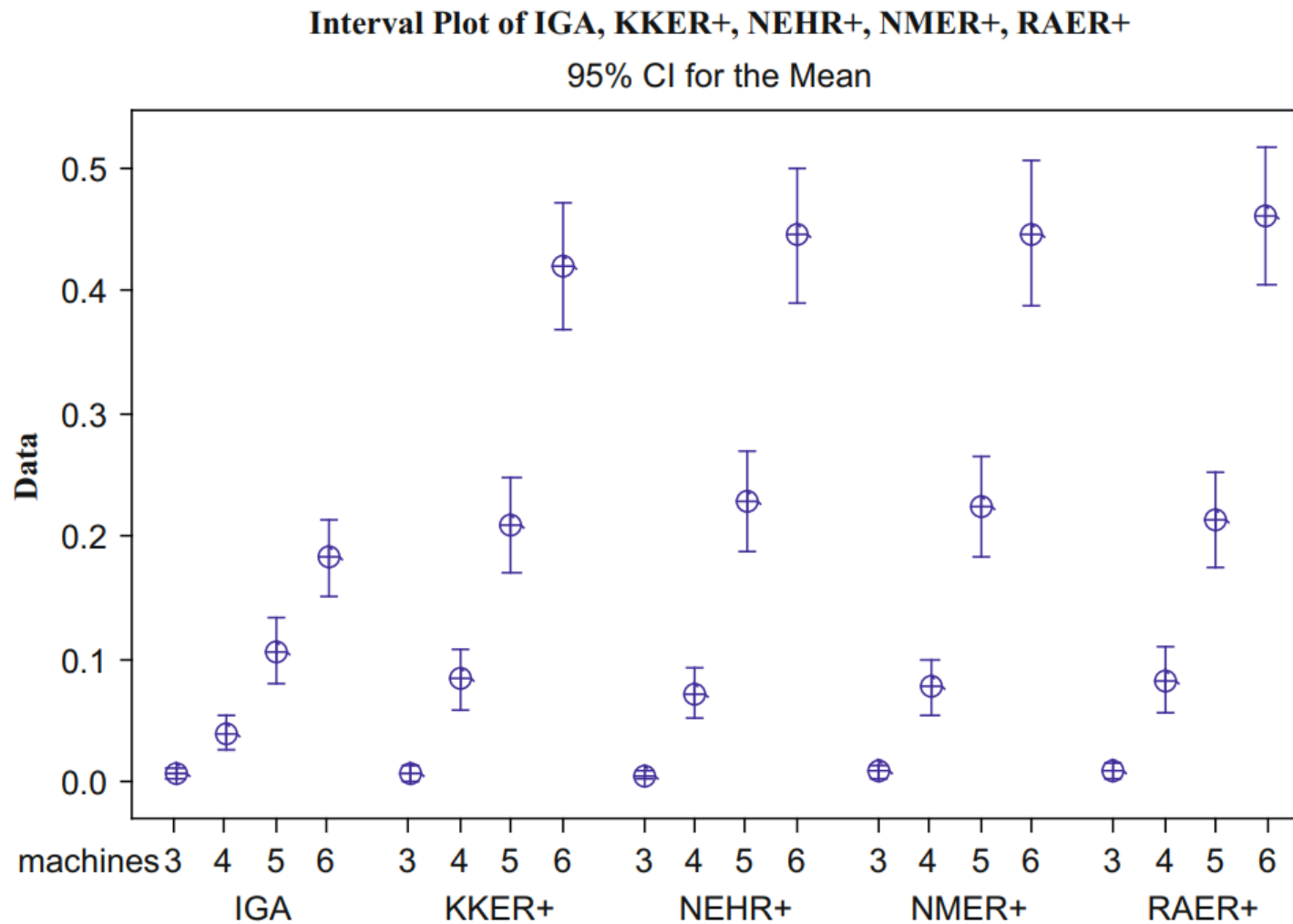


Fig. 9. Interval plot for the average index I_{hs} for each procedure stratified by the number of machines.

