



## Sequenciamento de Tarefas em Máquinas Paralelas de Processamento em Lotes com Entregas

Gilson Ataliba de Faria<sup>a</sup>, José Elias Claudio Arroyo<sup>a</sup>, André Gustavo dos Santos<sup>a</sup>,  
Thiago Henrique Nogueira<sup>b</sup>, Jonatas Batista Costa das Chagas<sup>a</sup>

gilataliba@yahoo.com.br, jarroyo@dpi.ufv.br, andre@dpi.ufv.br,  
thnogueira.ufv@gmail.com, jonatas.chagas@ufv.br

<sup>a</sup>Universidade Federal de Viçosa - Viçosa - MG - Brasil

<sup>b</sup>Universidade Federal de Viçosa - Rio Paranaíba - MG - Brasil

### RESUMO

Neste artigo trata-se o problema de sequenciamento de tarefas com tempo de processamento distintos em máquinas paralelas idênticas que processam mais de uma tarefa simultaneamente. As tarefas deverão ser sequenciadas nos veículos capacitados que estão disponíveis para realizar as entregas em momentos específicos. O objetivo do problema é determinar o sequenciamento das tarefas que maximize o ganho da empresa, sabido que uma tarefa só poderá ser sequenciada se seu prazo de entrega for respeitado. Neste trabalho propôs-se primeiramente uma formulação de Programação Linear Inteira Mista para o problema, em seguida propõe-se uma relaxação do método heurístico proposto na literatura para determinar um limite superior (UB) para a solução, e juntamente uma heurística construtiva para a obtenção de uma solução inicial. Uma heurística VND foi proposta com o objetivo de determinar soluções de melhor qualidade tendo como entrada a solução gerada pela heurística construtiva. Os resultados alcançados comprovaram a eficiência dos métodos heurísticos quando comparados ao limite superior (UB).

**PALAVRAS CHAVE.** Sequenciamento de Tarefas, Processamento em Lotes, Prazo de Entrega, Heurísticas, Programação Linear Inteira Mista.

**Tópicos:** L&T - Logística & Transporte, MH - Metaheurísticas, OC - Otimização Combinatória

### ABSTRACT

This paper studies a job scheduling problem with different processing time scheduling on identical parallel machines. These machines can process more than one jobs simultaneously. The jobs must be scheduling on vehicles with fixed capacity, available to perform the deliveries at a given moment. The objective is to define the jobs scheduling that maximizes the company's profit, knowing that a job can only be sequenced if its deadline is respected. In this work we first proposed the application of a Mixed Integer Linear Programming Formulation, then we propose a relaxation of the heuristic method proposed in the literature to obtain an upper bound for the solution, and then a constructive heuristic to obtain an initial solution. A VND heuristic was proposed in order to better determine quality solutions having as input the constructive solution. The results show that the heuristics are efficient when compared to the Upper Bound.

**KEYWORDS.** Job Scheduling, Batching Scheduling, Delivery Due Date, Heuristics, Mixed Integer Linear Programming.

**Paper topics:** L&T - Logistic and Delivery, MH - Metaheuristics, OC - Combinatorial Optimization



## 1. Introdução

O Problema de Sequenciamento de Tarefas em Máquinas Paralelas de Processamento em Lotes (STMPL) com entregas surge em várias situações do mundo real. O STMPL com entregas é um problema NP-Difícil (ver [Li et al., 2015]) que apresenta grande importância prática, pois combina dois importantes componentes da logística necessários para atender de forma eficiente os clientes. Tais componentes são o sequenciamento da produção e o processo de alocação de veículos para entrega. De acordo com Chang e Lee [2004], com a popularidade dos conceitos de *Just in Time*, as empresas buscam tratar de maneira integrada diversos estados da cadeia de suprimentos na busca por níveis de inventários mais competitivos.

Por meio de estudos que buscam compreender e adaptar os processos logísticos, é possível definir estratégias que permitam às empresas atender melhor seus clientes e aumentar consequentemente seus ganhos. Segundo Fuchigami e Rangel [2014], o sequenciamento de tarefas está diretamente relacionado à otimização de diversas medidas de desempenho que visam o bom funcionamento do sistema e a satisfação dos clientes.

Nos últimos anos, tem-se notado a atenção dos pesquisadores frente aos problemas que envolvem sequenciamento de tarefas em máquinas paralelas de processamento com entregas. Segundo Damodaran e Chang [2008], este problema pode ser encontrado em testes de circuitos elétricos e trabalhos com metal. O problema pode se apresentar também nas indústrias de semicondutores, farmacêutica, aeronáutica e de logística (Mathirajan e Sivakumar [2006] citado por Li et al. [2015]).

Leung e Chen [2013] abordaram um problema que considera uma única linha de produção que processa tarefas individualmente. As tarefas são agrupadas em lotes e alocadas aos veículos disponíveis para realização das entregas aos clientes, respeitando o limite máximo de ocupação de cada veículo. Os autores supracitados propuseram algoritmos de tempo polinomial para resolver esse problema considerando os seguintes objetivos: minimizar o máximo atraso na entrega das tarefas; minimizar o número de veículos utilizados sujeito à condição proposta no objetivo anterior; minimizar a soma ponderada do atraso máximo e o número de veículos.

