

Projeto de métodos de busca local para problemas de *scheduling* em ambientes de máquinas não relacionadas

Aluno: Rodney O. M. Diana

Orientador: Prof. Dr. Sérgio Ricardo de Souza (CEFET-MG)

Coorientadora: Profa. Dra. Elizabeth Fialho Wanner (CEFET-MG)

Sumário

1 Introdução

- *Introdução: Scheduling*
- *Máquinas paralelas Não Relacionadas*
- *Minimização da soma ponderada dos atrasos*

2 Estudo proposto

3 Metodologia e Resultados

4 Considerações finais

Introdução: *Scheduling*

Segundo o *Cambridge Dictionary*¹ a palavra *scheduling*, significa:

“A atividade de planejar as datas em que determinadas tarefas serão realizadas ou eventos irão ocorrer.”

Na pesquisa operacional, o termo *scheduling* normalmente é associado a problemas destinados a sequenciar/atribuir, na dimensão temporal, um conjunto de “tarefas”, de modo a otimizar um determinado objetivo estratégico.



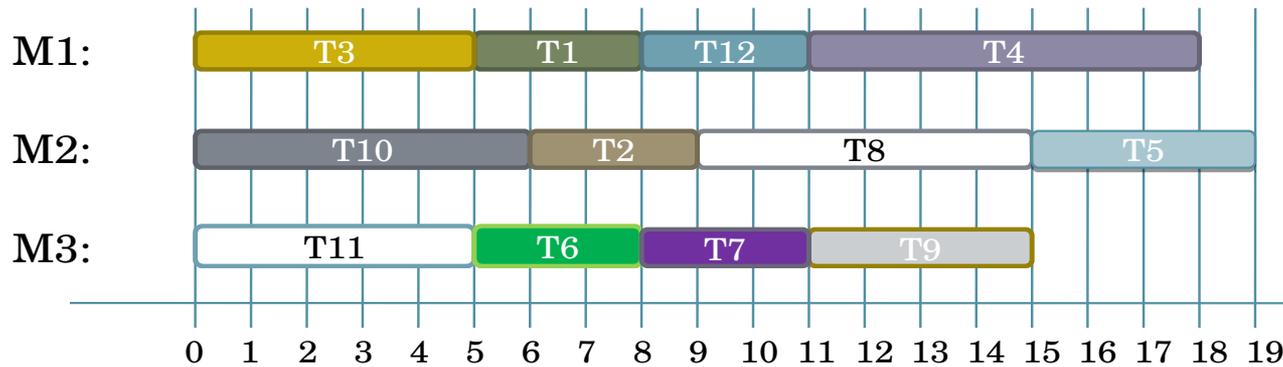
¹<https://dictionary.cambridge.org/pt/dicionario/ingles/scheduling> acessado em 08/06/2021

Introdução: *Scheduling*

- Em sua maioria problemas de natureza NP-Difícil.
- Necessitam de soluções computacionais específicas para contextos específicos.
- Em diversos cenários métodos exatos costumam apresentar elevado custo computacional.
- Soluções computacionais normalmente baseadas em métodos de inteligência computacional.

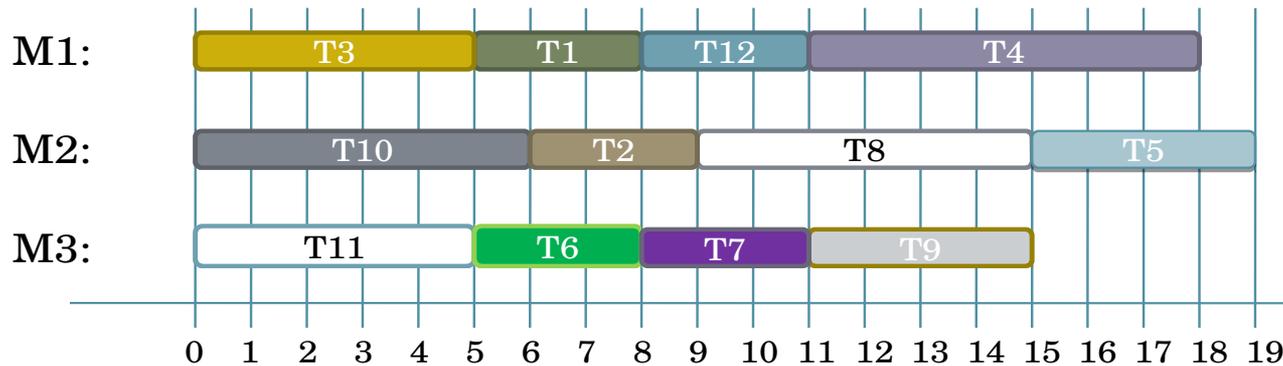
Máquinas paralelas Não Relacionadas

Problema de sequenciamento em máquinas paralelas



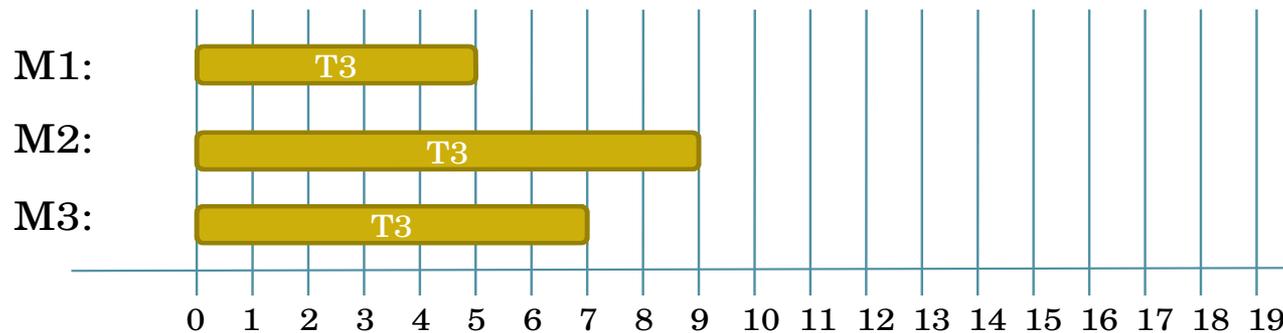
Máquinas paralelas Não Relacionadas

Problema de sequenciamento em máquinas paralelas



■ p_{ik} é o tempo de processamento da tarefa k , sequenciado na máquina i .

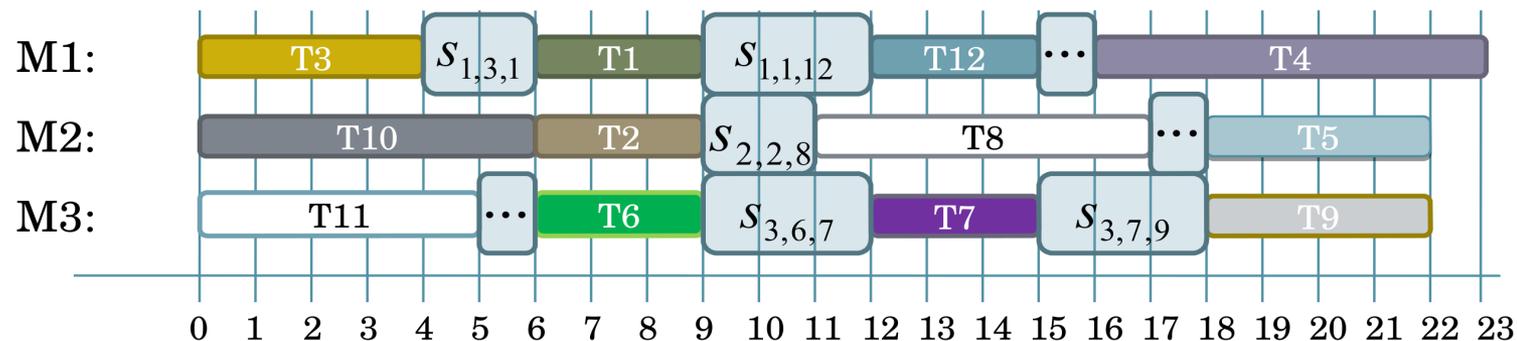
Máquinas paralelas não relacionadas



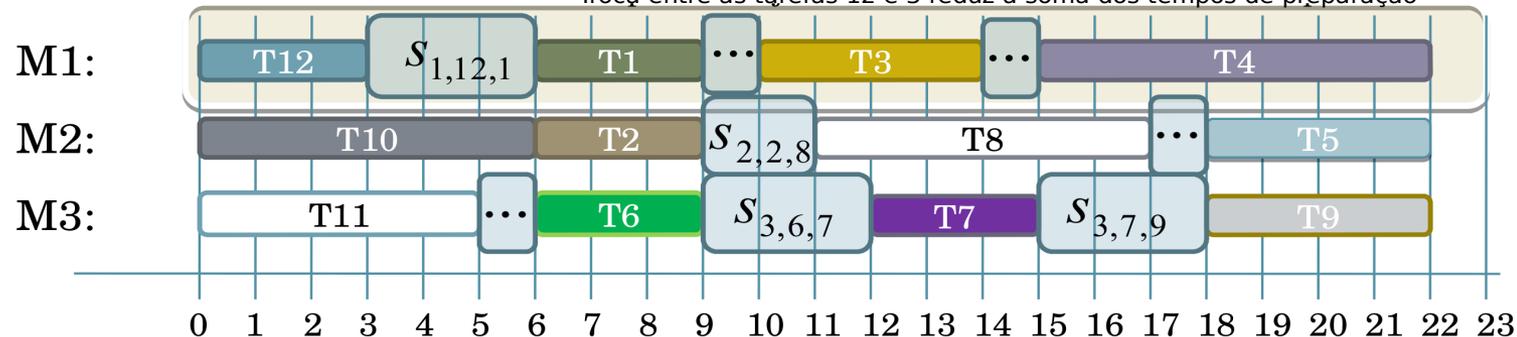
Tempos de preparação - Máquinas paralelas não relacionadas

- Ambientes conhecidos como UPMSMDST (*Unrelated Parallel Machine with Sequence and Machine Dependent Setup Times*).
- s_{ijk} é o tempo de preparação da tarefa k , necessário para ser processada na máquina i , quando é precedida pela tarefa j .