Análise dos resultados – Parte 2 - Resultados

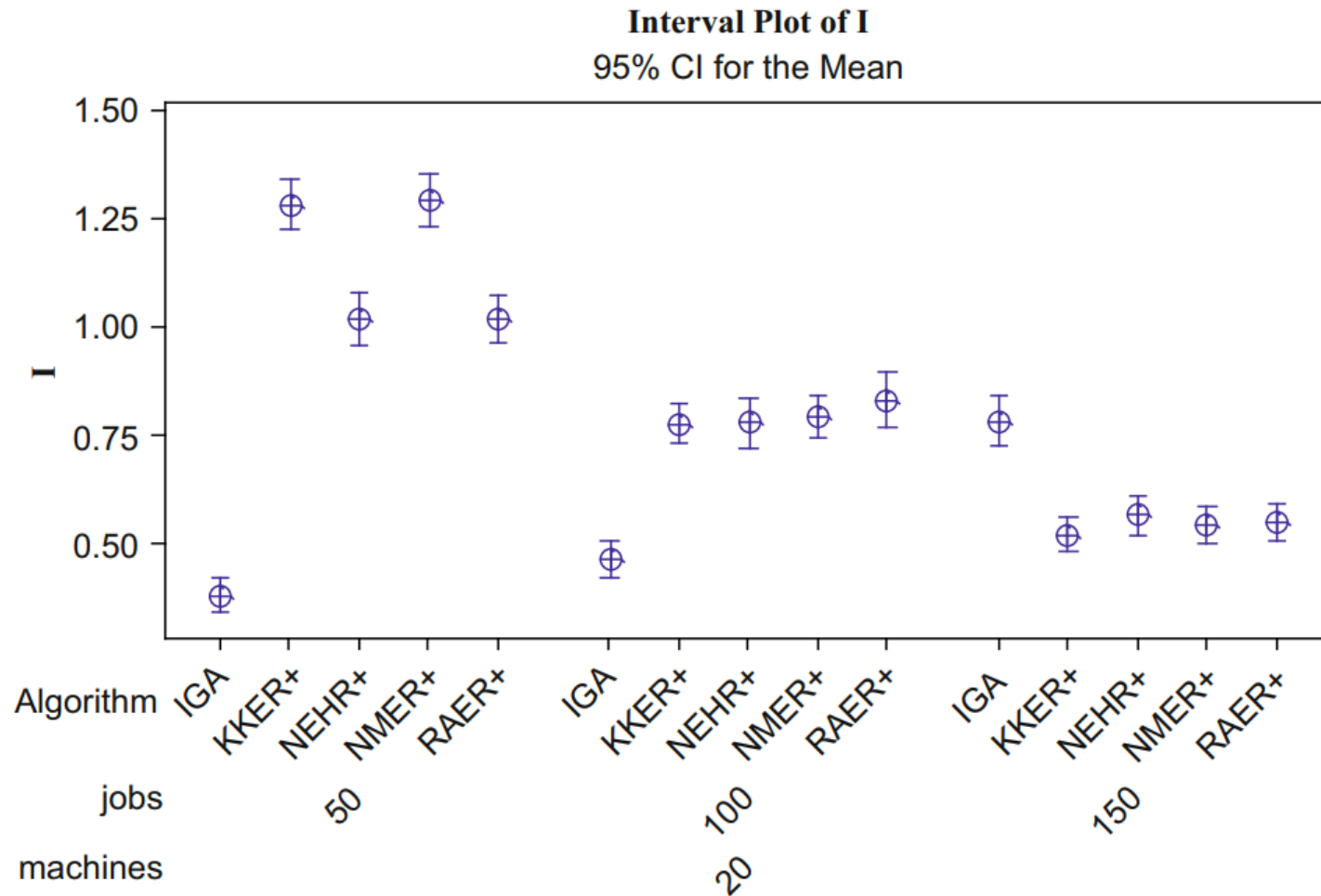


Fig. 10. Interval plot of the average of index I_{hs} for procedure stratified by jobs when $m=20$.

Análise dos resultados – Parte 2 - Resultados

Table 7

Average of index I_{hs} for each algorithm stratified by n and m .

Machines	Jobs	IGA	KKER+	NEHR+	NMER+	RAER+
5	50	0.0052	0.0076	0.0084	0.0073	0.0090
	100	0.2432	0.3921	0.3202	0.3911	0.3186
	150	0.3823	1.2855	1.0190	1.2949	1.0199
10	50	0.0037	0.0017	0.0023	0.0015	0.0040
	100	0.1531	0.1300	0.14869	0.1362	0.1700
	150	0.4644	0.7788	0.7811	0.7948	0.8342
20	50	0.0049	0.0038	0.0059	0.0039	0.0060
	100	0.1225	0.0844	0.0948	0.0758	0.1087
	150	0.7851	0.5221	0.5661	0.5435	0.5503

Conclusão

- **A heurística deste problema permite efetuar buscas em espaços diferentes mesmo com soluções iniciais idênticas (Vetor revolver)**
- Também foi apresentada uma heurística que permite continuar a busca no espaço de solução através de soluções com objetivos empatados, como estratégia para fugir de ótimos locais. (Soft-simulated annealing)

Conclusão

- Usar uma reformulação pode trazer resultados significantes, mesmo que a reformulação não pareça mudar muito o problema; **Prestar atenção ao potencial das propriedades do problema.**
- Podemos utilizar métodos estatísticos para avaliar diferentes variações das heurísticas e relação entre heurísticas e parâmetros; **Alto potencial de aplicação em outros trabalhos.**
- Podemos melhorar heurísticas através de critérios de desempate dos indicadores utilizados na heurística;