Agnētis et al. [2014] trataram o problema de sequenciamento da produção integrado a problemas de agendamento de entrega em lotes, considerando que o transporte possa ser realizado por um terceiro, o qual é responsável pelo sequenciamento da entrega. Os autores consideraram dois cenários, sendo que no primeiro, o fabricante determina a sequência de produção enquanto o provedor logístico terceirizado é responsável apenas por determinar o cronograma de entrega que minimiza o total de custos, respeitando os prazos de entrega. Já no segundo cenário, o provedor logístico terceirizado, além de ser responsável por determinar o cronograma de entrega, determina também a sequência de produção.

Cheng et al. [2015] consideram o processamento de tarefas que possuem dimensões e tempos de processamento distintos em uma única máquina capaz de processar mais de uma tarefa simultaneamente, quando respeitada a capacidade da máquina. Quanto à entrega, os autores consideram uma série de momentos (instantes de tempo), sendo que em cada momento tem-se um conjunto de veículos capacitados disponíveis para a realização das entregas. A função objetivo definida por Cheng et al. [2015] consiste na minimização dos custos totais de produção e entrega. Os mesmos autores desenvolveram um modelo matemático e propuseram a utilização da heurística Colônia de Formigas para tratar o problema. Os autores supracitados ainda frisaram a necessidade de se obter um limite inferior (*Lower Bound* - LB), com o intuito de testar o desempenho das heurísticas a serem implementadas.

Li et al. [2015] definiram o problema considerando uma lista de tarefas, com tempo de processamento idênticos, a serem processadas em lotes em máquinas paralelas idênticas com capacidade limitada pelo tamanho das tarefas sequenciadas. As tarefas devem ser sequenciadas e depois alocadas em veículos com capacidade fixa disponíveis em uma série de momentos de entrega. Para uma tarefa ser sequenciada deve-se respeitar o seu prazo limite de entrega estipulado



pelo cliente. O objetivo do problema trata-se da maximização dos lucros obtidos pelas tarefas sequenciadas. Li et al. [2015] propuseram uma heurística para solucionar o problema considerando as tarefas com tamanhos unitários. Os autores também propuseram duas heurísticas para resolver o problema assumindo que as tarefas apresentem tamanhos distintos. Os autores ainda propuseram a relaxação das próprias heurísticas propostas para obtenção de um limite superior (*Upper Bound* - UB) para a solução do problema. Li et al. [2015] citam como trabalho em aberto o estudo do mesmo problema quando as tarefas possuem tempos de processamento diferentes.

Neste trabalho, propõe-se diferentes formas de resolver o problema de sequenciamento de tarefas em máquinas paralelas idênticas de processamento em lotes com entrega considerando os tempos de processamento das tarefas distintos, colocado como tema em aberto por Li et al. [2015]. Para o tratamento deste problema, propõe-se uma formulação de programação linear inteira mista para obtenção de soluções ótimas. Como o problema trata-se de uma generalização do problema proposto por Li et al. [2015] que pertence à classe NP-Difícil, este problema também deve ser tratado como um problema NP-Difícil. Desta maneira, a formulação de um modelo matemático não é suficiente para resolver na otimalidade grandes instâncias de dados. Sendo assim, neste trabalho também propõe-se inicialmente utilizar o método proposto por Li et al. [2015] para a obtenção de um limite superior (*Upper Bound* - UB), e juntamente com a heurística construtiva (HC), proposta neste trabalho, para a obtenção de uma solução inicial. Em seguida aplica-se um algoritmo heurístico baseado no método *Variable Neighborhood Descent* (VND) para melhoria das soluções. As instâncias utilizadas para a realização dos experimentos foram geradas seguindo uma série de critérios que foram definidos baseando-se em Li et al. [2015].

## 2. Formulação do Problema

Considera-se o problema de sequenciamento de  $n$  tarefas a serem realizadas em  $m$  máquinas idênticas e paralelas de processamento em lote com capacidade igual a  $K$ . Cada tarefa  $i$  possui um tamanho  $S_i$ , uma data de entrega  $d_i$  e um tempo de processamento  $p_i$ . Então, um conjunto de tarefas poderá ser sequenciado em um lote se a soma de suas dimensões  $S_i$  for menor do que a capacidade  $K$  da máquina. Deve-se ater ao fato de que uma mesma tarefa não poderá ser sequenciada em mais de um lote e em mais de uma máquina. Considera-se a disponibilidade de  $V_t$  veículos para realizar as entregas em cada momento  $T_t$  ( $1 \leq t \leq z$ ), considerando-se  $T_1 < T_2 < \dots < T_z$ . Os veículos possuem capacidade igual a  $C$ , podendo transportar um número de tarefas tal que o somatório de seus tamanhos seja menor ou igual à  $C$ . Deve-se ater ao fato de que uma tarefa não pode estar associada a mais de um veículo e a mais de um momento de entrega.

