Flexible Job Shop Scheduling Problem with Transport Resources

Danilo Fernandes

Instituto de Computação – IC

Formulação do problema

O problema é constituído de um conjunto J de tarefas independentes, um conjunto M de máquinas e um conjunto V de veículos idênticos

Cada tarefa $i \in J$ é composta por n_i operações ordenadas $N_i = \{O_{ij} : j = 1, 2, \ldots, n_i\}$

Cada operação O_{ij} deve ser processada sem preempção em uma única máquina $m \in M_{ij}$, onde $M_{ij} \subset M$ é o conjunto de máquinas capazes de processar O_{ij}

Cada operação O_{ij} necessita de uma quantidade p_{ij}^m de tempo para ser processada na máquina m

Formulação do problema

Cada máquina $m \in M$ é capaz de processar uma única operação por vez

Caso O_{ij} e O_{ij+1} sejam alocadas nas máquinas p e $q \in M$ com $p \neq q$, será necessário um tempo t_{pq} para transportar a tarefa de uma máquina à outra

Cada transporte é realizado por um único veículo $v \in V$

Cada veículo $v \in V$ pode transportar uma única tarefa por vez

Todo veículo $v \in V$ pode trafegar de uma máquina à outra sem estar transportando tarefa alguma

Todo veículo $v \in V$ pode aguardar em uma máquina até o início do seu próximo tráfego (transportando uma tarefa ou não)

Formulação do problema

Toda tarefa $i \in J$ está inicialmente localizada em uma máquina L/U na qual aguarda para ser transportada à máquina onde será executada sua primeira operação

Todo veículo $v \in V$ está inicialmente posicionado na máquina L/U

Cada tarefa será concluída quando a execução de sua última operação for finalizada

Em toda máquina $m \in M$, não há limite na quantidade de operações que podem aguardar para serem executadas

Em toda máquina $m \in M$, não há limite na quantidade de tarefas que podem aguardar para serem transportadas para as próximas máquinas

O objetivo é minimizar o *makespan*, isto é, o tempo necessário para concluir todas tarefas