Tempos de preparação

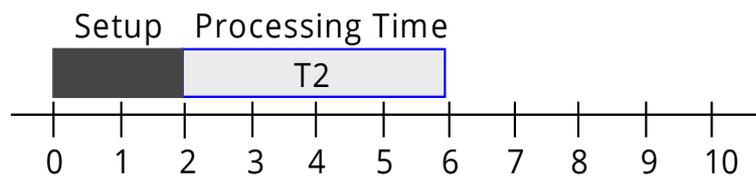


Troça entre as tarefas 12 e 3 reduz a soma dos tempos de preparação



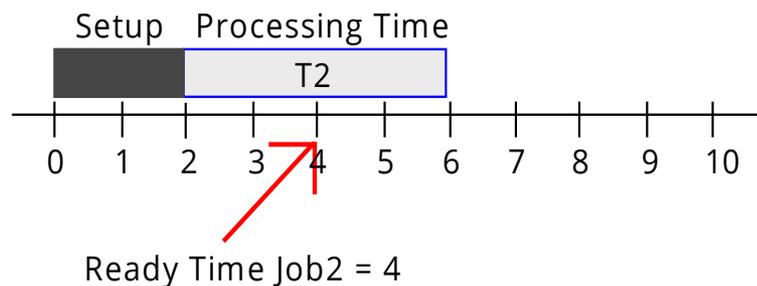
Instante de término e *Job Ready times*

- Dado uma máquina $i = 1$ e uma tarefa $k = 2$ na primeira posição do sequenciamento.
- Tempo de processamento (p_{ik}): $p_{1,2} = 4$
- Tempo de preparação (s_{ijk}): $s_{1,2,0} = 2$
- Instante de término (C_k): $C_2 = 6$



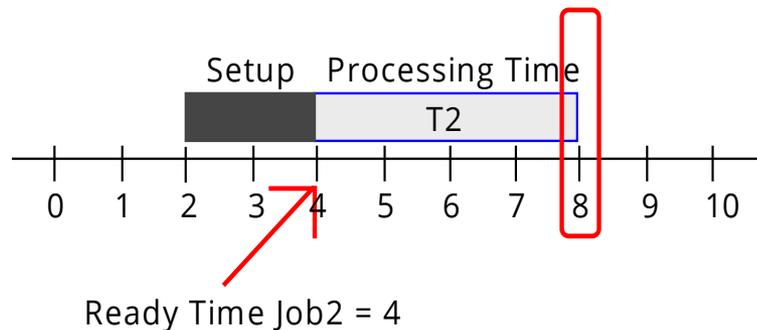
Instante de término e *Job Ready times*

- Dado uma máquina $i = 1$ e uma tarefa $k = 2$ na primeira posição do sequenciamento.
- Tempo de processamento (p_{ik}): $p_{1,2} = 4$
- Tempo de preparação (s_{ijk}): $s_{1,2,0} = 2$
- Instante de término (C_k): $C_2 = 6$
- *Job Ready Time* (r_k): $r_2 = 4$



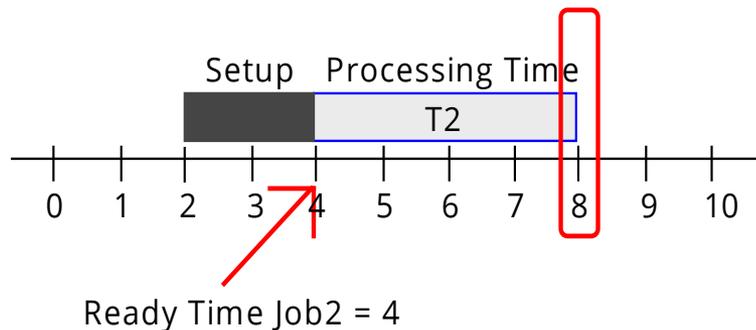
Instante de término e *Job Ready times*

- Dado uma máquina $i = 1$ e uma tarefa $k = 2$ na primeira posição do sequenciamento
- Tempo de processamento (p_{ik}): $p_{1,2} = 4$
- Tempo de preparação (s_{ijk}): $s_{1,2,0} = 2$
- Instante de término (C_k): $C_2 = 8$
- *Job Ready Time* (r_k): $r_2 = 4$



Instante de término e *Job Ready times*

- Dado uma máquina $i = 1$ e uma tarefa $k = 2$ na primeira posição do sequenciamento
- Tempo de processamento (p_{ik}): $p_{1,2} = 4$
- Tempo de preparação (s_{ijk}): $s_{1,2,0} = 2$
- Instante de término (C_k): $C_2 = 8$
- *Job Ready Time* (r_k): $r_2 = 4$
- $C_k = \max(r_k, C_{ij} + s_{ijk}) + p_{ik}$

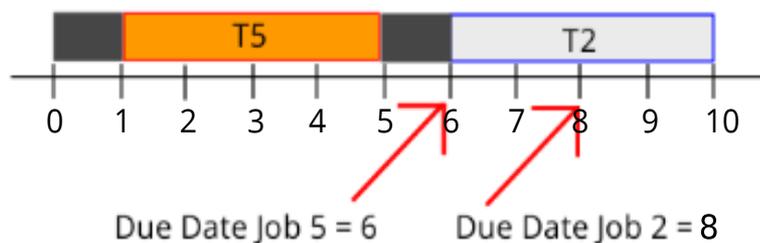


Minimização da soma ponderada dos atrasos

- A minimização da soma ponderada dos atrasos é um dos objetivos mais importantes estudados em ambientes UPMSMDST.

Minimização da soma ponderada dos atrasos

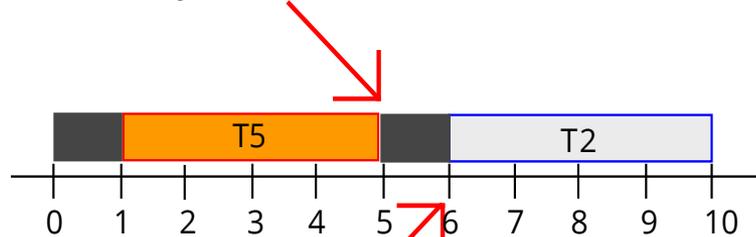
- A minimização da soma ponderada dos atrasos é um dos objetivos mais importantes estudados em ambientes UPMSMDST.
 - Dado uma máquina $i = 1$ e uma tarefa $k = 2$ atribuída após a tarefa $j = 5$;
 - Tempos de processamento (p_{ik}): $p_{1,2} = 4$, $p_{1,5} = 4$;
 - Tempos de preparação (s_{ijk}): $s_{1,2,5} = 1$, $s_{1,5,0} = 1$;
 - Instantes de término (C_k): $C_2 = 10$, $C_5 = 5$;
 - *Job Ready Times* (r_k): $r_2 = 4$, $r_5 = 0$.
 - Datas de entrega (d_k): $d_2 = 8$, $d_5 = 5$.



Minimização da soma ponderada dos atrasos

- A minimização da soma ponderada dos atrasos é um dos objetivos mais importantes estudados em ambientes UPMSMDST.
 - Dado uma máquina $i = 1$ e uma tarefa $k = 2$ atribuída após a tarefa $j = 5$;
 - Tempos de processamento (p_{ik}): $p_{1,2} = 4$, $p_{1,5} = 4$;
 - Tempos de preparação (s_{ijk}): $s_{1,2,5} = 1$, $s_{1,5,0} = 1$;
 - Instantes de término (C_k): $C_2 = 10$, $C_5 = 5$;
 - *Job Ready Times* (r_k): $r_2 = 4$, $r_5 = 0$.
 - Datas de entrega (d_k): $d_2 = 8$, $d_5 = 5$.

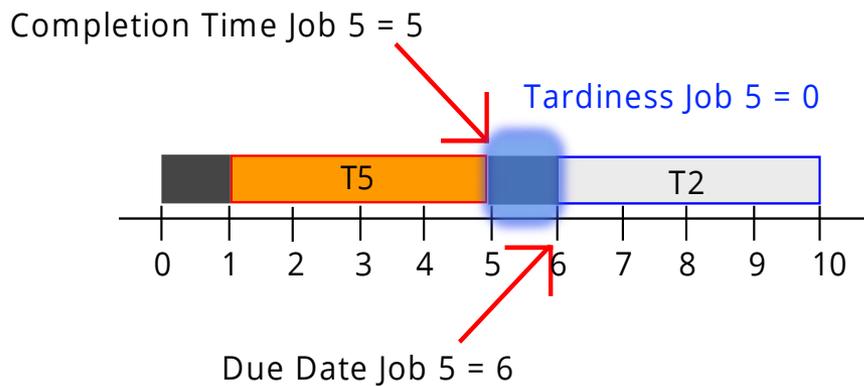
Completion Time Job 5 = 5



Due Date Job 5 = 6

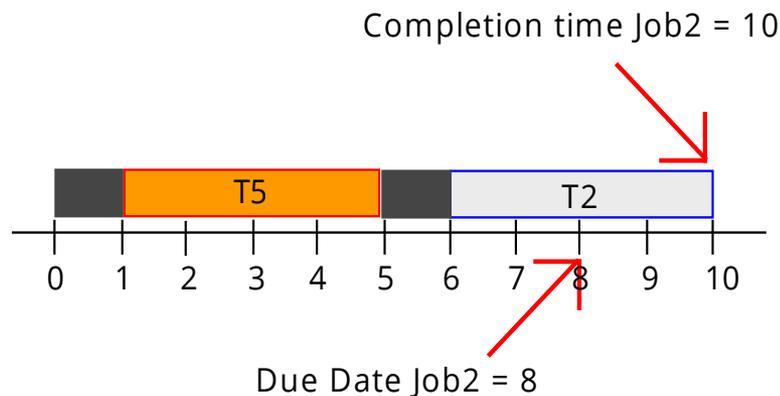
Minimização da soma ponderada dos atrasos

- A minimização da soma ponderada dos atrasos é um dos objetivos mais importantes estudados em ambientes UPMSMDST.
 - Dado uma máquina $i = 1$ e uma tarefa $k = 2$ atribuída após a tarefa $j = 5$;
 - Tempos de processamento (p_{ik}): $p_{1,2} = 4, p_{1,5} = 4$;
 - Tempos de preparação (s_{ijk}): $s_{1,2,5} = 1, s_{1,5,0} = 1$;
 - Instantes de término (C_k): $C_2 = 10, C_5 = 5$;
 - *Job Ready Times* (r_k): $r_2 = 4, r_5 = 0$.
 - Datas de entrega (d_k): $d_2 = 8, d_5 = 5$.