Cada tarefa  $i$  só será sequenciada em uma máquina e a um veículo, contribuindo com seu valor  $R_i$  no ganho final da empresa, se o prazo de entrega  $d_i$  for atendido. O prazo de entrega é atendido se o tempo de conclusão da tarefa  $i$  for menor do que o momento  $T_t$  ao qual estiver associada e  $T_t$  menor do que o prazo de entrega  $d_i$ . O tempo de conclusão da tarefa  $i$  é dado pelo tempo de processamento do lote em que esteja sequenciada mais o somatório do tempo de processamento dos lotes que a antecedem na máquina a qual esteja associada. O objetivo do problema é definir o sequenciamento das tarefas nas máquinas e nos veículos que maximize o ganho final da empresa.

A formulação de programação linear inteira mista (PLIM) proposta neste trabalho para o problema teve como base a formulação proposta por Cheng et al. [2015] para o problema de sequenciamento em uma única máquina de processamento em lotes com entrega. A seguir será apresentada a PLIM proposta.

Variáveis de decisão:

- $x_{ik}^l$ : variável binária que assume 1 se a tarefa  $i$  pertence ao lote  $k$  na máquina  $l$ , 0 caso contrário.
- $y_{it}^v$ : variável binária que assume 1 se a tarefa  $i$  é associada ao veículo  $v$  pertencente ao intervalo  $t$ , 0 caso contrário.



- $w_k^l$ : variável contínua que indica o tempo de processamento do lote  $k$  na máquina  $l$ .
- $o_k^l$ : variável contínua que indica o momento em que o lote  $k$  termina de ser processado na máquina  $l$ .
- $u_i$ : variável contínua que indica o momento em que a tarefa  $i$  termina de ser processada.

Utilizando essas variáveis é possível descrever a seguinte formulação PLIM:

$$\max f = \sum_{i=1}^n \sum_{t=1}^z \sum_{v=1}^{V_t} R_i y_{it}^v \quad (1)$$

$$\sum_{k=1}^n \sum_{l=1}^m x_{ik}^l \leq 1 \quad \forall i = 1..n \quad (2)$$

$$\sum_{t=1}^z \sum_{v=1}^{V_t} y_{it}^v \leq 1 \quad \forall i = 1..n \quad (3)$$

$$\sum_{i=1}^n S_i x_{ik}^l \leq K \quad \forall k = 1..n, l = 1..m \quad (4)$$

$$\sum_{i=1}^n S_i y_{it}^v \leq C \quad \forall t = 1..z, v = 1..V_t \quad (5)$$

$$w_k^l \geq p_i x_{ik}^l \quad \forall i = 1..n, k = 1..n, l = 1..m \quad (6)$$

$$o_1^l = w_1^l \quad \forall l = 1..m \quad (7)$$

$$o_k^l = o_{k-1}^l + w_k^l \quad \forall k = 2..n, l = 1..m \quad (8)$$

$$u_i \geq o_k^l - M(1 - x_{ik}^l) \quad \forall i = 1..n, k = 1..n, l = 1..m \quad (9)$$

$$\sum_{t=1}^z \sum_{v=1}^{V_t} y_{it}^v T_t \leq d_i \quad \forall i = 1..n \quad (10)$$

$$\sum_{k=1}^n \sum_{l=1}^m x_{ik}^l \geq \sum_{t=1}^z \sum_{v=1}^{V_t} y_{it}^v \quad \forall i = 1..n \quad (11)$$

$$u_i \leq \sum_{t=1}^z \sum_{v=1}^{V_t} y_{it}^v T_t \quad \forall i = 1..n \quad (12)$$

$$x_{ik}^l \in \{0, 1\} \quad \forall i = 1..n, k = 1..n, l = 1..m \quad (13)$$

$$y_{it}^v \in \{0, 1\} \quad \forall i = 1..n, t = 1..z, v = 1..V_t \quad (14)$$

$$u_i \geq 0 \quad \forall i = 1..n \quad (15)$$

$$o_k^l \geq 0 \quad \forall k = 1..n, l = 1..m \quad (16)$$

$$w_k^l \geq 0 \quad \forall k = 1..n, l = 1..m \quad (17)$$

A função objetivo, expressa pela equação (1), trata-se da maximização do ganho obtido pelas tarefas que foram sequenciadas. A equação (2) assegura que cada tarefa é associada no máximo a um lote em uma única máquina. A restrição (3) garante que cada tarefa só pode estar associada no máximo a um momento de entrega. As restrições (4) e (5) impedem, respectivamente, que a capacidade das máquinas e a capacidade de entrega sejam extrapoladas. O tempo necessário para produzir um lote completo é igual ao tempo de maior processamento das tarefas que o compõe, essa restrição é definida em (6). O momento de conclusão de cada lote é calculado pelas equações (7) e (8). A restrição (9) determina o tempo de conclusão de cada tarefa. A restrição (10) determina que cada tarefa deve ser entregue em um momento de entrega menor que seu prazo de entrega. A restrição (11) define que, para uma tarefa ser alocada para uma posição em uma máquina, ela



deverá estar associada a um momento de entrega. A restrição (12) garante que o período em que cada tarefa é concluída deve ser menor do que o momento de entrega ao qual estará associada. E, finalmente, as restrições de (13) até (17) definem o escopo e domínio das variáveis de decisão. O valor de  $M$  é dado pelo somatório do tempo de processamento de todas as tarefas da instância.