Formulação do problema

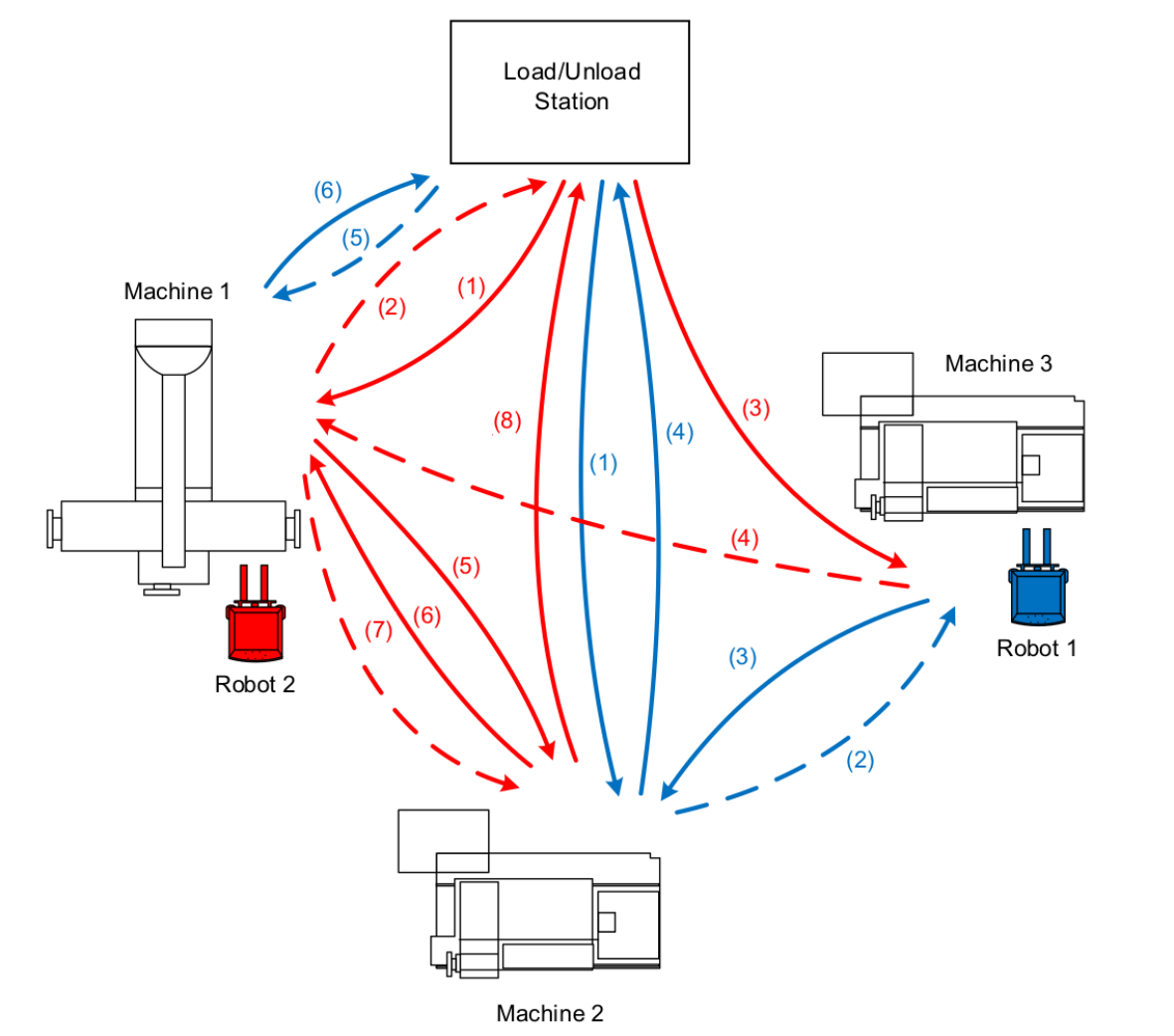

Figura 1: Ilustração de um escalonamento de máquinas e veículos

Formulação do problema

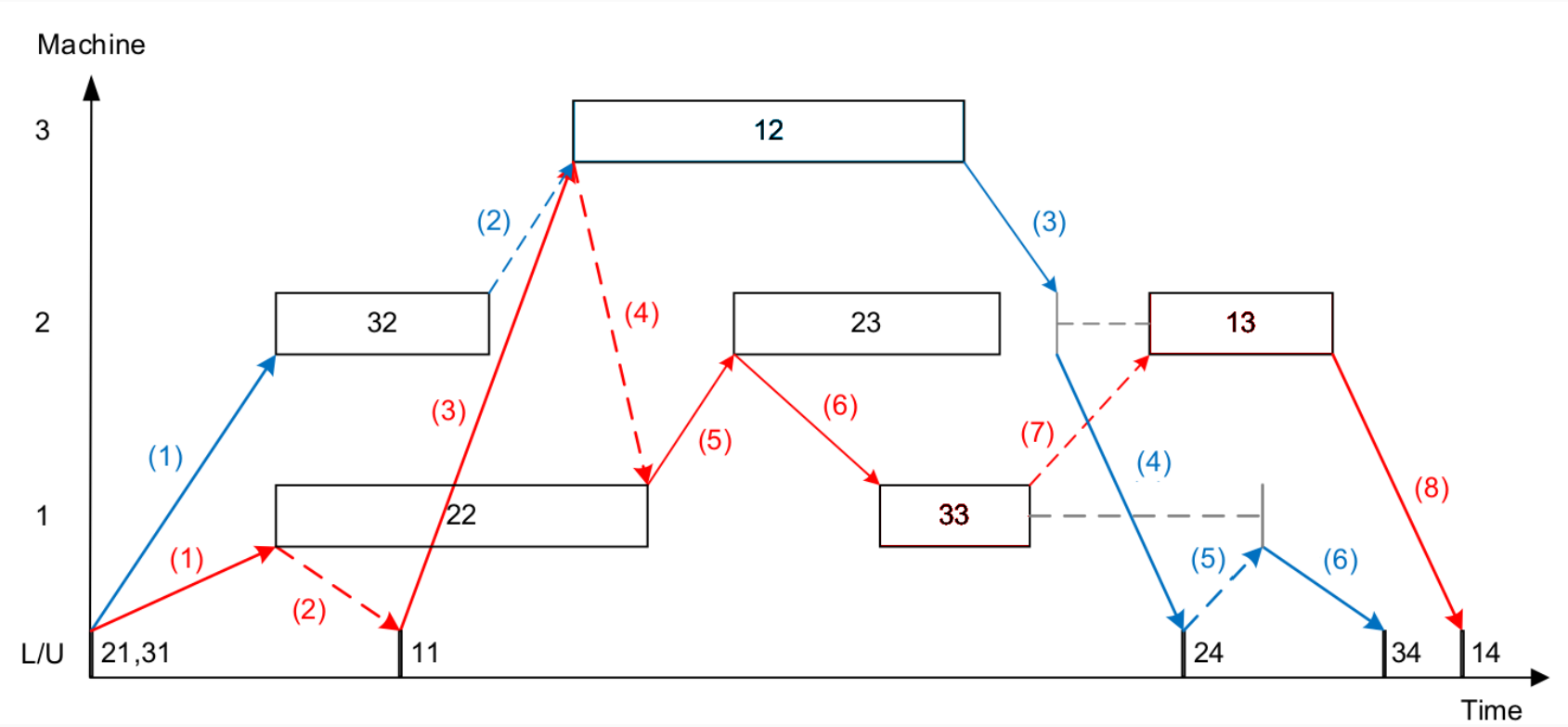

Figura 2: Gráfico de Gantt do escalonamento da figura 1

Revisão de literatura

Deroussi e Norre (2010) [2] foram os primeiros a propôr o problema e um conjunto de instâncias para o mesmo.

Adicionalmente, eles propuseram para o problema um *Iterated Local Search* (ILS) com as seguintes características:

- A solução é codificada em uma lista de transportes de tarefas a serem executadas por cada um dos veículos;
- As soluções vizinhas são obtidas pelos movimentos clássicos de inserção e troca;
- As perturbações consistem em três trocas aleatórias consecutivas;
- Um *Simulated Annealing* é usado para gerir a aceitação das soluções vizinhas;

Não foram reportados quaiquer resultados de experimentos computacionais.

Revisão de literatura

Kumar et al. (2011) [5] propuseram duas abordagens baseada em Evolução Diferencial e um novo conjunto de instâncias para o problema. Os resultados da aplicação destas abordagens sobre este conjunto de instâncias são reportados.

Zhang et al. (2012) [6] propuseram um procedimento hierárquico que combina um Algoritmo Genético com uma Busca Tabu. Adicionalmente, foram reportados resultados experimentais sobre o conjunto de instâncias porposto por Deroussi e Norre (2010).

Posteriormente, Zhang et al. (2013) [7] incrementaram esta abordagem com um procedimento de *Shifting Bottleneck*. Todovia, a qualidade das soluções, em média, permaneceu a mesma.

Revisão de Literatura

