



DESENVOLVIMENTO E APLICAÇÃO DE UMA HEURÍSTICA HÍBRIDA AO PROBLEMA DE SEQUENCIAMENTO DE UMA MÁQUINA COM INDEXAÇÃO NO TEMPO

Álif Rafael Fernandes Reis

Genivaldo Ribeiro dos Reis – Selma Abadia Fernandes Ribeiro

Universidade Federal de Viçosa – Campus de Rio Paranaíba, Rodovia MG - 230 Km 7 –

Rio Paranaíba-MG – CEP:38810000 – Caixa Postal 22

alif.reis@ufv.br

Thiago Henrique Nogueira

Acrisio Sebastião Nogueira – Magna Valeria Nogueira

Universidade Federal de Viçosa – Campus de Rio Paranaíba, Rodovia MG - 230 Km 7 –

Rio Paranaíba-MG – CEP:38810000 – Caixa Postal 22

thnogueira.ufv@gmail.com

RESUMO

O sistema de distribuição logística *crossdocking* pode ser visto como uma aplicação do problema de sequenciamento de máquinas. O problema é formulado como um problema de sequenciamento do tipo *flowshop* com duas máquinas e restrições de *crossdocking*, no qual a função objetivo busca minimizar o *makespan* (C_{max}). Este tipo de problema exige que os *jobs* do segundo estágio inicie seu processamento antes da conclusão de seus *jobs* precedentes. A heurística híbrida parte de uma solução viável obtida através da heurística construtiva, otimizando o tempo computacional, o que resulta em melhorias importantes no cálculo do *makespan* para médias e grandes instâncias, não resolvidas pelo modelo exato. Para esse propósito, um modelo de programação linear inteira com formulação baseada em indexação no tempo é considerado. Experimentos computacionais foram realizados comparando o modelo, resolvido por um *solver* comercial, ao seu relaxamento linear e à heurística híbrida. Os resultados obtidos mostraram a eficiência da heurística híbrida proposta.

PALAVRAS CHAVE. *Programação Linear Inteira, Sequenciamento de máquinas, Crossdocking.*

Logística, Programação Linear Inteira, Relaxação Linear, Heurísticas Construtivas, Heurística Híbrida.

ABSTRACT

The crossdocking logistic distribution system can be seen as an application of the machine-sequencing problem. The problem is formulated as a flow-type sequencing problem with two machines and crossdocking constraints, in which the objective function seeks to minimize the makespan (C_{max}). This type of problem requires that second stage jobs begin processing before completion of their previous jobs. The hybrid heuristic starts from a viable solution obtained through constructive heuristics, optimizing the computational time, which results in important improvements in the calculation of the makespan for medium and large instances, not solved by the exact model. For this purpose, an integer linear programming model with formulation based on time indexing is considered. Computational experiments were performed comparing the model, solved by a commercial solver, to its linear relaxation and to the hybrid heuristic. The results obtained showed the efficiency of the proposed hybrid heuristic.

KEYWORDS. *Integer Linear Programming, Machine Scheduling, Crossdocking.*

Logistics, Integer Linear Programming, Linear Relaxation, Constructive Heuristics, Hybrid Heuristic.



1. Introdução

Diante do atual cenário das tendências recentes como a efetividade da cadeia de valores, economia de escala e eficiência logística, juntamente com os desafios contingenciais como a competitividade e sazonalidade, as organizações viram-se obrigadas a buscar alternativas estratégicas em prol da otimização de suas redes de distribuição física agregando ao máximo conjuntos de valores aos produtos e ou serviços destinados às partes interessadas (*stakeholders*). Uma destas alternativas pode ser desdobrada no sistema de distribuição logística *crossdocking*.

Segundo [Pinedo 2008] e [Nogueira et al. 2016], sequenciamento é um processo de tomada de decisão utilizado regularmente em muitos setores de manufatura e serviço, além de lidar com a alocação de recursos para tarefas dado um período de tempo, e tem como finalidade otimizar um ou mais objetivos. Neste contexto, o *crossdocking* pode ser visto como uma aplicação do problema de sequenciamento de máquinas sendo este bem difundido na literatura.

Crossdocking é uma abordagem que elimina ou reduz significativamente as duas funções mais dispendiosas dos centros tradicionais de distribuição que são a estocagem e a coleta de produtos [Cota et al. 2014]. Para [Fonseca 2015], [Pinedo 2008] e [Cota et al. 2014], o sistema *crossdocking*, também conhecido como distribuição *flow-through*, é considerado um diferencial altamente competitivo que apresenta um notório potencial para controlar os custos globais de distribuição e também reduzir ou eliminar os estoques não-produtivos da cadeia de suprimentos. Para isso, um Centro de *Crossdocking* (CCD) opera com estoque limitado ou, se possível, nulo. De acordo com [Cota 2016], os Centros de *Crossdocking* operam recebendo carretas completas (cargas consolidadas) de diversos pontos de fornecimento da cadeia de suprimentos, onde cada veículo é recebido em uma doca específica. Dentro do centro, as cargas são retiradas, recombinaadas e recarregadas em carretas de saída, de acordo com os pedidos específicos dos clientes [Fonseca 2015]. Estas carretas então deixam a instalação com carga combinada composta por produtos de diversos fornecedores, dedicada a um cliente ou destino específico. A lógica dos CCD é ilustrada na Figura 1.