A relaxação do modelo foi obtida através da relaxação das variáveis inteiras do problema. Ou seja, uma tarefa poderá ser sequenciada parcialmente em uma máquina e um veículo. Considerando esse fato, poderá ocorrer de uma tarefa poder estar associada a mais de uma máquina e um veículo desde que suas dimensões sejam respeitadas.

### 3. Heurística Proposta

Neste trabalho propõe-se uma heurística construtiva (HC) com geração de limites. A heurística inicia-se com a construção de um limite superior (*Upper Bound* - UB). Com a informação deste UB, uma solução viável é construída (*Lower Bound* - LB). Posteriormente, uma heurística de melhoria (HM) é aplicada na solução viável construída.

A entrada da heurística é dada por uma lista de tarefas candidatas  $L$ , contendo as tarefas a serem sequenciadas. A estrutura (*Saida*) de resposta para a heurística proposta é composta por uma lista de máquinas na qual cada máquina  $m$  é uma lista de lotes e cada lote  $k$  é uma lista de tarefas sequenciadas. Também possui uma lista de momentos de entrega, em que cada momento  $t$  é uma lista de veículos, sendo que cada veículo  $v$  dessa lista contém uma lista de tarefas alocadas ao mesmo. Contém ainda uma lista de tarefas não sequenciadas e o ganho final  $R$  da configuração.

O problema STMPL com entrega tratado neste trabalho é uma variação do trabalho de Li et al. [2015], tendo como diferença os tempos de processamento que, no presente trabalho, variam entre os diversas tarefas. Como o problema descrito por Li et al. [2015] é NP-Difícil, e o problema estudado neste trabalho trata-se de uma generalização do problema proposto pelos autores supracitados, pode-se definir o problema aqui estudado como NP-Difícil. Dessa forma, para gerar um limite superior em tempo polinomial ao problema original, faz-se necessário prover uma relaxação combinatoria. A relaxação proposta inicialmente quebra todas as tarefas em tempo de processamento ( $p_i$ ) unitário, gerando  $p_i$  réplicas de cada tarefa  $i$ . Além disso, o ganho e o tamanho são divididos na mesma proporção entre todas as réplicas. Com esse novo problema, adiciona-se a condição de realizar a quebra das tarefas, e executa-se a relaxação do método *heu2* proposto por Li et al. [2015]. Li et al. [2015] prova que o método *heu2* encontra o ótimo.

Na sequência, a heurística construtiva (HC) utiliza a solução relaxada como entrada para gerar uma solução inicial viável. Para geração da solução viável, inicialmente percorre-se todas as réplicas alocadas a cada veículo e a cada momento, procura-se a última réplica de cada tarefa alocada. Este momento será associado a um parâmetro definido como entrega da tarefa não particionada. Definindo todas as entregas associadas as tarefas, constrói-se uma lista  $L$  com as tarefas ordenadas de forma decrescente por data de entrega. Em posse dessa lista, para gerar a solução viável inicial, faz-se o sequenciamento seguindo a ordem da lista  $L$ . O sequenciamento deve ser feito respeitando-se as capacidade  $C$  dos veículos em cada momento  $t$  e a capacidade  $K$  de cada máquina, além de ter que respeitar as restrições determinadas pelo prazo de entrega. Durante o sequenciamento na máquina, em cada lote  $k$ , a primeira tarefa sequenciada definirá o tempo máximo de processamento do lote. Dessa forma, as demais tarefas alocadas em cada lote  $k$  devem possuir um tempo de processamento menor ou igual ao tempo de processamento máximo do lote  $k$ . Caso a tarefa  $i$  não possa ser sequenciada em um lote  $k$  já sequenciado em uma máquina, seja devido à sua dimensão  $s_i$  ou ao seu tempo de processamento  $p_i$ , um novo lote deve ser criado na máquina que tiver menor tempo de conclusão até o momento, respeitando-se o prazo de entrega. Caso não seja possível criar um novo lote, a tarefa não é sequenciada. Lembrando-se que um lote só pode existir se o seu momento de conclusão for menor ou igual ao último momento de entrega disponível. Além de que uma tarefa  $i$  somente poderá ser alocada em um lote  $k$  em uma máquina  $m$  se houver um momento de entrega  $T_t \geq d_i$  e maior do que o tempo de conclusão da tarefa que é igual ao tempo de conclusão do lote no qual fora sequenciado. Ressalta-se que o tempo de conclusão de um lote  $k$  é



dado pelo tempo de processamento do mesmo mais o somatório dos tempos de processamento dos lotes que o antecedem. Nesse momento, uma solução viável é gerada e tem-se também um limite inferior para o problema tratado. O algoritmo proposto pode ser resumido em 3 passos principais:

1. Dado um conjunto de  $n$  tarefas, faz-se  $p_i$  réplicas de processamento unitário de cada tarefa  $i$ , com ganho e tamanho divididos em mesma proporção;
2. Considerando a quebra das tarefas, aplica-se a relaxação do método *heu2*, proposto por Li et al. [2015], para obtenção de um limite superior ao problema (UB);
3. Dada a solução inviável obtida pelo passo 2, como UB, realiza-se a viabilização da solução gerando uma primeira solução viável.

A HC proposta fornece simultaneamente um limite superior (UB) obtido pela heurística *heu2* e um inferior (LB) dado pela solução inicial obtida pela heurística construtiva proposta acima.

### 3.1. Heurística de Melhoria (HM)

A heurística de melhoria HM tem como dado de entrada a solução viável gerada pela HC e a lista  $L$ . A HM é baseada na meta-heurística *Variable Neighborhood Descent* (VND) (Mladenović e Hansen [1997], Hansen e Mladenović [2001] e Gendreau e Potvin [2010]). Foram utilizadas três vizinhanças V1, V2 e V3 na implementação do VND. Para todas as estruturas de vizinhanças considerou-se a estratégia de melhor vizinho (*Best Improvement*), que trata-se do caminhamento em toda a vizinhança.

A construção da vizinhança V1 tem como base para a geração de duas listas,  $L1$  e  $L2$ . A lista  $L1$  recebe as tarefas sequenciadas e a lista  $L2$  recebe as tarefas não sequenciadas, na solução dada como inicial. Faz-se uma troca entre duas tarefas de  $L1$  gerando uma nova lista  $L1'$ . Na sequência constrói-se a lista de candidatos  $L''$ , concatenando  $L1'$  e  $L2$  para então executar HC para que se tenha a nova solução.

A vizinhança V2 utiliza a mesma estrutura de V1. Diferentemente de V1, em V2 troca-se uma tarefa de  $L1$  com outra de  $L2$ , gerando duas novas listas  $L1'$  e  $L2'$ . Então na sequência constrói-se a lista de candidatos  $L''$ , concatenando  $L1'$  e  $L2'$  para então executar a HC e assim obter uma nova solução viável.

No caso da vizinhança V3, faz-se a troca aleatória na ordem de duas tarefas da lista de candidatos  $L$ . Na sequência utiliza-se a nova lista como entrada para a construção de uma nova solução pela heurística construtiva, tal como utilizado na heurística proposta.

## 4. Experimentos Computacionais

A formulação de programação linear inteira mista (PLIM) proposta foi implementada via ILOG Concert Technology C++ e foi executada utilizando o software ILOG CPLEX, versão acadêmica 12.1, com todas as configurações padrões. Já o algoritmo heurístico proposto foi codificado na linguagem C/C++, compilado com GNU g++ versão 4.8.4 e executado de modo sequencial. Todos os experimentos foram realizados em um computador Opteron(TM) 6272 CPU @ 1400MHz, com 96GB de memória RAM e sistema operacional Ubuntu 14.04.5 LTS, 64 bits.

### 4.1. Geração de Instâncias

Para realização dos experimentos utilizou-se 360 instâncias. As instâncias foram obtidas a partir de 5 replicações de um conjunto base de 72 possibilidades de intervalos de valores dos quais os parâmetros são sorteados. A geração das instâncias foi baseada em Li et al. [2015].

Para definir as possíveis combinações de parâmetros que deram origem ao conjunto base de instâncias definiu-se os índices  $a$ ,  $b$  e  $c$ . Sendo assim, a configuração que constitui uma dada instância é dada pela combinação dos intervalos de valores e os parâmetros associados aos índices  $a$ ,  $b$ ,  $c$ .

Na Figura 1, em se tratando do sequenciamento nos momentos de entrega tem-se:  $z$  = número de momentos de entrega;  $T_z$  = momento de entrega de maior valor;  $maxV_{eic}$  = número





máximo de veículos disponíveis por momento de entrega;  $minV_{ec}$  = número mínimo de veículos disponíveis para cada momento  $t$  de forma que  $minV_{ec} \leq V_t \leq maxV_{ec}$ ;  $minT_{em}$  e  $maxT_{em}$  = valor mínimo e máximo respectivamente, que cada momento de entrega  $T_t$  tal que  $1 \leq t \leq z$  pode assumir. Quanto aos demais parâmetros tem-se:  $m$  = número de máquinas;  $n$  = número de tarefas que assumem valores fixos a depender do conjunto base.