Deroussi (2014) [1] propôs uma abordagem que usa o PSO para resolver o problema de atribuição e escalonamento dos transportes e uma heuristica construtiva gulosa que atribui cada operação à primeira máquina disponível. Além disso, o escalonamento das operações em cada máquina é determinado pelo tempo de chegada.

Adicionalmente, foram reportados resultados experimentais sobre o conjunto de instâncias porposto por Deroussi e Norre (2010). Todavia as melhores soluções obtidas não foram tão boas quanto aquelas obtidas por Zhang et al. (2012, 2013).

Revisão de Literatura

Homayouni et al. (2020) [3] propuseram uma abordagem na qual utilizam um *Multistart Biased Random Key Genetic Algorithm* (BRKGA) para realizar o escalonamento das operações e um algoritmo heurístico guloso para atribuir os transportes aos veículos e as operações às máquinas.

Foram conduzidos experimentos nos conjuntos de instâncias propostos por Deroussi e Norre (2010) e Kumar et al. (2011). Em média, foram obtidos melhores resultados que Zhang et al. (2012, 2013) e Kumar et al. (2011).

Revisão de Literatura

Homayouni e Fontes (2021) [4] proporam a primeira modelagem em programação linear inteira mista para o FJSPT. Além disso, foi proposta uma abordagem baseada no *Late Acceptance Hill-Climbing Algorithm* e três novos conjuntos de instâncias para o problema.

Novamente foram conduzidos experimentos nos conjuntos de instâncias propostos por Deroussi e Norre (2010) e Kumar et al. (2011). Em média, foram obtidos melhores resultados que Homayouni et al. (2020).

Solução proposta

A abordagem proposta consiste em um Multi-start Iterated Local Search with Randomized Variable Neighborhood Descent (RVND).

Para a geração da solução inicial de cada ILS, será gerada uma sequência aleatória viável de todas as operações e em seguinda será empregado o Algoritmo Heurístico Guloso proposto em Homayouni et al. (2020).