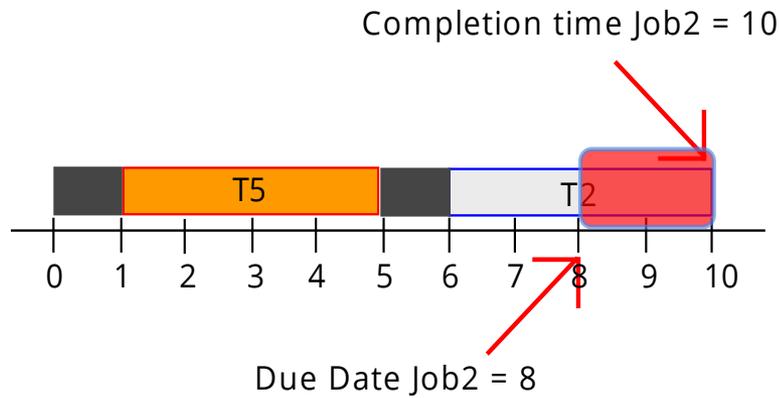
Minimização da soma ponderada dos atrasos

- A minimização da soma ponderada dos atrasos é um dos objetivos mais importantes estudados em ambientes UPMSMDST.
 - Dado uma máquina $i = 1$ e uma tarefa $k = 2$ atribuída após a tarefa $j = 5$;
 - Tempos de processamento (p_{ik}): $p_{1,2} = 4$, $p_{1,5} = 4$;
 - Tempos de preparação (s_{ijk}): $s_{1,2,5} = 1$, $s_{1,5,0} = 1$;
 - Instantes de término (C_k): $C_2 = 10$, $C_5 = 5$;
 - *Job Ready Times* (r_k): $r_2 = 4$, $r_5 = 0$.
 - Datas de entrega (d_k): $d_2 = 8$, $d_5 = 5$.



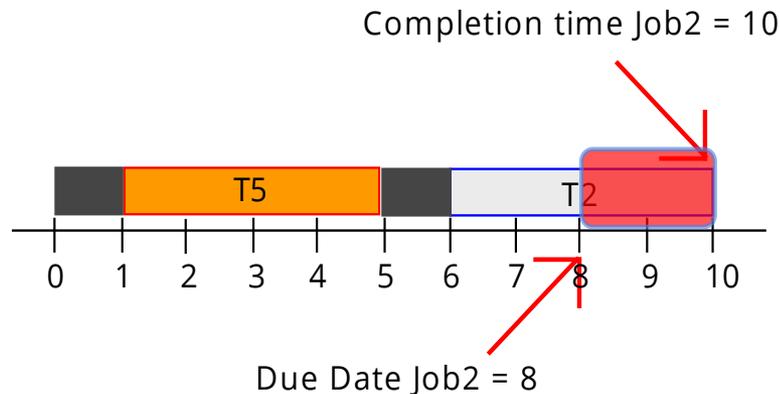
Minimização da soma ponderada dos atrasos

- A minimização da soma ponderada dos atrasos é um dos objetivos mais importantes estudados em ambientes UPMSMDST.
 - Dado uma máquina $i = 1$ e uma tarefa $k = 2$ atribuída após a tarefa $j = 5$;
 - Tempos de processamento (p_{ik}): $p_{1,2} = 4, p_{1,5} = 4$;
 - Tempos de preparação (s_{ijk}): $s_{1,2,5} = 1, s_{1,5,0} = 1$;
 - Instantes de término (C_k): $C_2 = 10, C_5 = 5$;
 - *Job Ready Times* (r_k): $r_2 = 4, r_5 = 0$.
 - Datas de entrega (d_k): $d_2 = 8, d_5 = 5$.



Minimização da soma ponderada dos atrasos

- A minimização da soma ponderada dos atrasos é um dos objetivos mais importantes estudados em ambientes UPMSMDST.
 - Dado uma máquina $i = 1$ e uma tarefa $k = 2$ atribuída após a tarefa $j = 5$;
 - Tempos de processamento (p_{ik}): $p_{1,2} = 4, p_{1,5} = 4$;
 - Tempos de preparação (s_{ijk}): $s_{1,2,5} = 1, s_{1,5,0} = 1$;
 - Instantes de término (C_k): $C_2 = 10, C_5 = 5$;
 - Job Ready Times (r_k): $r_2 = 4, r_5 = 0$.
 - Datas de entrega (d_k): $d_2 = 8, d_5 = 5$.
 - Atraso da tarefa k is: $T_k = \max(C_k - d_k, 0)$.



Minimização da soma ponderada dos atrasos

- O critério de otimização de minimização da soma ponderada dos atrasos é definido com:

$$TWT = \sum_{k=1}^N w_k T_k \quad (1)$$

sendo:

T_k : unidades de atraso da tarefa k .

w_k : custo (peso) por unidade de atraso da tarefa k .

N número de tarefas presentes no sequenciamento.

Projeto de métodos de busca local

- Cenário encontrado literatura de problemas em ambientes UPMSMDST:
 - **Muita importância** dada na análise/projeto de operadores de otimização global (diversificação).
 - **Poucas informações** a respeito da eficiência de componentes usados na construção de operadores de busca local (intensificação).
- Qual a importância dos operadores de busca local na convergência das metaheurísticas?
- Quais tipos de movimentos tendem a gerar estruturas de vizinhança mais eficientes?
- Quais as formas mais eficientes de explorar estruturas de vizinhança?
- A ordem de exploração das estruturas costuma fazer diferença na convergência?

Projeto de métodos de busca local

- Neste estudo propomos uma metodologia para o projeto de operadores de busca local de metaheurística destinadas a solução de problemas de sequenciamento em ambientes UPMSMDST.
- Metodologia utiliza o *Variable Neighborhood Descent* (VND) como operador de busca local.
- Metodologia considera três hipóteses:
 - **Hipótese i:** A ordem, o método (variações VND) e a forma que as estruturas de vizinhança são exploradas podem incrementar/decrementar o desempenho do operador de busca local baseado em VND;
 - **Hipótese ii:** As metaheurísticas têm diferentes demandas do operador de busca local. Assim, a importância de cada estrutura de vizinhança varia de acordo com a metaheurística avaliada;
 - **Hipótese iii:** O desempenho de uma metaheurística, já consolidada, pode ser incrementado apenas com a substituição do operador de busca local.

Projetando operadores de busca local

- Operadores projetados usando variações do *Variable Neighborhood Descent* (VND) [4].
- Avaliamos três variações do VND:
 - *Basic-VND*;
 - *Randon-VND*;
 - *Union-VND*.
- Estudo realizado sobre seis estruturas de vizinhança:
 - Inserção Interna - NE.II;
 - Troca Interna - NE.IS;
 - Inserção Externa - NE.EI;
 - Troca Externa - NE.ES;
 - Inserção *Iterated Greedy Search* - NE.IGS;
 - Troca *Iterated Greedy Search* - NE.SIGS.
- Duas formas de exploração das estruturas de vizinhança:
 - *Best Improvement*;
 - *First Improvement*.
- Duas ordem de exploração das estruturas de vizinhança:
 - *Cardinalidade de permutações*;
 - *Impacto esperado na solução*.

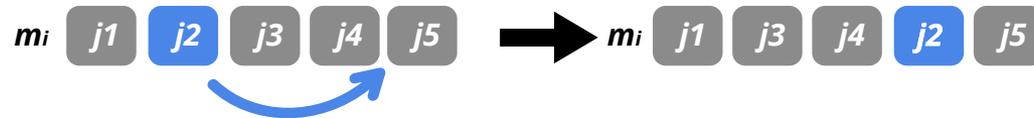
Projetando operadores de busca local

- No VND as estruturas de vizinhança são exploradas de forma cíclica, sistêmica e determinística.

```
1: function VND( $s, N_s, kmax$ )
2:    $s' \leftarrow NULL$ ;
3:    $s^* \leftarrow s$ ;
4:    $k \leftarrow 1$ ;
5:    $non\_improvement \leftarrow 0$ ;
6:   while ( $non\_improvement < kmax$ ) do
7:      $s' \leftarrow \operatorname{argmin}_{\forall x \in N_k(s^*)} f(x)$ ;
8:      $k \leftarrow \operatorname{change\_neighborhood}(k, s', s^*)$ ;
9:     if  $f(s') < f(s^*)$  then
10:        $s^* \leftarrow s'$ ;
11:        $non\_improvement \leftarrow 0$ ;
12:     else
13:        $non\_improvement \leftarrow non\_improvement + 1$ ;
14:     end if
15:   end while
16: return  $s^*$ ;
17: end function
```

Estruturas de Vizinhança: Inserção Interna - NE.II

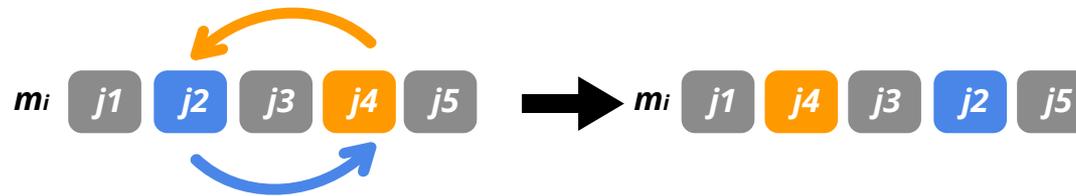
- Uma tarefa k , atribuída a uma máquina i , é inserida na posição p da mesma máquina i .