A associação entre índices, intervalos de valores e parâmetros está representada na Figura 1. Por exemplo se  $a = 0$ :  $n = 20$  e  $z = 2$ ; se  $b = 0$ :  $m = 3$ ; se  $c = 0$ :  $T_z = 30$ ,  $maxV_{ec} = 3$ ,  $minT_{em} = 5$  e  $maxT_{em} = 28$ .

índice a	n	z	índice b	m	índice c	Tz	maxVeic	MinTem	MaxTem
0	20	2	0	3	0	30	3	5	28
1	30	2	1	4	1	45	3	6	39
2	40	3	2	5	2	30	7	5	28
3	60	4			3	45	7	6	39
4	80	5							
5	100	6							

Tamanho da tarefa		Ganho da Tarefa		Capacidade da Máquina	Prazo de entrega	
min	max	Min	Max	40	min	max
1	40	1	100			T1
					Tempo de Processamento	
					Min	Max
					1	6

Figura 1: Esquema de geração das instância.

Para cada tarefa  $i$ , os parâmetros ganho ( $R_i$ ), tamanho da tarefa ( $s_i$ ), tempo de processamento ( $p_i$ ) e prazo de entrega ( $d_i$ ), são sorteados dentro de um intervalo  $max$  e  $min$  fixo independente do conjunto de instâncias. A capacidade da máquina é fixa de acordo com o valor em tabela e a capacidade do veículo é dada pela equação (18).

$$Capacidade\_do\_Veículo = \frac{0,8 \times (\sum_{i=0}^n s_i)}{\sum_{t=0}^{z-1} v_t} \quad (18)$$

Todos os valores obtidos para os parâmetros pertencem a uma distribuição uniforme tendo como limites os respectivos intervalos estipulados. Para o problema considerou-se os parâmetros tempo de processamento, prazo de entrega e intervalos de entrega assumindo valores inteiros e os demais valores reais.

Para facilitar a explicação dos resultados cada instância representada por  $abc$  receberá um ID de forma que a primeira 000 receberá ID = 1, a segunda 001 receberá ID = 2 até a ultima instância, variando primeiro  $c$ , depois  $b$  e depois  $a$ .

#### 4.2. Resultados Computacionais

Para avaliar os resultados, calcula-se a relação entre os valores das diversas soluções encontradas, tanto pelo modelo como pelas fases da heurística proposta. As diversas relações são chamadas de GAP, conforme definidos a seguir.

Para o cálculo do  $GAP_{ModeloRelaxado}$  considerou-se o modelo e sua relaxação comparando-se o valor obtido para a função objetivo (FO) em cada caso. A equação (19) foi utilizada para o cálculo do  $GAP_{ModeloRelaxado}$ .

$$GAP_{ModeloRelaxado} = \frac{FO_{ModeloRelaxado} - FO_{Modelo}}{FO_{Modelo}} \quad (19)$$

Para o cálculo do  $GAP_{UB}$  (Upper Bound) comparou-se o valor da função objetivo encontrada através da heurística relaxada com a função objetivo (FO) obtida pelo modelo. A equação (20) foi utilizada para o cálculo do  $GAP_{UB}$

$$GAP_{UB} = \frac{FO_{UB} - FO_{Modelo}}{FO_{Modelo}} \quad (20)$$



O valor do  $GAP_{Modelo}$  é fornecido pelo próprio CPLEX, quando executa-se o modelo de programação linear inteira mista. O  $GAP_{HC}$  faz a relação entre valor da função objetivo (FO) obtido para o UB e o obtido conforme a HC, calculada pela equação (21).

$$GAP_{HC} = \frac{FO_{UB} - FO_{HC}}{FO_{UB}} \quad (21)$$

O  $GAP_{HM}$  é obtido pela diferença entre o valor de FO do UB com o valor obtido de FO pela HM. Essa diferença é calculada pela equação (22).

$$GAP_{HM} = \frac{FO_{UB} - FO_{HM}}{FO_{UB}} \quad (22)$$

A Tabela 1 apresenta os resultados alcançados pelas estratégias de solução mencionadas anteriormente. Cada linha da tabela informa o resultado médio obtido para as 5 instâncias de cada conjunto base. As duas primeiras colunas (ID e *abc*) da tabela identificam os conjuntos de instâncias e as demais colunas informam o valor médio do GAP e do tempo médio obtidos através da execução da formulação de programação linear inteira mista (PLIM)<sup>1</sup>, da heurística proposta para determinar um valor de limite superior (UB) para a solução, da heurística construtiva (HC) e da heurística de melhoria (HM).

Em relação à formulação do modelo, observou-se que dentre as 360 instâncias testadas conseguiu-se alcançar o valor ótimo em 53% dos casos. Dentre os 47% cujo valor encontrado para a função objetivo não foi o ótimo, em 4 instâncias não foi possível encontrar uma solução viável pelo modelo matemático considerando o tempo limite para o modelo de 1000 segundos. Duas das instâncias sem solução pertencem ao conjunto base 72, uma ao conjunto base 71 e uma ao conjunto base 69. Esses elementos não foram incluídos no cálculo dos valores de GAP, mas atestam o fato de o modelo matemático não conseguir obter nenhuma solução viável em algumas instâncias. Este fato, associado à porcentagem de instâncias para as quais não se obteve o valor ótimo para a função objetivo, mostra a necessidade do uso de métodos aproximados eficientes.