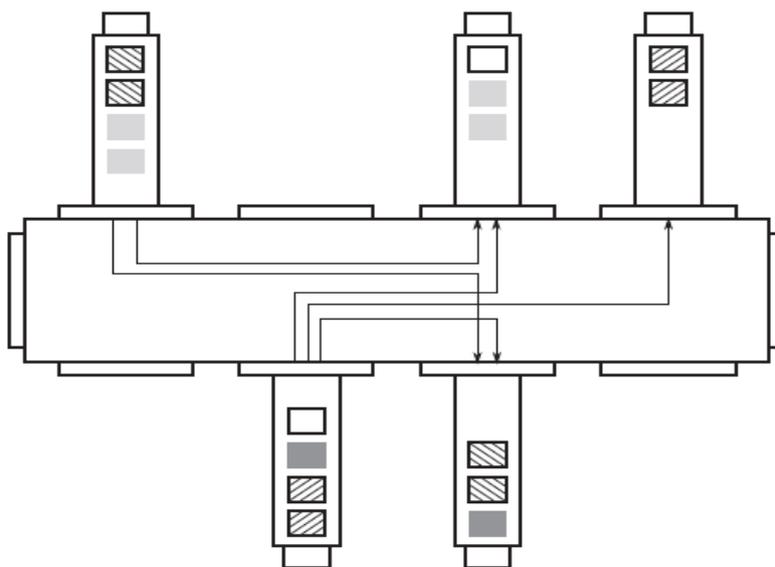


Figura 1 – Representação esquemática de um CCD. Adaptado de [Belle et al. 2012].

Este artigo busca aplicar o sequenciamento de máquinas com indexação no tempo em um centro de *crossdocking*. Para isso, um modelo de programação linear inteira indexado no tempo é utilizado. Além de propor uma heurística híbrida que parte da solução viável obtida através de uma heurística construtiva, melhorando-a. O modelo e a heurística híbrida são testados computacionalmente e seus resultados avaliados.



2. Trabalhos relacionados ao tema *Crossdocking*

Analisando as diferentes aplicações do sequenciamento de máquinas disponíveis na literatura, pode-se notar o quão amplo são as abordagens à respeito dos CCD com enfoques nas diversas etapas do processo logístico. Segundo [Belle et al. 2012], centros de *crossdocking* lidam com problemas envolvendo tomadas de decisões desde o nível estratégico ao operacional, os quais, de acordo com [Boysen e Fliedner 2010] são: localização do Centro de *Crossdocking*, layout, atribuição de docas, sequenciamento de caminhões, sequenciamento de recursos internos, roteamento de veículos.

[Chen e Lee 2009] estudaram o problema de sequenciamento de forma análoga ao problema clássico de *flowshop* com duas máquinas ($F2|CD|C_{max}$), cujo objetivo é minimizar o *makespan*. Segundo [Cota et al. 2014], esse problema considera as restrições de *crossdocking*, isto é, pode-se iniciar o carregamento em um caminhão se todos os produtos necessários para aquele caminhão tiverem sido descarregado na doca de entrada. Modelos indexados no tempo foram propostos inicialmente por [Souza e Wolsey 1992] no problema *non-preemptive single machine scheduling*. Para [Souza e Wolsey 1992] e [Cota et al. 2014], nos modelos indexados no tempo, o horizonte de tempo é dividido em períodos bem definidos e, assim as variáveis de decisão são também indexadas em cada um dos períodos estabelecidos. Tal abordagem remete na criação de modelos com um alto número de restrições e variáveis de decisão, portanto, suas relaxações lineares fornecem duas superiores muito mais fortes que outras formulações disponíveis na literatura.

3. Formulação do Problema

O modelo de programação inteira considerado neste trabalho adota a formulação de indexação no tempo proposta por [Fonseca 2015] e trata a função objetivo como proposto por [Chen e Lee 2009], cujo parâmetro de otimização é minimizar o *makespan* (C_{max}).

3.1 Definição do Problema

O problema será tratado neste estudo analogamente a um sequenciamento comum em linhas de produção de tal modo que docas recebem e descarregam ou carregam caminhões de maneira equivalente como máquinas processam ordens específicas de produção (*jobs*). Desta forma, temos que o problema é sequenciar o descarregamento dos caminhões de chegada e o carregamento dos caminhões de saída de forma a minimizar o tempo total de conclusão das tarefas (*makespan*).

Este problema foi proposto inicialmente por [Chen e Lee 2009], no qual o problema de sequenciamento de caminhões em um centro de *crossdocking* é definido como uma extensão do problema de *flowshop* com duas máquinas em série (*two-machine crossdocking flow shop problem*), cujo objetivo é minimizar o *makespan* (C_{max}). Nessa abordagem específica, existe apenas uma doca de recebimento e uma doca de despacho onde a primeira máquina representa a doca de entrada (máquina 1 - M1), a qual realiza a operação de descarregamento dos caminhões; e a segunda máquina representa a doca de saída (máquina 2 - M2), que realiza as operações de carregamento contendo produtos com um mesmo destino específico. Ainda para [Chen e Lee 2009], esse problema é caracterizado como NP-difícil e representado por: ($F2|CD|C_{max}$).

3.2 Modelo Matemático

Para facilitar o entendimento do modelo e da heurística híbrida proposta, a notação utilizada é sumarizada abaixo.

Conjuntos

- $J^1 = \{j_1^1, j_2^1, \dots, j_n^1\}$ representa os caminhões de chegada, que devem ser processados em M1.
- $J^2 = \{j_1^2, j_2^2, \dots, j_m^2\}$ representa os caminhões de saída, que devem ser processados em M2.