- Seleciona aleatoriamente uma máquina i , onde $\sum_{k \in N_i} \alpha_k T_k > 0$.
- Aplica *Best Improvement* ou *First Improvement* na máquina selecionada.

Estruturas de Vizinhança: Troca Interna - NE.IS

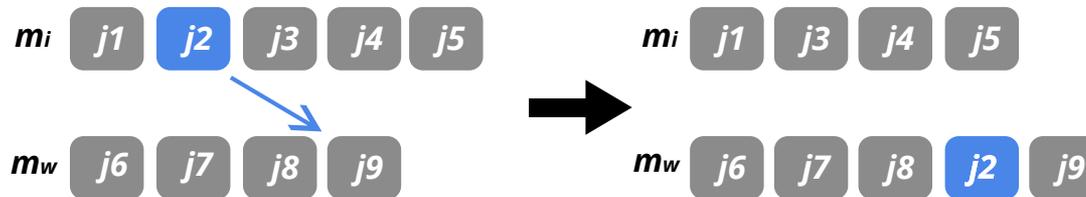
- Duas tarefas, k e j , $k \neq j$, atribuídas a mesma máquina i , tem suas posições trocadas.



- Seleciona aleatoriamente uma máquina i , onde $\sum_{k \in N_i} \alpha_k T_k > 0$.
- Aplica *Best Improvement* ou *First Improvement* na máquina selecionada.

Estruturas de Vizinhança: Inserção Externa - NE.EI

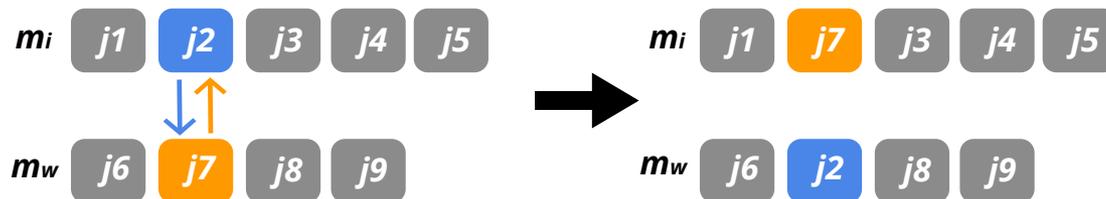
- Uma tarefa k , atribuída a máquina i , é redirecionada para uma máquina w , $i \neq w$.



- Seleciona aleatoriamente uma máquina i , onde $\sum_{k \in N_i} \alpha_k T_k > 0$.
- Passo2: Seleciona aleatoriamente uma máquina, sendo $w \neq i$.
- Aplica *Best Improvement* ou *First Improvement*, removendo tarefas de máquina i e atribuindo estas a máquina w .
- Se ao final apresentar melhora, retorna a solução.
- Caso contrário, volta-se para o passo 2 até não haver máquinas $w \neq i$ para serem exploradas.

Estruturas de Vizinhança: Troca Externa - NE.EI

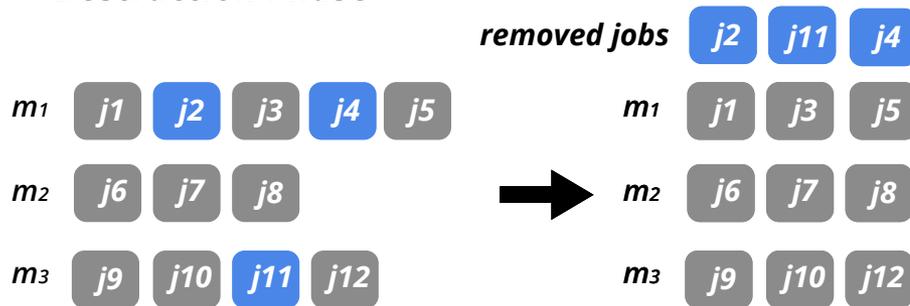
- Uma tarefa k , atribuída a máquina i , e uma tarefa j atribuída a uma máquina w , sendo $i \neq w$, são trocadas entre si.



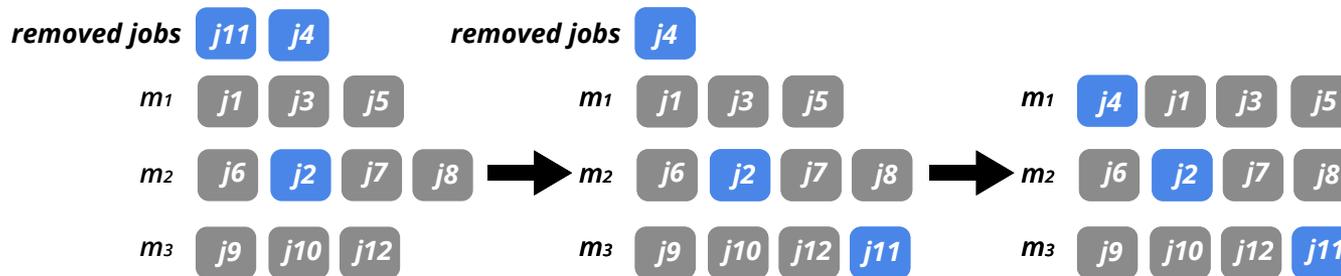
- Step 1: Seleciona aleatoriamente uma máquina i , onde $\sum_{k \in N_i} \alpha_k T_k > 0$.
- Step 2: Seleciona aleatoriamente uma máquina w , sendo $w \neq i$.
- Step 3: Aplica *Best Improvement* ou *First Improvement*, para todas as atribuídas as máquinas i e w .
- Step 4: Se ao final apresentar melhora, retorna a solução.
- Step 5: Caso contrário, volta-se para o passo 2 até não haver máquinas $w \neq i$ para serem exploradas.

Estruturas de Vizinhança: Inserção *Iterated Greedy* - NE

Destruction Phase



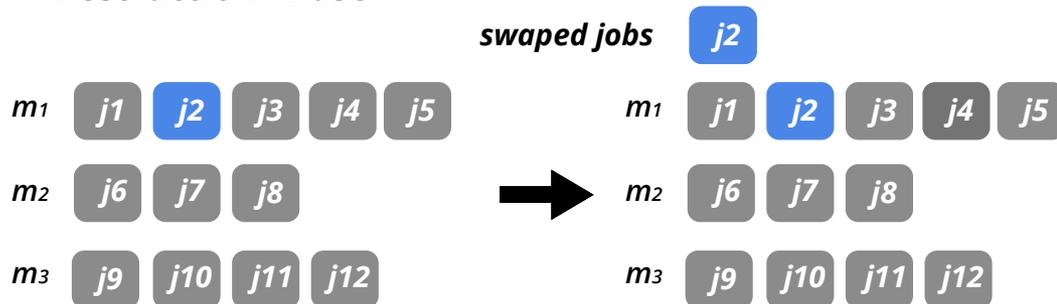
Construction Phase



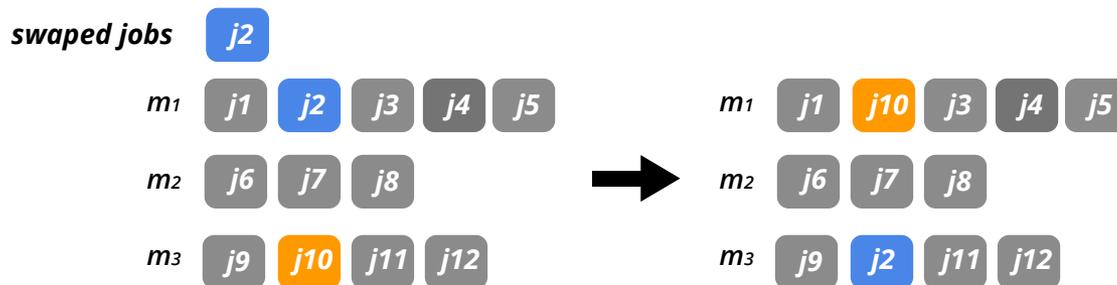
Step Seleciona, remove r_j tarefas e reconstrói a solução.

Estruturas de Vizinhança: Troca *Iterated Greedy* - NE.SIG

Destruction Phase



Construction Phase



Stg Seleciona, marca r_j tarefas e reconstrói a solução através do movimento de troca.