Solução proposta

```
1:  procedure MultistartILS(I_R, I_ILS, I_P)
2:      MKS* ← ∞
3:      for iter = 1, ..., I_R do
4:          S ← GreedyHeuristic()
5:          S' ← S
6:          iter_ILS ← 0
7:          while iter_ILS < I_ILS do
8:              S ← LocalSearch(S)
9:              if MKS(S) < MKS(S') then
10:                 S' ← S
11:                 iter_ILS ← 0
12:             end if
13:             S ← Perturbation(S', I_P)
14:             iter_ILS ← iter_ILS + 1
15:         end while
16:         if MKS(S') < MKS* then
17:             S* ← S'
18:             MKS* ← MKS(S')
19:         end if
20:     end for
21:     return S*
22: end procedure
```

Solução proposta

```
1:  procedure LocalSearch(S)
2:      initialize the set of neighborhoods 𝒩 in a random order
3:      k ← 1
4:      while k ≤ 𝒩 do
5:          S′ ← BestImprovement(𝒩_k, S)
6:          if MKS(S′) < MKS(S) then
7:              reinitialize 𝒩 in a random order
8:              k ← 1
9:              S ← S′
10:         else
11:             k ← k + 1
12:         end if
13:     end while
14:     return S
15: end procedure
```

Solução proposta

Algorithm 2. Greedy heuristic algorithm for resource assignment

Retrieve O_{ij} the operation for resource assignment.
Get the available time t_v and location l_v of each vehicle and available time c_m of each machine.
if $j = 1$ **then**
  Assign vehicle v^* using a heuristic rule $v^* = \arg\min_{v \in V}\{t_v + T^{LU}_{l_v}\}$
  Calculate vehicle departure time from the LU area: $d_{v^*} = t_{v^*} + T^{LU}_{l_{v^*}}$.
  Assign machine m_{i1} using a heuristic rule : $m_{i1} = \arg\min_{m \in I_{i1}}\{\max\{c_m,\ d_{v^*} + T^m_{LU}\} + p^m_{i1}\}$
  Update available time and location of the assigned vehicle: $t_{v^*} = d_{v^*} + T^{m_{i1}}_{LU}$, $l_{v^*} = m_{i1}$.
  Update available time of the assigned machine and calculate operation completion time: $c_{m_{i1}} = c_{i1} = \max\{t_{v^*}, c_{m_{i1}}\} + p^{m_{i1}}_{i1}$.
else
  Get the completion time $c_{i(j-1)}$ of operation $O_{i(j-1)}$ and its assigned machine $m_{i(j-1)}$.
  Assign vehicle v^* using a heuristic rule $v^* = \arg\min_{v \in V}\{|t_v + T^{m_{i(j-1)}}_{l_v} - c_{i(j-1)}|\}$
  Calculate vehicle departure time from machine $m_{i(j-1)}$: $d_{v^*} = \max\{t_{v^*} + T^{m_{i(j-1)}}_{l_{v^*}},\ c_{i(j-1)}\}$.
  Assign machine m_{ij} using a heuristic rule: $m_{ij} = \arg\min_{m \in I_{ij}}\{\max\{c_m,\ d_{v^*} + T^m_{m_{i(j-1)}}\} + p^m_{ij}\}$
  if $m_{ij} \neq m_{i(j-1)}$ **then**
    Update available time and location of the assigned vehicle: $t_{v^*} = d_{v^*} + T^{m_{ij}}_{m_{i(j-1)}}$, $l_{v^*} = m_{ij}$,
  end if
  Update available time of the assigned machine and calculate operation completion time: $c_{m_{ij}} = c_{ij} = \max\{c_{m_{ij}},\ t_{v^*}\} + p^{m_{ij}}_{ij}$.
end if

Solução proposta

Jobs permutation	2	3	1	3	2	1	3	1	2	2	1	3

(a) *step 1: random jobs permutation*

Operations sequence	O_{21}	O_{31}	O_{11}	O_{32}	O_{22}	O_{12}	O_{33}	O_{13}	O_{23}	O_{24}	O_{14}	O_{34}

(b) *step 2: decoded random operations sequence*

Operations sequence	O_{21}	O_{31}	O_{11}	O_{32}	O_{22}	O_{12}	O_{33}	O_{13}	O_{23}	O_{24}	O_{14}	O_{34}
Machine assignment	1	2	1	1	2	1	1	1	1	1	2	2
Vehicle assignment	1	2	2	2	2	1	1	1	2	1	1	2