As restrições de *crossdocking* são representadas pela seguinte condição: para cada *job* $j_j^2 \in J^2$, existe um conjunto S_j de *jobs* em J^1 , tal que j_j^2 só pode ser processado em M2 se todos os *jobs* em S_j tiverem sido concluídos em M1. Considera-se que cada elemento do subconjunto S_j apresente pelo menos um elemento, ou seja, um determinado *job* $J \in J^2$ possui ao menos um *job* $J \in J^1$ como precedente.



Parâmetros de Entrada

- n : Número de jobs a serem processados na máquina 1.
- m : Número de jobs a serem processados na máquina 2.
- p_{ij} : Tempo de processamento do job j na máquina i .
- T : Horizonte de tempo considerado, uma primeira estimativa é a soma de processamento de todos os jobs.
- S_j : conjunto de subconjuntos precedentes de J^1 correspondentes ao job, $j \in J^2$, $S_j \in S$.
- $x_{ij} = 1$ se o job j começar no instante t , e igual a 0 caso contrário, $j \in J^1$.
- $y_{ij} = 1$ se o job j começar no instante t , e igual a 0 caso contrário, $j \in J^2$.
- C_{max} = Tempo que se demora para processar todos os jobs nos dois estágios.

Variáveis de Decisão

- A variável x_{jt} ($\forall j \in J^1, \forall t \in T$) será igual a 1, se o job j começar no instante t . Caso contrário o valor da variável será igual a 0.
- A variável y_{jt} ($\forall j \in J^2, \forall t \in T$) será igual a 1, se o job j começar no instante t . Caso contrário o valor da variável será igual a 0.

O modelo matemático completo é apresentado a seguir:

$$\text{Min } Z = C_{max} \quad (1)$$

sujeito à

$$\sum_{t=0}^{T-p_{1j}} x_{jt} = 1, \forall j \in J^1, \quad (2)$$

$$\sum_{t=0}^{T-p_{2j}} y_{jt} = 1, \forall j \in J^2, \quad (3)$$

$$\sum_{j \in J^1} \sum_{s=\max(0; t-p_{1j}+1)}^t x_{js} \leq 1, \forall t \in T, \quad (4)$$

$$\sum_{j \in J^2} \sum_{s=\max(0; t-p_{2j}+1)}^t y_{js} \leq 1, \forall t \in T, \quad (5)$$

$$\sum_{t=0}^{T-p_{2j}} ty_{tj} - \sum_{t=0}^{T-p_{1k}} (t+p_{1k})x_{kt} \geq 0, \forall j \in J^2 \wedge \forall k \in S_j, \quad (6)$$

$$C_{max} \geq p_{2j} + \sum_{t=0}^{T-p_{2j}} ty_{tj}, \forall j \in J^2, \quad (7)$$

$$x_{jt} \in 0,1, \forall j \in J^1 \wedge \forall t \in T, \quad (8)$$

$$y_{jt} \in 0,1, \forall j \in J^2 \wedge \forall t \in T, \quad (9)$$

$$C_{max} \geq 0. \quad (10)$$



O conjunto de restrições (2) diz que cada $job\ j_j^1 \in J^1$ deve iniciar seu processamento em uma, e somente uma, data dentro do horizonte de planejamento T , não sendo permitido que se associem duas ou mais datas distintas para o início do processamento de um job . De maneira análoga ao conjunto de restrições (2), o conjunto de restrições (3) trabalha com o mesmo raciocínio, porém aplicado aos $jobs$ processados nas máquinas do segundo estágio, $j_j^2 \in J^2$. As restrições indicadas por (4) garantem que um $job\ j_j^1 \in J^1$ não inicia seu processamento enquanto outro estiver sendo processado na máquina 1, da seguinte forma: para cada $t \in T$, a restrição verifica se algum $job\ j_j^1 \in J^1$ começou a ser processado em uma data maior ou igual a $t - p_{1j} + 1$. Caso afirmativo, j_j^1 ainda está sendo processado, e nenhum job pode iniciar seu processamento em t . O conjunto de restrições (5) trabalha de forma análoga com o anterior, no entanto é aplicado aos $jobs\ j_j^2 \in J^2$. O conjunto (6) trata das precedências dos $jobs$, denominadas restrições do tipo *crossdocking* (CD). Para cada restrição de precedência existente na instância, cria-se uma restrição desse conjunto, na qual, a data de liberação de cada job do segundo estágio, deve ser maior ou igual a data de conclusão da tarefa precedente. Caso o job possuir mais que um precedente, prevalece aquela de maior relativa ao job precedente com maior data de conclusão, isto é, as restrições relativas aos precedentes, que são processados primeiro, torna-se redundantes e prevalece aquela relativa ao job precedente com maior data de conclusão. O conjunto de restrições (7) indicam que a variável (C_{max}), *makespan*, deve ser predominantemente a maior data de conclusão dos $jobs$ do segundo estágio. Em última escala, os conjuntos de restrições (8) à (10) definem o *domínio* das variáveis do modelo. A função objetivo é dada por (1) e visa minimizar o tempo necessário para o processamento de todos os $jobs$ (*makespan*), ou seja, tem como objetivo específico minimizar a data de conclusão do último job processado pela máquina 2.