Ordem de exploração das estruturas de vizinhança

- Cardinalidade de permutações $NE.O_{card}$:

$$NE.O_{card} = \{NE.IGS, NE.SIGS, NE.IS, NE.II, NE.ES, NE.EI\} \quad (2)$$

- Impacto esperado na solução $NE.O_{imp}$:

$$NE.O_{imp} = \{NE.II, NE.IS, NE.EI, NE.ES, NE.IGS, NE.SIGS\} \quad (3)$$

Metodologia experimental

- Foram reprojctados os operadores de busca local de três metaheurísticas de estado da arte para minimização do TWT:
 - *Iterated Greedy Search* - IGS [2];
 - *Artificial Bee Colony* - ABC [7];
 - *Genetic Algorithm* - GA [8].
- Algoritmos implementados na linguagem Java, com JDK 1.8 disponíveis em <https://github.com/rodsxe/VND-UPMSSD>;
- Optamos neste trabalho por não recalibrar os parâmetros das metaheurísticas;
- Experimentos realizados sobre as instâncias propostas por *Lin e Hsieh* (2014)[3], divididas em 4 fatores:
 - ❶ Fator de severidade dos tempos de preparação: $\eta = [0.02; 2]$;
 - ❷ Fator de proximidade entre datas de entrega: $\tau = [0.3; 0.9]$;
 - ❸ Fator de proximidade das datas de entrega com o instante inicia de planejamento: $R = [0.25; 1]$;
 - ❹ Fator de proximidade dos *job ready times*: $r_\tau = [1; 10]$.
- Cada combinação de fatores $\eta \times \tau \times R \times r_\tau$ possui 20 instâncias e totalizam 16 grupos de instâncias *small* ($n \times m = 12 \times 3$) e *large* ($n \times m = 100 \times 10$);
- Critério de parada 1 e 60 segundos respectivamente para as instâncias do tipo *small* e *large* reescalado pelo *Single Thread Performance* (STP) da abordagem estado da arte IHM [3].

Metodologia: Projeto dos operadores de busca local

- *Hipótese i*: A ordem, método (variações VND) e forma que as estruturas de vizinhança são exploradas, podem incrementar/decrementar a performance do operador de busca local baseado em VND.
- Metodologia para de calibração robusta de fatores controláveis proposta por Taguchi [5, 6].
- Método utiliza um desenho de experimental do tipo fatorial fracionado.
- Avaliamos um fator com 2 níveis e sete fatores com 3 níveis.
- Em um desenho experimental fatorial completo necessitaria de $2 \times 3^7 r = 4374r$ experimentos, r sendo o número de replicas.
- Utilizou-se o arranjo ortogonal $L_{18}(2^1 \times 3^7)$ necessitando $18 * r$ experimentos.

Tabela: Lista de fatores controláveis para projeto do VND como operador de busca local.

Fatores Controláveis								
Níveis	Ordem	Variação VND	NE.II	NE.IS	NE.EI	NE.ES	NE.IGS	NE.SIGS
1	$NE.O_{card}$	Basic-VND	NA	NA	NA	NA	NA	NA
2	$NE.O_{imp}$	Random-VND	FI	FI	FI	FI	2	1
3	–	Union-VND	BI	BI	BI	BI	3	2

Metodologia: Projeto dos operadores de busca local

- Análise é dada pelo indicador *signal-to-noise ratio* (S/N_{ratio}).
- Problema de minimização utilizamos a função de perda “*smaller the better*”, expressa por:

$$S/N_{ratio} = -10 \log \left(\frac{1}{r} \sum_{i=1}^r y_i^2 \right) \quad (4)$$

sendo:

$$y_i = \frac{1}{10} \sum_{j=1}^{10} \frac{f(i, j) - best_{f(i)}}{best_{f(i)}} \quad (5)$$

- Resultados expressos por gráficos de efeitos principais.

Resultados: Projeto dos operadores de busca local

■ Metaheurística *Iterated Greedy Search*

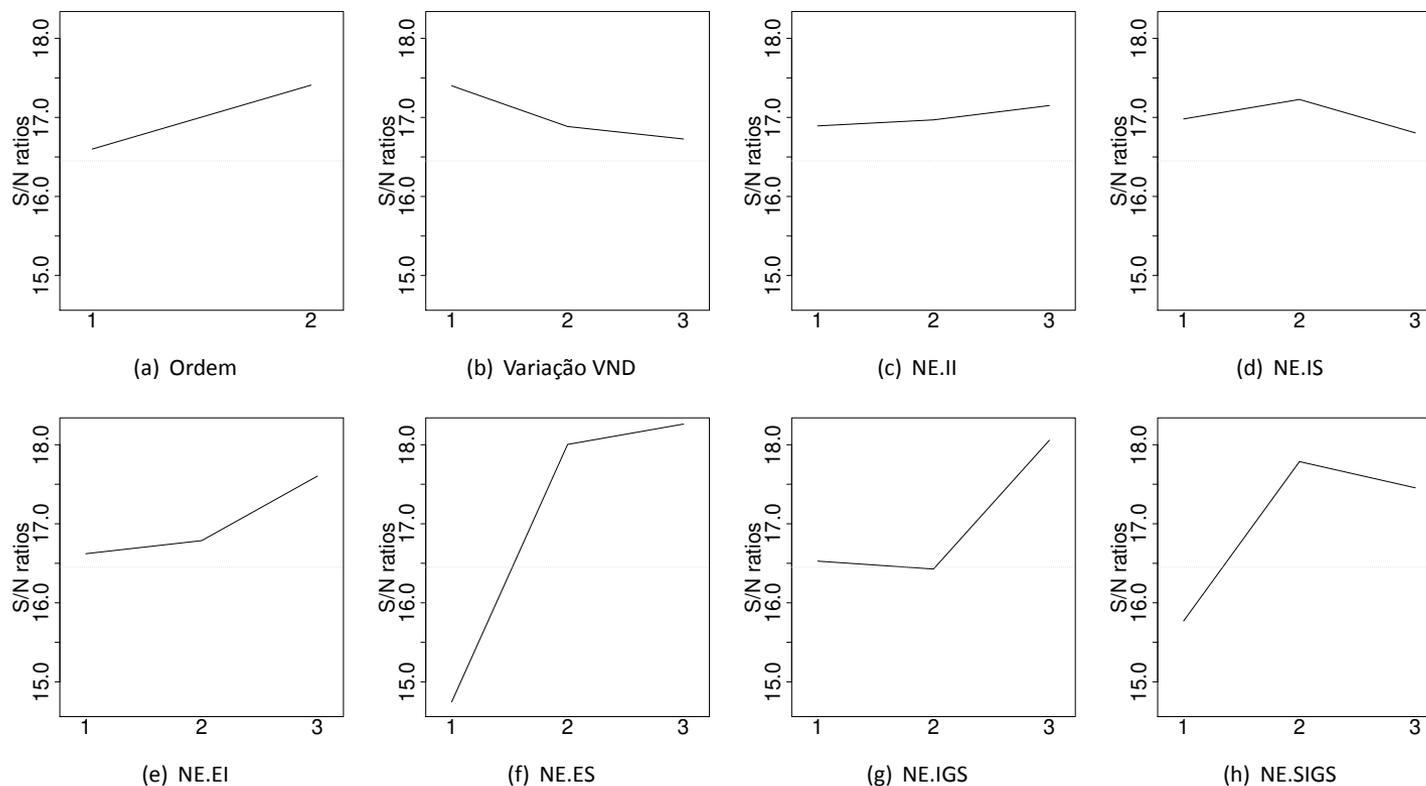


Figura: S/N_{ratio} para os fatores controláveis do VND com operador de busca local do IGS.

Resultados: Projeto dos operadores de busca local

■ Metaheurística *Artificial Bee Colony*

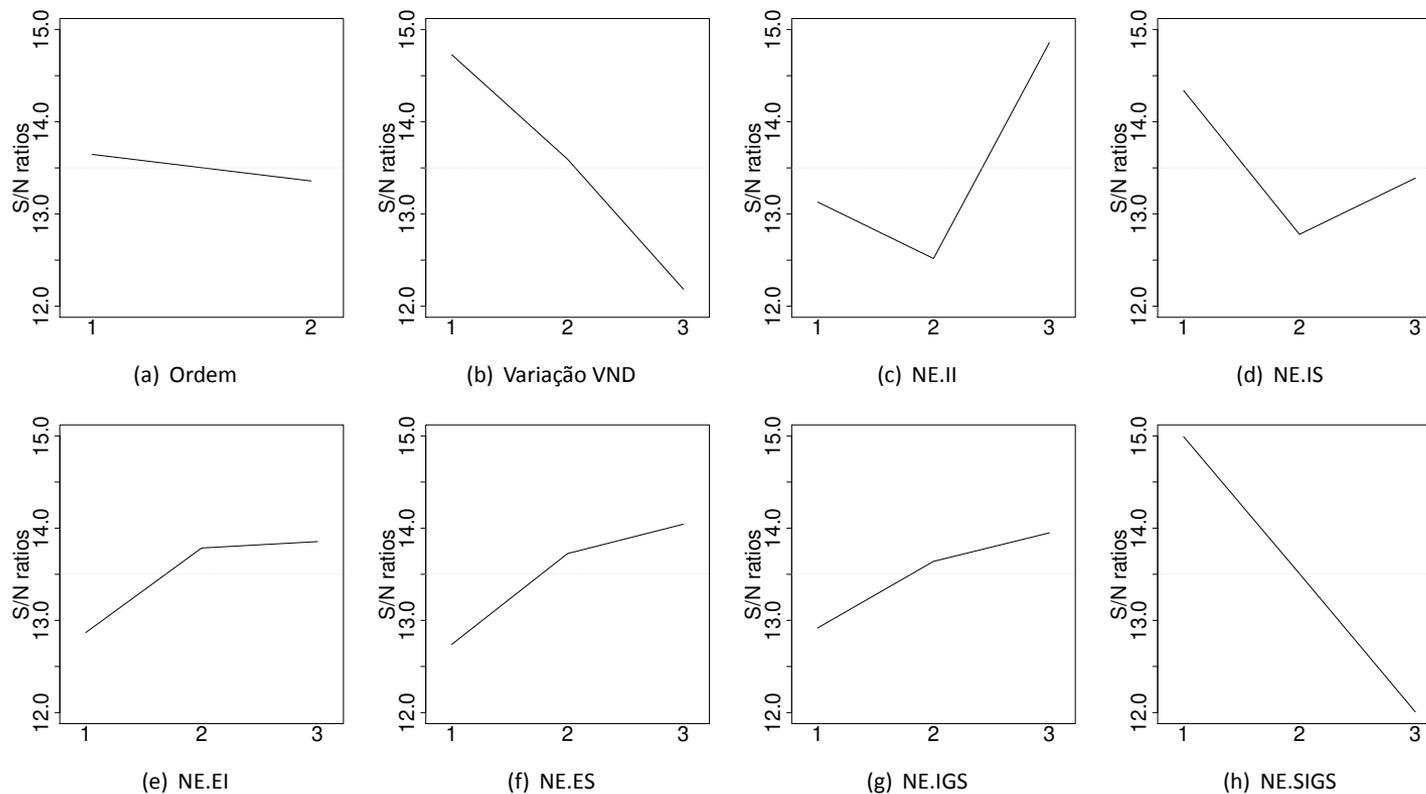


Figura: S/N_{ratio} para os fatores controláveis do VND com operador de busca local do ABC.