(c) *step 3: initial feasible solution matrix*

Operations sequence	O_{21}	O_{31}	O_{11}	O_{32}	O_{22}	O_{12}	O_{33}	O_{13}	O_{23}	O_{24}	O_{14}	O_{34}
Machine assignment	$M3$	$M2$	$M1$	$M3$	$M2$	$M3$	$M1$	$M2$	$M3$	$M1$	$M4$	$M4$
Vehicle assignment	1	2	2	2	2	1	1	1	2	1	1	2

(d) *decoded initial feasible solution matrix*

Figura 3: Exemplo ilustrativo: geração e decodificação de solução inicial (fonte: Homayouni e Fontes (2021))

Solução proposta

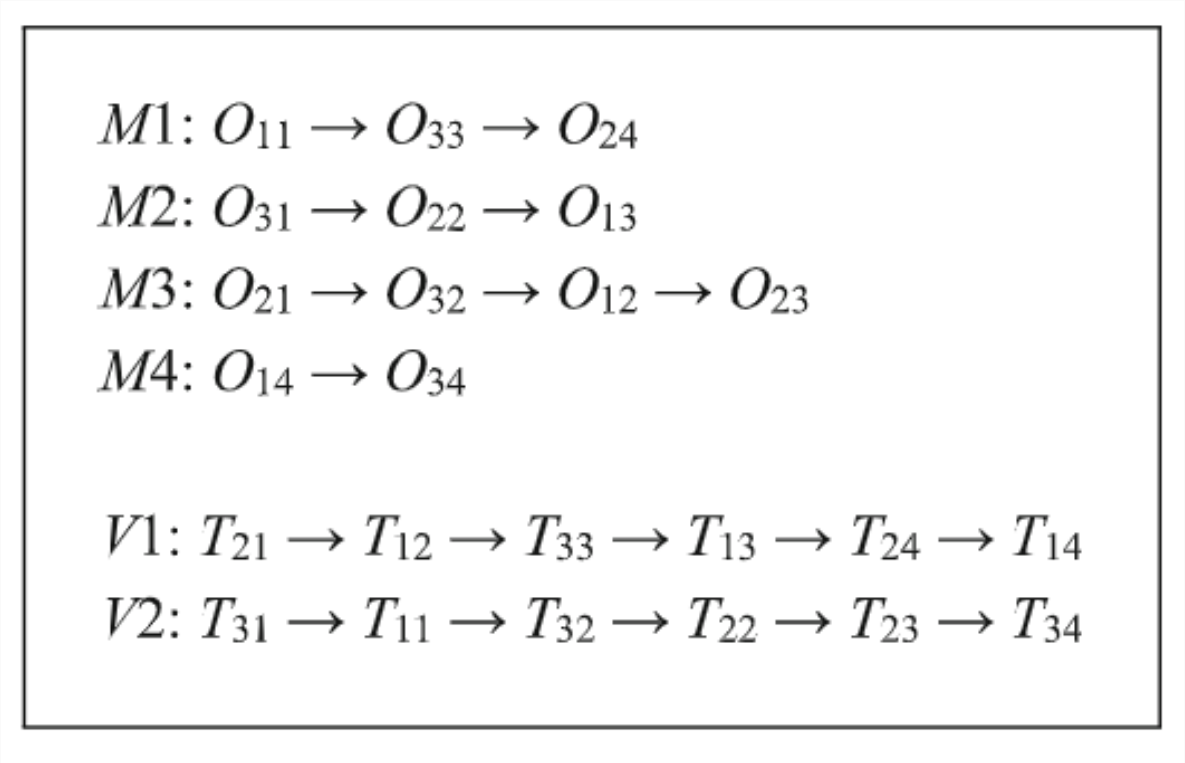

Figura 4: Exemplo ilustrativo: sequências das operações nas máquinas e veículos para a solução da figura 3 (fonte: Homayouni e Fontes (2021))

Solução proposta

Estruturas de vizinhança:

- Inserção de operação;
- Troca entre operações;
- Mudança de máquina;
- Mudança de veículo.

Perturbação:

- Três trocas aleatórias entre operações;
- Três mudanças aleatórias de máquina;
- Três mudanças aleatórias de veículo.