3.3 Heurísticas Construtivas Polinomiais

O procedimento heurístico polinomial é separado em dois estágios, sendo executado no primeiro estágio e posteriormente no segundo. O primeiro estágio corresponde aos $jobs$ de entrada e o segundo estágio corresponde aos $jobs$ de saída.

- ESTÁGIO 1:
 - i. O procedimento heurístico no primeiro estágio (máquina 1) via *LNS* (*Large Number of Successors*) ponderado pelo tempo de processamento;
 - ii. No empate, a heurística construtiva polinomial utilizada remete ao *LPT* (*Longest Processing Time*).
- ESTÁGIO 2:
 - i. No segundo estágio defina R (data de chegada), como a data de disponibilidade;
 - ii. Utilizar R como critério;
 - iii. Usar o *LPT* no desempate.

3.4 Heurística Híbrida Proposta

Visando obter soluções de melhor qualidade, uma implementação da heurística *Nawaz-Enscore-Ham* (NEH) é também proposta. O NEH é aplicado na solução gerada pela heurística construtiva polinomial, considerando a mesma como uma lista de prioridade, tendo como objetivo principal a ampliação das possibilidades de escolha de posicionamentos de uma nova tarefa no sequenciamento parcial, considerando, não somente a última posição da fila de processamento, mas todos os posicionamentos viáveis, que acarreta no menor aumento da função objetivo possível. Desta forma, a aplicação da heurística NEH, proposta por [Nawaz et al. 1983], na solução gerada pela heurística construtiva polinomial remete ao desenvolvimento da heurística híbrida proposta neste trabalho. O algoritmo da heurística NEH é apresentado abaixo:

Input: Uma lista ordenada de J trabalhos

Output: Uma solução viável

Para cada $j = 1$ até J faça:



MelhorMáquina $\leftarrow 0$;

MelhorPosição $\leftarrow 0$;

MelhorObjetivo $\leftarrow \infty$;

Para cada Máquina m faça:

Para cada posição P avaliada em m faça:

Objetivo \leftarrow função objetivo da solução parcial assumindo que j é sequenciado na posição P' da fila de processamento de m ;

Se Objetivo $<$ MelhorObjetivo então:

MelhorMáquina $\leftarrow m$;

MelhorPosição $\leftarrow P'$;

MelhorObjetivo \leftarrow Objetivo;

Sequencie $J [j]$ na posição MelhorPosição da fila de processamento da MelhorMáquina;

Retorne Solução

4. Experimentos Computacionais

Os experimentos computacionais foram realizados em um computador LENOVO com memória de 8GB, processador Intel Pentium Core i5 e sistema operacional LINUX Mint. Foi utilizado a linguagem de programação C++ e o software de otimização CPLEX 12.4 com as configurações padrões.

4.1. Geração de Instâncias

As instâncias utilizadas nesse trabalho foram baseadas nas instâncias utilizadas por [Chen e Lee, 2009]. As propriedades das instâncias geradas e aplicadas no modelo foram:

- Existem dois grupos de instâncias propriamente dito: o primeiro possui *jobs* curtos, cujos tempos de processamento foram gerados uniformemente entre 1 e 10. O segundo grupo apresenta *jobs* longos com tempos de processamento gerados uniformemente entre 10 e 100. O principal objetivo desta variação nos tempos de processamento remete na avaliação da influência do tamanho do horizonte de tempo do método de resolução.
- Para cada grupo de instâncias o número de *jobs* do primeiro estágio (máquina 1) é fixo e igual a 5, 10, 20, 40, 60, 80 ou 100. O número de *jobs* do segundo estágio (máquina 2) varia, sendo $n\alpha$, onde $\alpha = [0.6, 0.8, 1.0, 1.2, 1.4]$.
- O número máximo de precedentes para cada *job* na máquina 2 é gerado por meio de uma distribuição uniforme entre 1 e $n-1$.

Visando facilitar a referência ao longo da apresentação dos resultados e análises, cada subgrupo de instâncias geradas possuirá um código único característico sendo representado por $n-m-np-g$, onde n é o número de *jobs* na máquina 1, m o número de *jobs* na máquina 2, np é o número máximo de precedentes de cada *job* na máquina 2 e g é o grupo pelo qual pertence ao subgrupo. As Tabelas abaixo apresentam um resumo das instâncias geradas e suas características inerentes.

Os valores dos tempos de processamento dos *jobs*, presentes nos grupos J^1 ou J^2 foram definidos de forma aleatória, assim como o número de precedentes, de acordo com a distribuição uniforme definida na tabela. A configuração definida busca avaliar os resultados quando a quantidade de *jobs* é alterada. Assim, além desta configuração, é analisado as interferências relativas ao tempo de processamento dos *jobs* e ao número de precedentes de um *job* que será processado na máquina 2.



Tabela 1: Sumário das instâncias geradas para teste, separadas em grupo 1 e 2, e subgrupos, de acordo com o número de jobs na máquina 1 ($JobsM1(n)$). Além disso, a tabela informa o número de jobs na máquina 2 ($jobsM2(m)$), o número de precedentes do job $j \in J^2$ (NP), o tempo de processamento dos jobs (TP), o código da instância (Código).