Os resultados descritos na Tabela 1 foram utilizados na construção dos gráficos apresentados nas Figuras 2 a 5, com o objetivo de destacar alguns aspectos observados neste trabalho. A Figura 2 mostra as curvas de GAP para o Upper Bound ( $GAP_{UB}$ ) e modelo relaxado ( $GAP_{ModeloRelaxado}$ ). O fato de as curvas estarem praticamente sobrepostas comprova a eficácia da heurística relaxada na obtenção de Upper Bound para o problema.

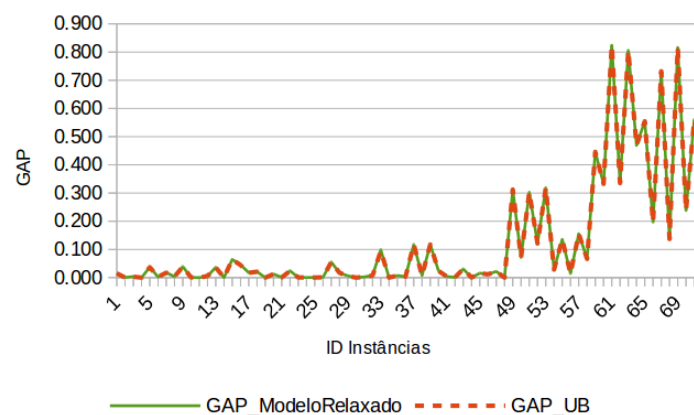


Figura 2: Comparação entre o GAP médio apresentado pelo modelo relaxado e o apresentado pelo UB

Quanto ao tempo, pode-se perceber pela Figura 3 que o tempo para se alcançar um valor de FO na heurística relaxada é consideravelmente menor em relação ao tempo necessário para que

<sup>1</sup>As colunas GAP-r ( $GAP_{ModeloRelaxado}$ ) e t-r(s) informam os resultados obtidos pela execução da PLIM relaxada.





o modelo relaxado encontre um valor de FO, comprovando a eficiência da heurística em relação a relaxação do modelo.

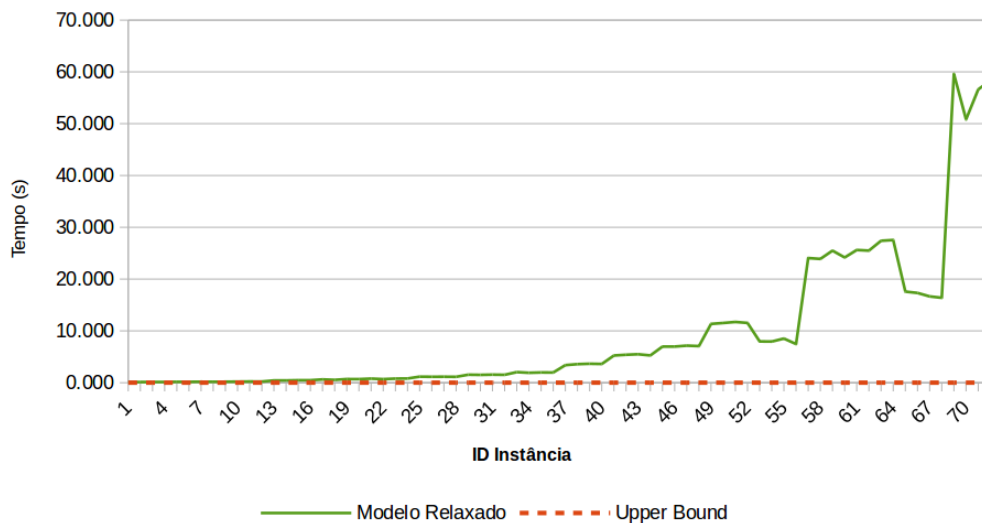


Figura 3: Comparação entre o tempo de execução da relaxação do problema com o tempo de execução do modelo relaxado.

A Figura 4 apresenta as curvas de GAP médio para o modelo ( $GAP_{Modelo}$ ), para a HC ( $GAP_{HC}$ ) e para HM ( $GAP_{HM}$ ). Pode-se observar que até a instância 37, apesar das oscilações, as instâncias foram resolvidas pelo modelo obtendo soluções com  $GAP_{Modelo}$  próximo ou igual a 0 e em um tempo inferior a 1000 segundos, com valores de tempo inferiores ou pouco maiores que 600 segundos. A partir da instância 37, o modelo apresenta mais dificuldade de conseguir um valor de FO com GAP próximo de zero, necessitando de um tempo de execução próximo ao limite máximo de 1000 segundos. Então pode-se concluir que as instâncias tornam-se mais difíceis de resolver à medida que aumenta-se o número de máquinas, tarefas e veículos, além do tamanho do intervalo de onde são sorteados os momentos de entrega. Todas as análises foram realizadas baseadas nos valores médios de GAP e tempo para cada uma das 72 combinações, o que explica o fato de algumas instâncias apresentarem um valor de FO com um GAP diferente de 0 em um tempo menor que o limite de execução do modelo. Dentro de um mesmo conjunto de instâncias, devido ao fato de alguns parâmetros serem sorteados aleatoriamente, encontram-se instâncias com complexidade diferentes que deslocam o valor da média para mais ou para menos.