Resultados: Projeto dos operadores de busca local

■ Metaheurística *Genetic Algorithm*

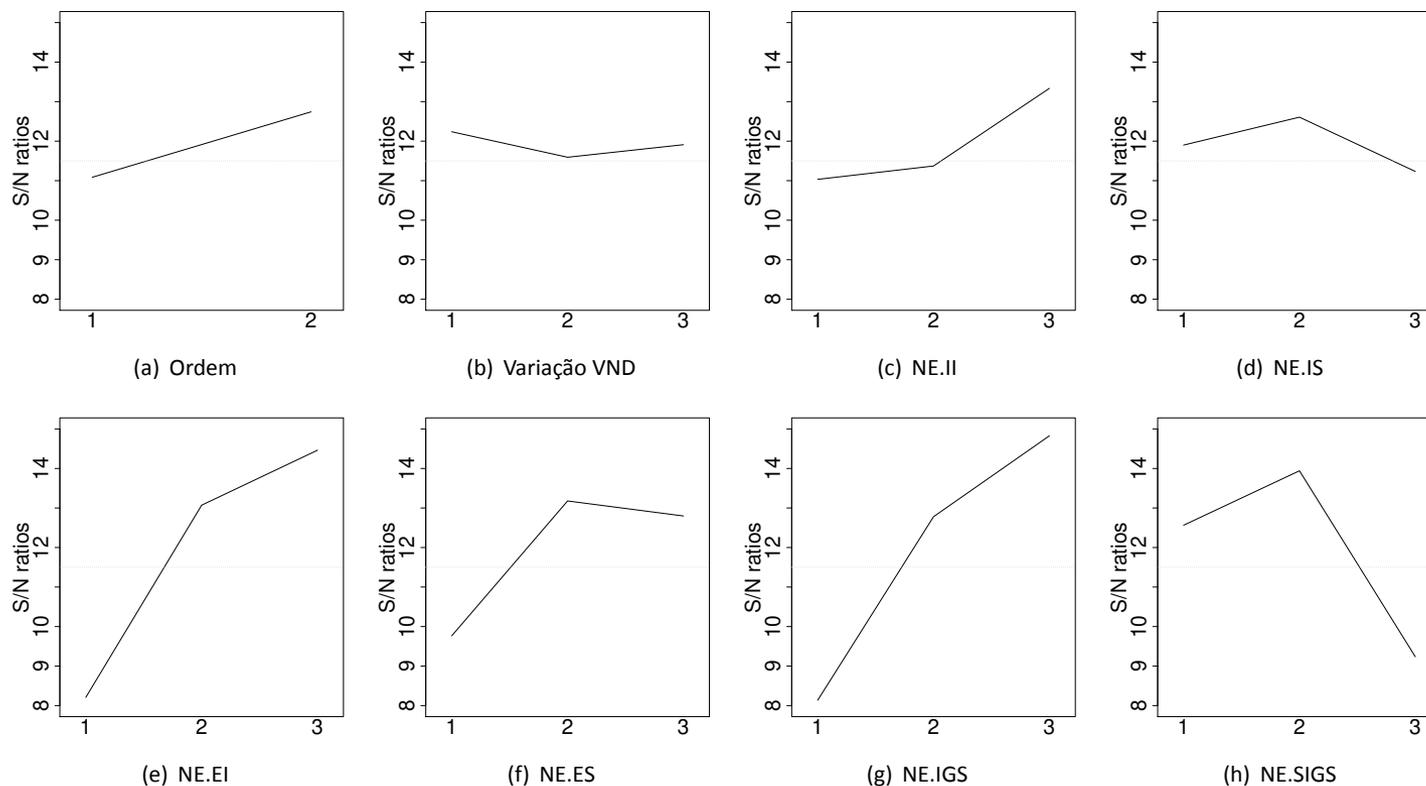


Figura: S/N_{ratio} para os fatores controláveis do VND com operador de busca local do GA.

Resultados: Projeto dos operadores de busca local

Tabela: Sumário da configuração do VND desenhado para cada metaheurística.

Fator	IGS	ABC	GA
Ordem	$NE.O_{imp}$	$NE.O_{card}$	$NE.O_{imp}$
Variação VND	<i>Basic-VND</i>	<i>Basic-VND</i>	<i>Basic-VND</i>
NE.II	BI	BI	BI
NE.IS	FI	NA	FI
NE.EI	BI	BI	BI
NE.ES	BI	BI	FI
NE.IGS	3	3	3
NE.SIGS	1	NA	1

Metodologia: Análise de estruturas de vizinhança

- *Hipótese ii*: Metaheurística tem diferentes demandas do operador de busca local, assim a importância de cada estrutura de vizinhança varia de acordo com a metaheurística avaliada.
- Análise de componentes por:
 - *Ablation*;
 - Remoção de estruturas de vizinhança.

Resultados: Análise de Ablation

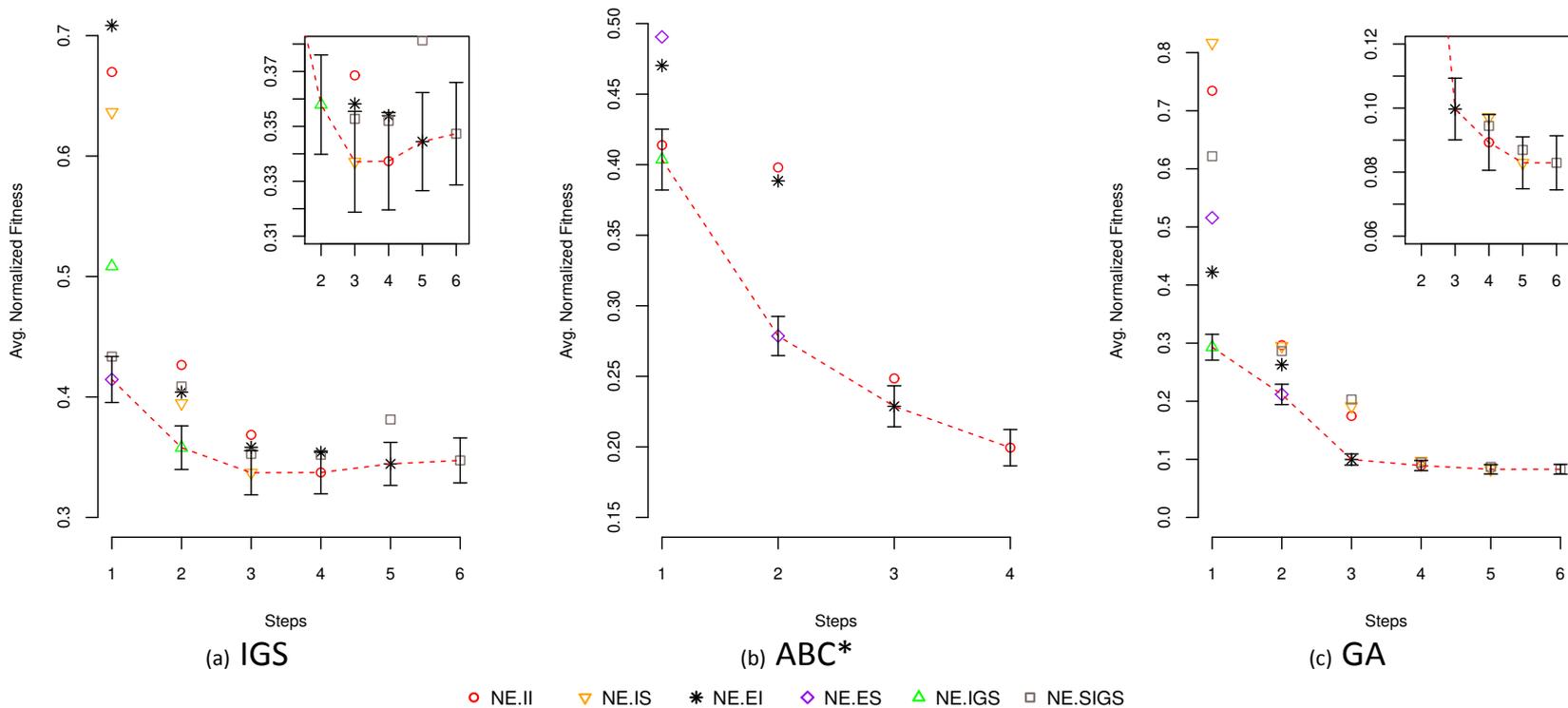


Figura: Análise de *ablation* do comportamento das estruturas de vizinhança para cada metaheurística. Cada ponto indica a performance da estrutura de vizinhança em cada passo. A linha ligando as estruturas de vizinhança indica os componentes que acarretaram em maior contribuição na convergência da metaheurística em cada passo.