Grupo	Jobs M1(n)	Jobs M2(m)	NP	TP	Código
1	5	3-4-5-6-7	U(1,4)	U(1,10)	m-np-1
1	10	6-8-10-12-14	U(1,9)	U(1,10)	m-np-1
1	20	12-16-20-24-28	U(1,19)	U(1,10)	m-np-1
1	40	24-32-40-48-56	U(1,39)	U(1,10)	m-np-1
1	60	36-48-60-72-84	U(1,59)	U(1,10)	m-np-1
2	5	3-4-5-6-7	U(1,4)	U(1,100)	m-np-2
2	10	6-8-10-12-14	U(1,9)	U(1,100)	m-np-2
2	20	12-16-20-24-28	U(1,19)	U(1,100)	m-np-2
2	40	24-32-40-48-56	U(1,39)	U(1,100)	m-np-2
2	60	36-48-60-72-84	U(1,59)	U(1,100)	m-np-2

4.2 Resultados Computacionais

Os resultados computacionais realizados estão dispostos nas Tabelas 2, 3 e 4 respectivamente. O GAP de otimalidade é calculado da seguinte forma:

$$GAP (\%) = \left(\frac{LS - LI}{LS} \right) \times 100.$$

Já a melhoria (Improvement) obtida com a heurística híbrida proposta é calculada da seguinte forma:

$$Improvement (\%) = \left(\frac{GAP_{Construção} - GAP_{NEH}}{GAP_{Construção}} \right) \times 100.$$

Para cada instância, diferentes problemas foram resolvidos e a média dos resultados é apresentada na Tabela 2. O tempo de execução foi limitado em 1 hora. Se em menos de 1 hora é encontrada a solução ótima do problema, a execução é automaticamente interrompida, caso a solução não seja encontrada em menos de uma hora, a execução é interrompida e os valores arquivados. O traço (-) significa que o valor correspondente não foi encontrado. A Tabela 3 descreve os resultados gerais com a aplicação da heurística híbrida.



Tabela 2: Comparação entre Modelo Completo e Relaxação Linear para o Grupo 1 de instâncias. A coluna Código da Instância refere-se a instância trabalhada, os itens MC e RL significam modelo completo e relaxação linear respectivamente. Já as siglas FO, Melhor_LI, LI, LS, GAP (%) e T representam a função objetivo, melhor limite inferior encontrado, limite inferior, limite superior, o GAP de otimalidade e o tempo de processamento em segundos respectivamente. O traço (-) significa que o valor correspondente não foi encontrado.

Código da Instância	MC				RL	
	FO	Melhor_LI	GAP (%)	T(seg)	FO	T(seg)
5-3-4-1	33	33	0	0.2	23	0.0
5-4-4-1	33	33	0	0.3	23	0.0
5-5-4-1	34	34	0	0.5	23	0.0
5-6-4-1	39	39	0	1.3	24	0.0
5-7-4-1	44	44	0	3.7	25	0.0
10-6-9-1	59	59	0	20.2	36	0.0
10-8-9-1	66	66	0	167.5	39	0.1
10-10-9-1	71	71	0	369.6	39	0.1
10-12-9-1	82	77	5	2440.5	40	0.1
10-14-9-1	90	86	5	3141.6	44	0.1
20-12-19-1	135	114	15	3600	65	0.3
20-16-19-1	153	102	32	3600	66	0.7
20-20-19-1	167	107	36	3600	68	0.6
20-24-19-1	190	115	38	3600	72	0.9
20-28-19-1	206	130	36	3600	82	1.0
40-24-39-1	306	164	46	3600	122	2.0
40-32-39-1	-	-	-	3600	126	3.5
40-40-39-1	-	-	-	3600	124	4.9
40-48-39-1	-	-	-	3600	136	12.7
40-56-39-1	-	-	-	3600	156	26.2
60-36-59-1	-	-	-	3600	176	34.5
60-48-59-1	-	-	-	3600	176	43.0
60-60-59-1	-	-	-	3600	181	133.1
60-72-59-1	-	-	-	3600	203	112.7
60-84-59-1	-	-	-	3600	241	122.8



Tabela 3: Comparação entre Modelo Completo e Relaxação Linear para o Grupo 2 de instâncias. A coluna Código da Instância refere-se a instância trabalhada, os itens MC e RL significam modelo completo e relaxação linear respectivamente. Já as siglas FO, Melhor_LI, LI, LS, GAP (%) e T representam a função objetivo, melhor limite inferior encontrado, limite inferior, limite superior, o GAP de otimalidade e o tempo de processamento em segundos respectivamente. O traço (-) significa que o valor correspondente não foi encontrado.

Código da Instância	MC				RL	
	FO	Melhor_LI	GAP (%)	T(seg)	FO	T(seg)
5-3-4-2	317	317	0	21	225	0.2
5-4-4-2	330	330	0	354.9	229	0.3
5-5-4-2	343	339	1	1570.1	227	0.4
5-6-4-2	393	340	12	2388.3	237	0.6
5-7-4-2	441	376	13	2882.5	246	1.9
10-6-9-2	625	438	29	3600	355	10.4
10-8-9-2	731	454	37	3600	363	12.4
10-10-9-2	820	443	45	3600	375	12.5
10-12-9-2	1014	435	56	3600	393	31.9
10-14-9-2	1125	415	62	3600	409	68.3
20-12-19-2	-	-	-	3600	651	195.4
20-16-19-2	-	-	-	3600	655	237.7
20-20-19-2	-	-	-	3600	674	322.6
20-24-19-2	-	-	-	3600	720	453.4
20-28-19-2	-	-	-	3600	828	1697.9
40-24-39-2	-	-	-	3600	1184	3600
40-32-39-2	-	-	-	3600	1176	3600
40-40-39-2	-	-	-	3600	1021	3600
40-48-39-2	-	-	-	3600	1280	3600
40-56-39-2	-	-	-	3600	-	3600
60-36-59-2	-	-	-	3600	-	3600
60-48-59-2	-	-	-	3600	-	3600
60-60-59-2	-	-	-	3600	-	3600
60-72-59-2	-	-	-	3600	-	3600
60-84-59-2	-	-	-	3600	-	3600