Resultados

Instance	MILP	Hybrid ILS				LAHC			
	C^*_{max}	C_{max}	GAP	$\sigma\%$	$\overline{T}$	C_{max}	GAP	$\sigma\%$	$\overline{T}$
fjsp1	134	138	2.99	0.79	7.43	138	2.99	1.46	5.13
fjsp2	114	114	0.00	0.71	3.60	114	0.00	1.51	4.70
fjsp3	120	120	0.00	0.00	4.57	120	0.00	1.43	6.90
fjsp4	114	116	1.75	0.91	7.30	116	1.75	1.69	8.87
fjsp5	94	94	0.00	0.00	1.73	94	0.00	0.00	2.87
fjsp6	138	138	0.00	1.04	6.37	138	0.00	1.27	6.67
fjsp7	110	112	1.82	1.27	7.90	114	3.64	1.72	6.13
fjsp8	178	178	0.00	0.10	9.07	178	0.00	0.37	7.37
fjsp9	144	144	0.00	0.00	5.47	144	0.00	1.13	6.50
fjsp10	174	174	0.00	0.59	10.87	174	0.00	1.06	8.47
Max			2.99	1.27	10.87		3.64	1.72	8.87
Mean			0.66	0.54	6.43		0.84	1.16	6.36
Min			0.00	0.00	1.73		0.00	0.00	2.87

Tabela 1: Resultados sobre o conjunto de instâncias de Deroussi & Norre (2010)

Resultados

Instance	Hybrid ILS				LAHC			
	C_{max}^{best}	C_{max}^{avg}	$\sigma\%$	$\overline{T}$	C_{max}^{best}	C_{max}^{avg}	$\sigma\%$	$\overline{T}$
MKT01	203	210.20	1.48	128.80	177	185.63	2.20	34.77
MKT02	142	146.70	1.83	173.33	128	134.80	2.59	51.10
Max			1.83	173.33			2.59	51.10
Mean			1.65	151.07			2.40	42.93
Min			1.48	128.80			2.20	34.77

Tabela 2: Resultados sobre instâncias do conjunto proposto por Homayouni & Fontes (2021)

Resultados

Instances	M – J – O – A
fjsp1	8 – 7 – 19 – 2
fjsp2	8 – 6 – 15 – 2
fjsp3	8 – 6 – 16 – 2
fjsp4	8 – 5 – 19 – 2
fjsp5	8 – 5 – 13 – 2
fjsp6	8 – 6 – 18 – 2
fjsp7	8 – 8 – 19 – 2
fjsp8	8 – 6 – 20 – 2
fjsp9	8 – 5 – 17 – 2
fjsp10	8 – 6 – 21 – 2
MKT01	6 – 10 – 55 – 2.1
MKT02	6 – 10 – 58 – 4.1

Tabela 3: Descrição das instâncias

Conclusão

Embora a abordagem proposta mostre-se bastante robusta e obtenha excelentes resultados em instâncias pequenas, a falta de estruturas de dados auxiliares capazes que atualizar o *makespan* em $O(1)$ tornou-a muito custosa computacionalmente em grandes instâncias quando comparada com a literatura.

Portanto, trabalhos futuros incluiriam a concepção e implementação de tal estrutura de dados auxiliar, o que possivelmente implicaria em uma representação diferente da solução.

References i


L. Deroussi.
A hybrid pso applied to the flexible job shop with transport.
pages 115–122, 05 2014.


L. Deroussi and S. Norre.
Simultaneous scheduling of machines and vehicles for the flexible job shop problem.
In *International Conference on Metaheuristics and Nature Inspired Computing*, pages 1–2, 2010.


M. Homayouni, D. Fontes, and J. Gonçalves.
A multistart biased random key genetic algorithm for the flexible job shop scheduling problem with transportation.
International Transactions in Operational Research, pages 1–29, 10 2020.

References ii


S. M. Homayouni and D. B. M. M. Fontes.
Production and transport scheduling in flexible job shop manufacturing systems.
Journal of Global Optimization, 79:463–502, 2021.


M. Kumar, R. Janardhana, and C. Rao.
Simultaneous scheduling of machines and vehicles in an fms environment with alternative routing.
The International Journal of Advanced Manufacturing Technology, 53:339–351, 03 2011.


Q. Zhang, H. Manier, and M.-A. Manier.
A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times.
Computers Operations Research, 39:1713–1723, 07 2012.

References iii


Q. Zhang, H. Manier, and M.-A. Manier.
Metaheuristics for job shop scheduling with transportation.
Metaheuristics for Production Scheduling, pages 465–493, 05 2013.