*A calibração pela abordagem de Taguchi indicou que as estruturas de vizinhança NE.IS e NE.SIGS deterioraram a performance do VND integrado a abordagem ABC, assim, a influência destas duas estruturas não é analisada para a metaheurística ABC.

Resultados: Remoção de estruturas de vizinhança

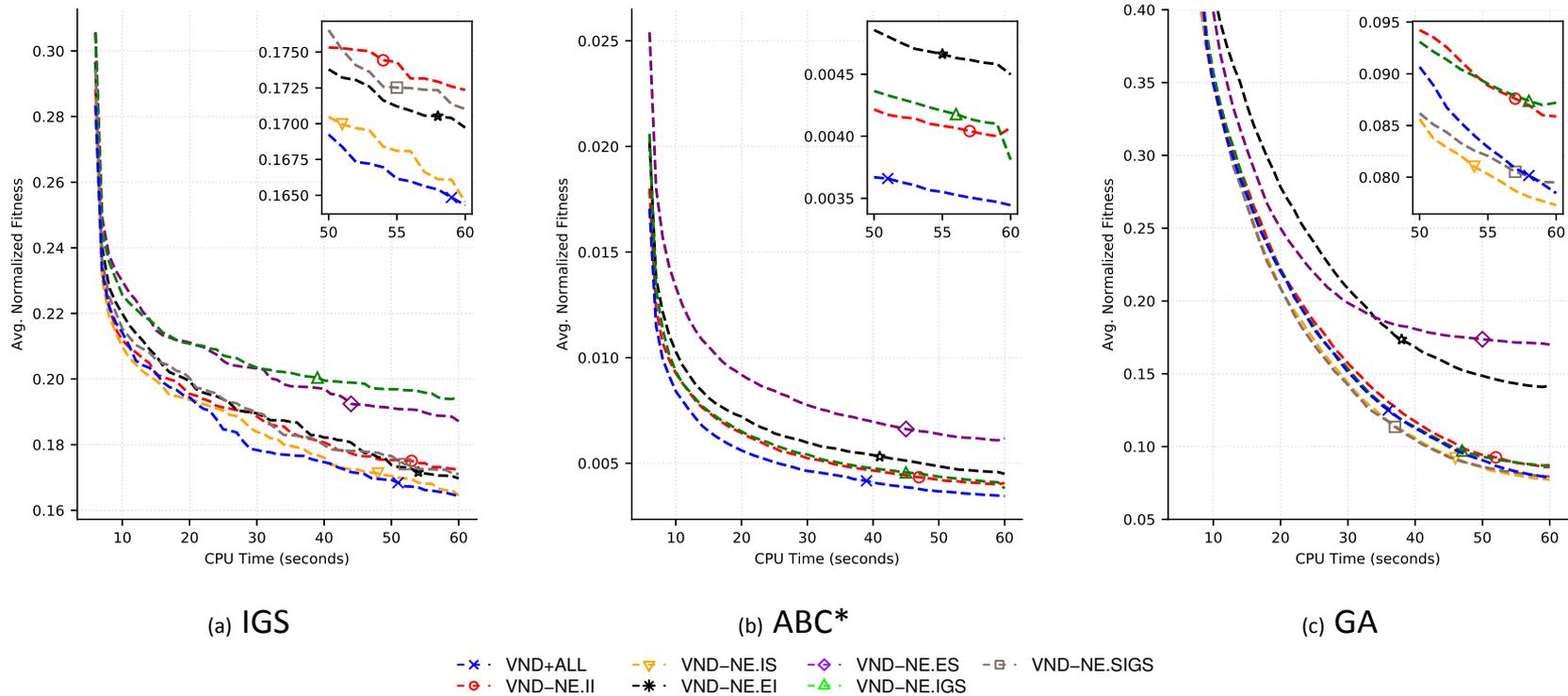


Figura: Gráficos de convergência para cada metaheurística. Cada configuração do VND considera a remoção de uma estrutura de vizinhança do VND afim de verificar a sua influência na convergência da metaheurística.
 *A calibração pela abordagem de Taguchi indicou que as estruturas de vizinhança NE.IS e NE.SIGS deterioraram a performance do VND integrado a abordagem ABC, assim, a influência destas duas estruturas não é analisada para a metaheurística ABC.

Metodologia: Comparação entre metaheurísticas

- *Hipótese iii*: O desempenho de uma metaheurística, já consolidada, pode ser incrementado apenas com a substituição do operador de busca local.
- Para cada metaheurística validamos a hipótese através do teste pareado Wilcoxon signed-rank test, sendo :

$$H_0 : md_1 - md_2 = 0 \quad (6)$$

$$H_1 : md_1 - md_2 < 0 \quad (7)$$

sendo md_1 e md_2 respectivamente, a mediana obtida pela metaheurística integrada ao VND e a mediana da abordagem com busca local nativa (IGS, ABC, GA).

- Teste realizado para os 16 grupos $\eta \times \tau \times R \times r_\tau$ e pareado pela média de 10 execuções de cada instância.
- Comparamos ainda as “novas” abordagens contra abordagens estado da arte IHM [3] e ILS [1] considerando:

$$H_0 : md_1 - md_2 = 0 \quad H_1 : md_1 - md_2 < 0 \quad (8)$$

$$H_0 : md_1 - md_2 = 0 \quad H_1 : md_1 - md_2 > 0 \quad (9)$$

Resultados: Comparação entre metaheurísticas

- IGS com operador de busca local reprojeto (IGS-VND) contra abordagens de estado da arte:
 - m_1 = Mediana IGS-VND; m_2 = Mediana abordagem comparada;

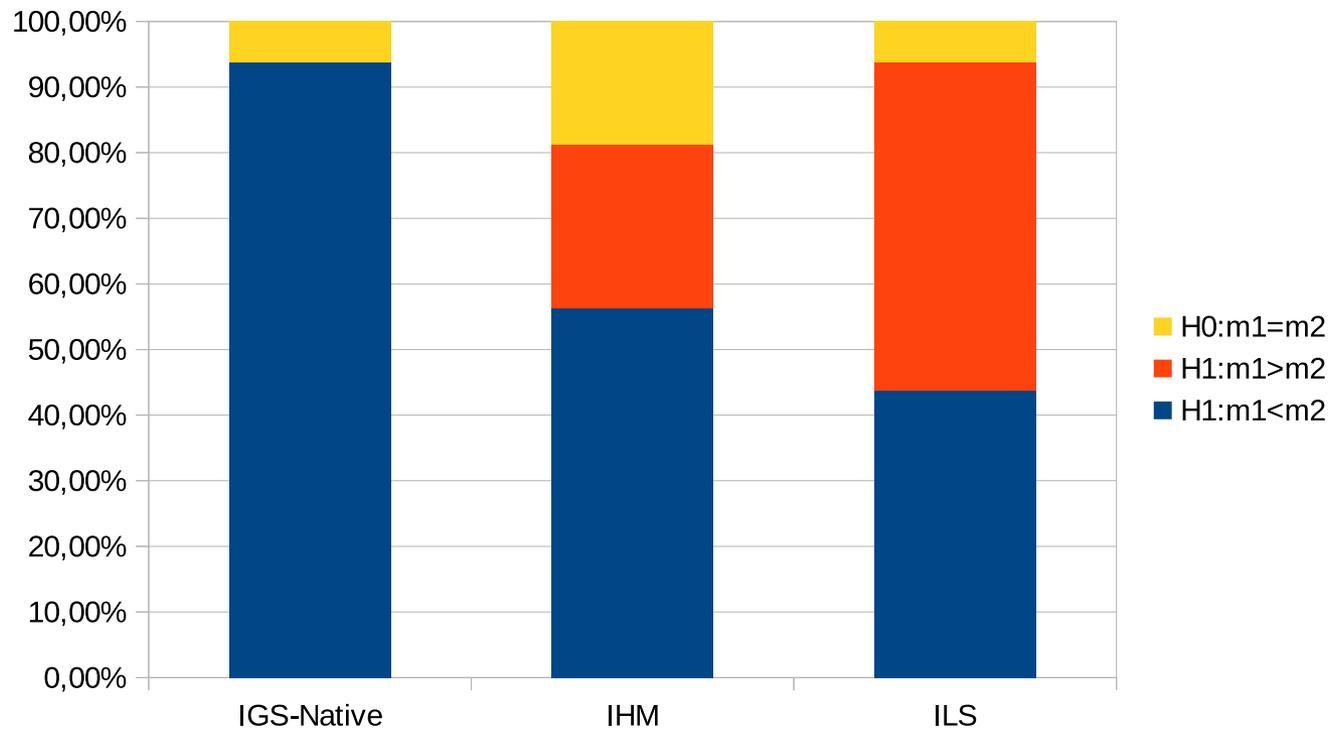


Figura: Sumário IGS-VND contra abordagens de estado da arte.