Tabela 4: Sumário dos resultados da heurística híbrida proposta. Os resultados encontrados são relativos às instâncias descritas na Tabela 1. O campo “Tempo” correspondente ao tempo computacional; “Construção” é a heurística construtiva polinomial; “NEH” é o resultado da heurística híbrida proposta; “Improvement” corresponde ao ganho percentual obtido ao incorporar a heurística NEH à heurística construtiva polinomial proposta; “BEST” representa os melhores resultados obtidos; “Average” representa a média dos resultados.

Código da Instância	Construção		NEH		Improvement	Código da Instância	Construção		NEH		Improvement		
	Gap	Tempo	Gap	Tempo			Gap	Tempo	Gap	Tempo			
5-3-4-1	Best	2.9%	0	0.0%	0	100.00%	5-3-4-2	Best	0.0%	0	0.0%	0	0.00%
	Average	11.7%	0	2.4%	0	79.74%		Average	9.1%	0	1.2%	0	87.18%
5-4-4-1	Best	0.0%	0	0.0%	0	0.00%	5-4-4-2	Best	1.3%	0	0.0%	0	100.00%
	Average	10.1%	0	3.6%	0	64.03%		Average	10.6%	0	3.9%	0	63.62%
5-5-4-1	Best	0.0%	0	0.0%	0	0.00%	5-5-4-2	Best	0.0%	0	0.0%	0	0.00%
	Average	6.8%	0	2.8%	0	58.64%		Average	6.9%	0	2.8%	0	58.65%
5-6-4-1	Best	0.0%	0	0.0%	0	0.00%	5-6-4-2	Best	0.0%	0	0.0%	0	0.00%
	Average	12.4%	0	4.7%	0	62.19%		Average	12.6%	0	4.6%	0	63.55%
5-7-4-1	Best	0.0%	0	0.0%	0	0.00%	5-7-4-2	Best	0.0%	0	0.0%	0	0.00%
	Average	11.1%	0	3.4%	0	69.33%		Average	11.1%	0	3.7%	0	66.53%
10-6-9-1	Best	10.1%	0	0.0%	0	100.00%	10-6-9-2	Best	9.4%	0	0.0%	0	100.00%
	Average	14.2%	0	1.9%	0	86.39%		Average	12.9%	0	1.4%	0	88.92%
10-8-9-1	Best	0.0%	0	0.0%	0	0.00%	10-8-9-2	Best	6.4%	0	0.0%	0	100.00%
	Average	14.2%	0	4.6%	0	67.78%		Average	13.6%	0	4.1%	0	70.24%
10-10-9-1	Best	4.0%	0	0.0%	0	100.00%	10-10-9-2	Best	4.8%	0	1.0%	0	78.88%
	Average	12.6%	0	4.9%	0	61.34%		Average	15.1%	0	6.3%	0	58.42%
10-12-9-1	Best	5.6%	0	0.0%	0	100.00%	10-12-9-2	Best	4.5%	0	0.0%	0	100.00%
	Average	15.7%	0	5.8%	0	63.21%		Average	15.5%	0	5.6%	0	64.12%
10-14-9-1	Best	6.9%	0	0.0%	0	100.00%	10-14-9-2	Best	7.6%	0	0.0%	0	100.00%
	Average	18.8%	0	5.9%	0	68.60%		Average	18.7%	0	6.6%	0	64.82%
20-12-19-1	Best	4.3%	0	1.5%	0	64.93%	20-12-19-2	Best	5.0%	0	1.0%	0	79.41%
	Average	13.4%	0	5.3%	0	60.33%		Average	12.8%	0	5.0%	0	61.40%
20-16-19-1	Best	8.4%	0	1.6%	0	80.48%	20-16-19-2	Best	8.4%	0	2.3%	0	72.85%
	Average	17.6%	0	8.9%	0	49.52%		Average	17.2%	0	9.0%	0	47.48%
20-20-19-1	Best	11.3%	0	4.4%	0	60.67%	20-20-19-2	Best	11.3%	0	4.8%	0	57.06%
	Average	20.3%	0	11.9%	0	41.62%		Average	20.4%	0	11.3%	0	44.83%
20-24-19-1	Best	14.8%	0	5.7%	0	61.68%	20-24-19-2	Best	14.5%	0	6.3%	0	56.69%
	Average	22.3%	0	13.7%	0	38.62%		Average	21.9%	0	13.5%	0	38.48%
20-28-19-1	Best	14.3%	0	1.1%	0	92.05%	20-28-19-2	Best	14.5%	0	0.9%	0	94.10%
	Average	23.2%	0	13.3%	0	42.56%		Average	22.6%	0	13.4%	0	40.57%
40-24-39-1	Best	11.2%	0	4.3%	0	61.43%	40-24-39-2	Best	11.9%	0	4.2%	0	64.51%
	Average	15.6%	0	9.2%	0	40.71%		Average	19.4%	0	9.0%	0	53.71%
40-32-39-1	Best	14.2%	0	9.6%	0	32.96%	40-32-39-2	Best	12.5%	0	8.0%	0	35.72%
	Average	18.3%	0	12.2%	0	33.09%		Average	17.8%	0	11.8%	0	33.81%
40-40-39-1	Best	16.8%	0	12.1%	0	28.07%	40-40-39-2	Best	15.6%	0	9.1%	0	41.94%
	Average	21.6%	0	13.7%	0	36.44%		Average	19.9%	0	13.2%	0	33.44%
40-48-39-1	Best	18.6%	0	11.5%	0	37.92%	40-48-39-2	Best	19.0%	0	12.2%	0	35.83%
	Average	23.1%	0	17.0%	0	26.24%		Average	23.1%	0	17.2%	0	25.55%
40-56-39-1	Best	21.6%	0	16.1%	0	25.52%	40-56-39-2	Best	21.8%	0	15.6%	0	28.28%
	Average	26.3%	0	19.7%	0	25.15%		Average	26.4%	0	19.6%	0	25.87%
60-36-59-1	Best	10.3%	0	5.3%	0	48.26%	60-36-59-2	Best	10.0%	0	5.3%	0	47.42%
	Average	12.8%	0	7.6%	0	40.36%		Average	12.7%	0	7.5%	0	40.79%
60-48-59-1	Best	15.8%	0	9.2%	0	41.46%	60-48-59-2	Best	15.9%	0	9.5%	0	40.22%
	Average	19.1%	0	12.3%	0	35.80%		Average	18.7%	0	12.3%	0	34.04%
60-60-59-1	Best	19.0%	0	12.2%	0	35.66%	60-60-59-2	Best	19.3%	0	12.2%	0	36.85%
	Average	23.3%	0	17.3%	0	25.98%		Average	23.2%	0	17.0%	0	26.87%
60-72-59-1	Best	19.0%	0	12.5%	0	34.24%	60-72-59-2	Best	19.6%	0	13.0%	0	33.76%
	Average	25.0%	0	18.9%	0	24.58%		Average	24.7%	0	18.7%	0	24.46%
60-84-59-1	Best	20.6%	0	16.9%	0	18.34%	60-84-59-2	Best	21.0%	0	17.5%	0	16.85%
	Average	26.5%	0	21.1%	0	20.41%		Average	26.4%	0	21.0%	0	20.69%
Média Geral dos Best		10.0%		5.0%		50.3%		10.2%		4.9%		51.7%	
Média Geral dos Average		17.4%		9.7%		44.5%		17.3%		9.6%		44.7%	

4.3 Análise dos Resultados

O experimento executado mostrou que o modelo proposto é eficiente para resolver instâncias com $n=5$ e $n=10$ jobs com tempos de processamento curtos, e em sua maioria, em baixo tempo computacional. Analisando as Tabela 2 e 3 respectivamente, pode-se notar que a solução ótima para as maiores instâncias do Grupo 1 não foram identificadas e no Grupo 2 apenas duas instâncias tiveram suas soluções ótimas identificadas, $GAP = 0$. Esse fato expõe a dificuldade de



resolução de modelos indexados no tempo. Segundo [Fonseca 2015], com a indexação no tempo, o número de variáveis é proporcional ao horizonte de tempo, e quanto maior o número de jobs da instância, maior o horizonte de tempo para sequenciá-los. Fatores estes que implicam no aumento da complexidade do problema, sendo necessariamente preciso um maior esforço computacional para resolvê-lo.

Por meio dos resultados expostos nas Tabelas 2 e 3 respectivamente, relacionados aos Grupos 1 e 2, considerando os jobs de entrada de tamanho entre 5 e 10, podemos notar que os GAP's são iguais a 0, logo o modelo resolve no ótimo. De 10 a 40 obtém-se um GAP médio de aproximadamente 19,4%. De 40 a 60 não foi possível resolver o problema, isto é, o valor correspondente não foi encontrado. Analogamente, ainda por meio dos resultados expostos na Tabela 3 relacionada ao Grupo 2, considerando os jobs de entrada de tamanho entre 5 a 10, podemos notar que os GAP's possui um valor médio de 25,5%. De 10 a 40 e de 40 a 60, temos que não foi possível resolver o problema, isto é, o valor correspondente não foi encontrado.

A partir da Tabela 4 observa-se que a heurística NEH implementada sobre as soluções viáveis obtidas pelas heurísticas, proporcionou melhorias importantes no cálculo do *makespan* para médias e grandes instâncias, não resolvidas pelo modelo exato. Como pode ser observado, a heurística híbrida desenvolvida fornece uma melhoria (Improvement) média, considerando os dois cenários (Grupo 1 e Grupo 2) de 50,3% para a média geral dos melhores resultados (Best) e 44,5% para a média geral dos resultados médios (Average) no *makespan* quando comparado com o *makespan* alcançado pela heurística construtiva polinomial com relação ao Grupo 1 e de 51,7% para a média geral dos melhores resultados (Best) e 44,7% para a média geral dos resultados médios (Average) no *makespan* quando comparado com o *makespan* alcançado pela heurística construtiva polinomial com relação ao Grupo 2 respectivamente.