Resultados: Comparação entre metaheurísticas

- ABC com operador de busca local reprojetoado (ABC-VND) contra abordagens de estado da arte:
 - m_1 = Mediana ABC-VND; m_2 = Mediana abordagem comparada;

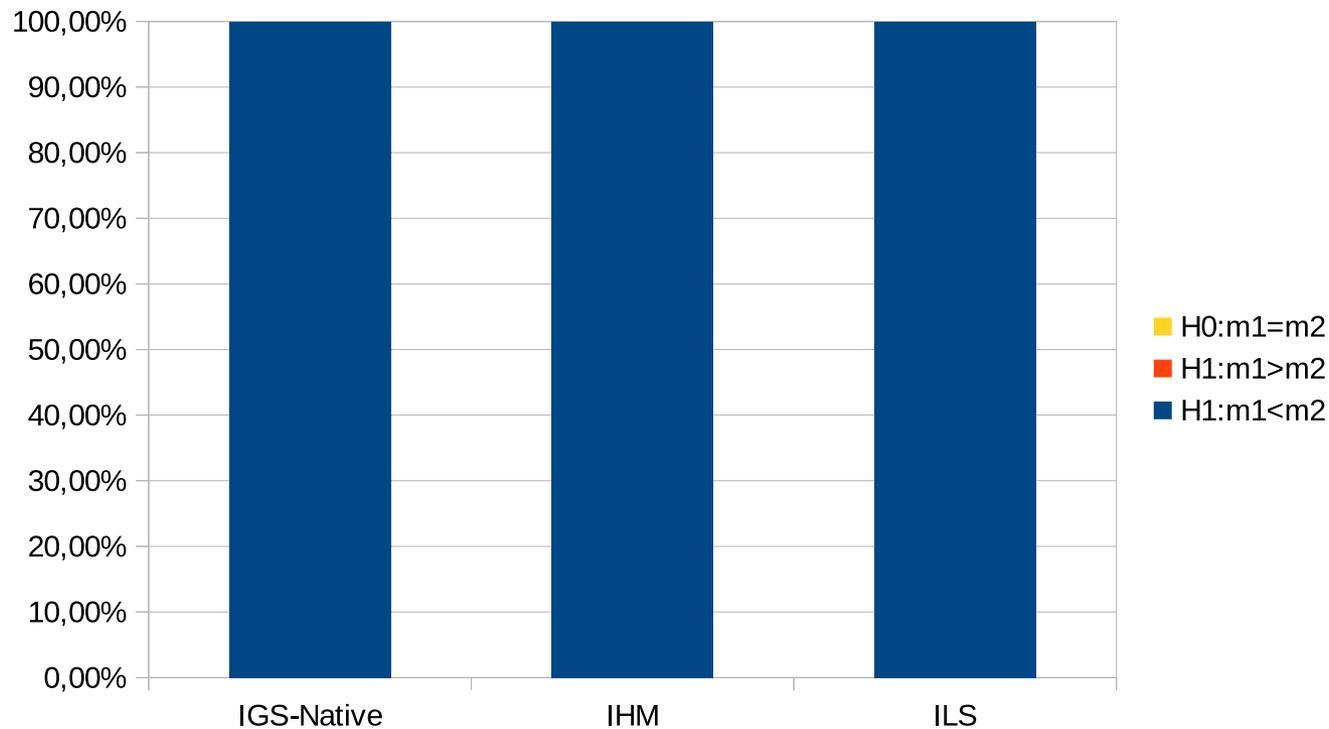


Figura: Sumário ABC-VND contra abordagens de estado da arte.

Resultados: Comparação entre metaheurísticas

- GA com operador de busca local reprojetoado (GA-VND) contra abordagens de estado da arte:

- m_1 = Mediana GA-VND; m_2 = Mediana abordagem comparada;

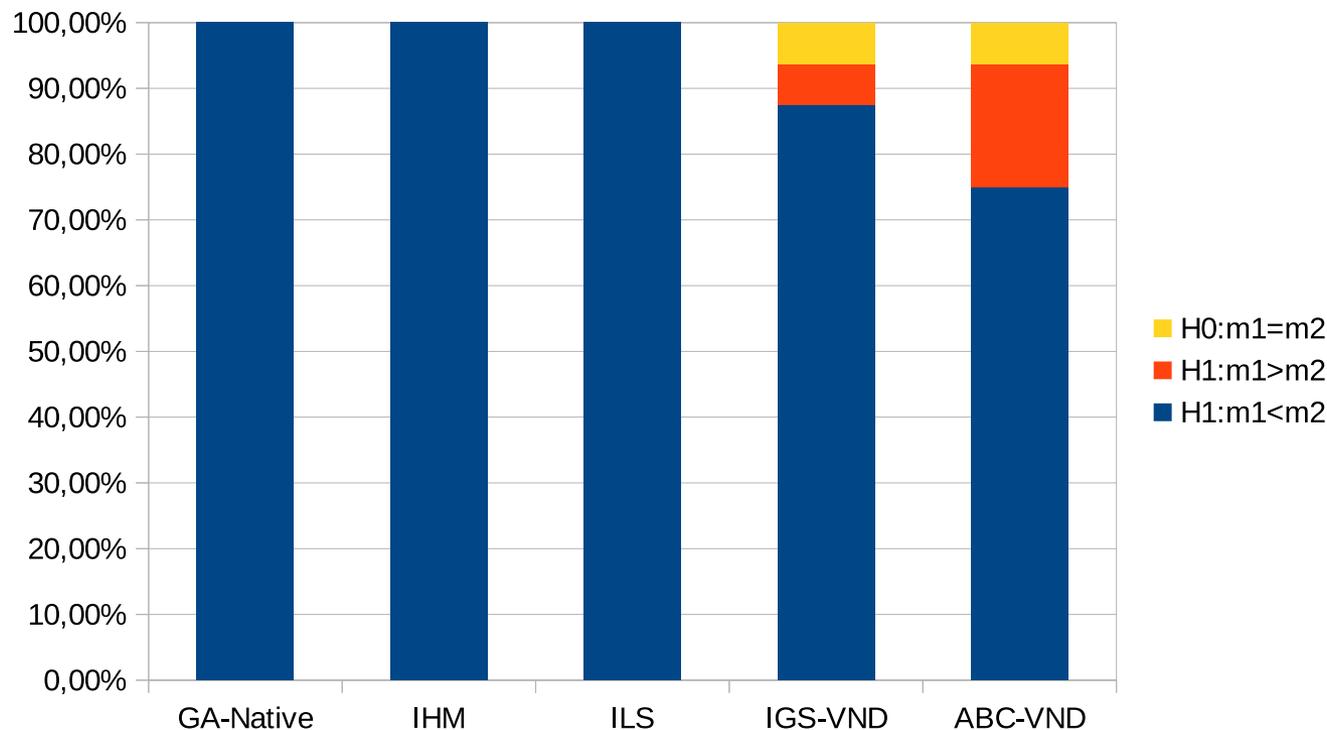


Figura: Sumário GA-VND contra abordagens de estado da arte.

Resultados: Comparação entre metaheurísticas

Tabela: Teste de Friedman para comparação do desempenho geral das metaheurística.

Chi-squared	df	p-value
1835.2	7	<2.2E-16

Tabela: Rank obtido pelo teste de Friedman do desempenho geral das abordagens baseada em metaheurísticas.

Metaheurística	Soma dos ranks	Posição*
GA-VND	422.5	a
ABC-VND	666.0	b
IGS-VND	1185.5	c
ILS	1269.0	d
IGS-LS	1607.5	e
IHM	1647.0	e
ABC-LS	2227.5	f
GA-LS	2459.0	g

*Posições com mesma letra indica que não há diferença significância entre as metaheurísticas.

Contribuições e considerações finais

- Proposto uma metodologia para projeto do *Variable Neighborhood Descent (VND)* como operador de busca local de metaheurísticas para minimização do TWT em ambientes UPMSSDST.
- Reprojeto-se operadores de busca local para três metaheurísticas de estado da arte (i.e. IGS [2], ABC [7], GA [8]).
- Os resultados obtidos através do reprojeto dos operadores de busca local indicou importantes direções:
 - Ordem de exploração das estruturas são bastantes relevantes.
 - *Basic-VND* indica ser mais relevante.
 - Três estruturas de vizinhança (i.e. NE.EI, NE.ES e NE.IGS) indicam dominar a convergência.
 - As estruturas de vizinhança NE.IGS, NE.ES trabalham em conjunto.
 - A estrutura NE.II bastante relevante para ajuste fino.
 - Duas estruturas, NE.IS e NE.SIGS, pouco relevantes no processo de busca.
 - Um projeto criterioso (mais científico) de operadores de busca local pode acarretar em incremento significativo no desempenho das abordagens metaheurísticas.

Contribuições e considerações finais

Este estudo deu origem ao artigo:

Diana, R. O. M.; de Souza, S. R. (2020) *Analysis of variable neighborhood descent as a local search operator for total weighted tardiness problem on unrelated parallel machines. Computers & Operations Research*, v.117
DOI:10.1016/j.cor.2020.104886.

Bibliografia

-  R. O. M. Diana, S. R. de Souza, and M. F. França Filho. A variable neighborhood descent as ILS local search to the minimization of the total weighted tardiness on unrelated parallel machines and sequence dependent setup times. *Electronic Notes in Discrete Mathematics*, 66:191–198, 2018. 5th International Conference on Variable Neighborhood Search.
-  S.-W. Lin, C.-C. Lu, and K.-C. Ying. Minimization of total tardiness on unrelated parallel machines with sequence- and machine-dependent setup times under due date constraints. *The International Journal of Advanced Manufacturing Technology*, 53(1):353 – 361, 2011.
-  Y.-K. Lin and F.-Y. Hsieh. Unrelated parallel machine scheduling with setup times and ready times. *International Journal of Production Research*, 52(4):1200 – 1214, 2014.
-  N. Mladenovic and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.
-  G. Taguchi. *Introduction to Quality Engineering*. UNIPUB/Kraus International. White Plains, 1986.
-  G. Taguchi. *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Cost*. UNIPUB/Kraus International. White Plains, 1987.
-  K.-C. Ying and S. W. Lin. Unrelated parallel machines scheduling with sequence and machine dependent setups times and due dates constraints. *International Journal of Innovative Computing, Information and Control*, 8(5):3279 – 3297, 2012.
-  J. R. Zeidi and S. MohammadHosseini. Scheduling unrelated parallel machines with sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 81(9):1487–1496, Dec 2015.