De modo generalizado, para o Grupo 1, considerando os *jobs* de entrada de tamanho entre 5 e 10, na Tabela 2 o modelo matemático completo resolve no ótimo (GAP médio = 1,0%). Portanto, com a utilização da heurística híbrida NEH, cujos resultados estão descritos na Tabela 3, o GAP médio cai para 0,0096%, tendo uma melhoria de 99,04%. Considerando os jobs de entrada de tamanho entre 10 e 40, na Tabela 2 o modelo matemático completo apresenta um (GAP médio = 19,4%). Portanto, com a utilização da heurística híbrida NEH, o GAP médio cai para 4,53%, tendo uma melhoria de 76,65%. Já com os jobs de entrada de tamanho entre 40 e 60, na Tabela 2 o modelo matemático completo não apresenta uma resposta. Portanto, com a utilização da heurística híbrida NEH, o GAP médio sobe para 4,53%, tendo uma melhoria extraordinária. Analogamente, para o Grupo 2, considerando os jobs de entrada de tamanho entre 5 e 10, na Tabela 3 o modelo matemático completo resolve no ótimo (GAP= 0) apenas para duas instâncias resultando num GAP médio de 25,5% e com a utilização da heurística híbrida NEH, cujos resultados estão descritos na Tabela 4, o GAP médio cai para 0,1%, tendo uma melhoria de 99,61%. Considerando os jobs de entrada de tamanho entre 10 e 40, na Tabela 2 não foi possível resolver o problema, isto é, o valor correspondente não foi encontrado. Portanto, com a utilização da heurística híbrida NEH, o GAP médio é da ordem de 4,36%, tendo uma melhoria extraordinária. Já com os jobs de entrada de tamanho entre 40 e 60, na Tabela 2 o modelo matemático completo não apresenta nenhuma resposta e com a utilização da heurística híbrida NEH, o GAP médio é da ordem de 10,66 %, tendo uma melhoria extraordinária.

Dessa forma, a heurística híbrida desenvolvida proporcionou melhorias significativas para as médias e grandes instâncias, e resultados próximos ao ótimo para instâncias menores que foram comparadas ao modelo exato e a relaxação linear do modelo.

5. Considerações Finais

Neste trabalho, foi apresentado um novo modelo de Programação Matemática e uma heurística híbrida para o Problema de sequenciamento de uma máquina. O desempenho do modelo mostrou-se exclusivamente dependente do número de *jobs* e do tempo de processamento. A formulação matemática aqui considerada, por empregar indexação no tempo, apresentou dificuldades para resolver problemas característicos de maiores dimensões. Os resultados apontaram que a relaxação linear apresentou dificuldade de resolução nas instâncias de grande porte, consumindo grande magnitude de tempo. Já a heurística híbrida proposta, heurística NEH



incorporada à heurística construtiva polinomial, se mostrou bastante eficiente uma vez que obteve limites mais próximos do ótimo em tempos computacionais otimizados praticamente para todas as instâncias.

Diante do fatos expostos conclui-se que a utilização da heurística híbrida proposta é uma forma altamente eficiente para a resolução de problemas de sequenciamento de uma máquina com indexação no tempo, uma vez que com a incorporação da heurística NEH à heurística construtiva polinomial, como mostra a Tabela 4, resultou numa melhoria média de 50,3% dos GAP's para o Grupo 1 de instâncias e uma melhoria média de 51,7% dos GAP's para o Grupo 2 de instâncias. Ainda com relação à heurística híbrida desenvolvida e com os resultados da Tabela 4, podemos notar que sua aplicação no contexto do sequenciamento de uma máquina em um centro de *crossdocking*, causou melhorias impactantes e altamente eficientes no cálculo do *makespan* para todas as instâncias utilizadas, principalmente para as instâncias de médias e grandes dimensões.

6. Agradecimentos

Ao PIBIC/CNPq pelo apoio financeiro e por todas as equipes envolvidas na execução deste trabalho.

Referências

Belle, J. V.; Valckemaers, P.; Cattrysse, . Cross-Dock: State of the Art. Omega, leuven, Janeiro 2012.

Boysen, N. e Fiedner, M. (2010). Cross dock scheduling: Classification, literature review and Research legenda. *Omega: The International Journal of Management Science*, 38:413-422.

Chen, F.; Lee, C. Y. Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 2009.

Cota, P. M. Problema de Sequenciamento de Caminhões em Centros de Crossdocking com Múltiplas docas, Belo Horizonte, Maio 2016.

Cota, P. M.; Lira, E. G.; Ravetti, G. O problema de sequenciamento de caminhões em centros de crossdocking com múltiplas docas. Simpósio Brasileiro de Pesquisa Operacional, Belo Horizonte , Setembro 2014.

Fonseca, B. O problema de Sequenciamento de Caminhões em um Centro de Crossdocking com duas Máquinas. Universidade Federal de Minas Gerais. Belo Horizonte. 2015.

Nawaz, M.; Enscore, E. E.; Ham, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega: The International Journal of Management Science* , p. 91-95, 1983.

Nogueira, H. et al. Two-dedicated-parallel-machine scheduling approach for a two-dock truck scheduling problem in a cross-docking center, Rio Paranaíba - MG, 2016.

Paula, M. R. D. Heurísticas para a minimização dos atrasos em sequenciamento de máquinas paralelas com tempos de preparação dependentes da sequência. Universidade Federal de Minas Gerais. Belo Horizonte. 2008.

Pinedo, M. L. Scheduling: theory, algorithms, and systems. Springer Science & Business Media, 2008.

Souza, J.P. e Wolsey, L.A. (1992). A time indexed formulation on non-preemptive single machine scheduling problems. *Mathematical Programming*, 54: 353-367.