Ao comparar-se os valores de GAP obtidos pelo modelo, pela HC e pela HM, observa-se que nos casos das instâncias mais fáceis os melhores resultados oscilam entre o modelo e a HM, sendo a HC a que apresenta os piores resultados. Quando analisadas as instâncias com o ID a partir de 37, observa-se que HM ganha na maioria dos casos. A partir da instância 50, a HM ganha em todos os casos. Quanto à HC, não apresentam valores de GAP menores que o modelo até a instância 57. Mesmo assim o uso de HC utilizada conjuntamente com a HM apresentou-se bastante eficaz, visto que a heurística construtiva é utilizada para gerar a solução inicial viável necessária para a heurística de melhoria e internamente na própria heurística de melhoria.

Analisando-se a Figura 5 observa-se que o tempo de execução de HC mais HM apresenta menores valores em relação a execução do modelo. Desta forma, apesar de HC perder, em relação ao GAP, para o modelo, pode-se considerá-la eficiente como solução inicial para a heurística de melhoria, visto que o conjunto (HC-HM) consegue alcançar resultados próximo ou melhor que o modelo com um tempo de execução bem mais baixo. Além disso o tempo de execução de HC para gerar a solução inicial representa 0,1% do tempo total do conjunto HC mais HM.

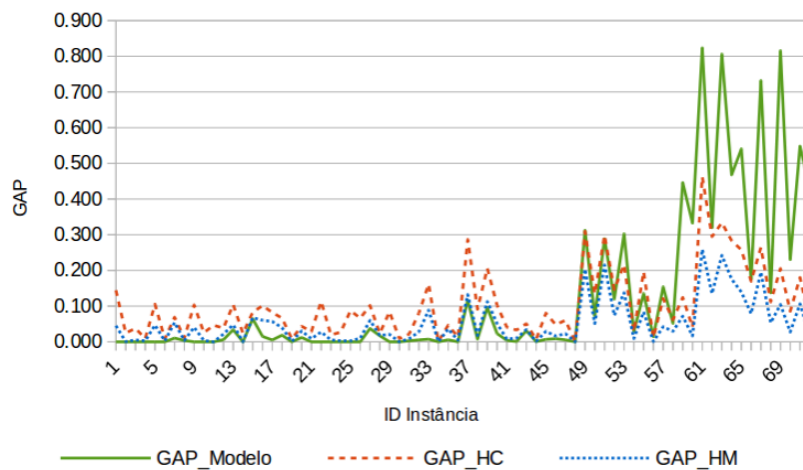


Figura 4: Comparação entre os GAPs médios calculados para HC, HM e o obtido pelo CPLEX para o modelo

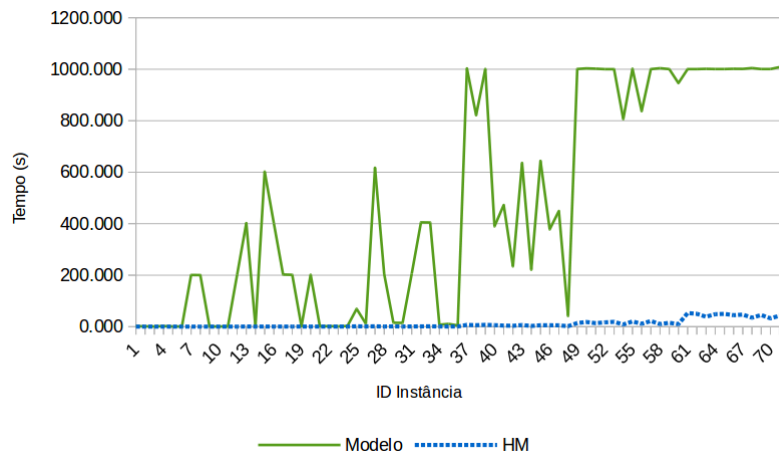


Figura 5: Comparação entre tempo de execução do modelo com o tempo de execução de HM.

## 5. Conclusão

Este trabalho apresentou o problema de sequenciamento de tarefas em máquinas paralelas idênticas de processamento em lotes com entrega, considerando os tempos de processamento distintos das tarefas. Para resolver o problema foi proposta uma formulação de programação linear inteira mista, a utilização de uma heurística da literatura para a obtenção do UB, e o uso da mesma heurística apoiada por uma heurística construtiva para a obtenção de uma solução inicial. Com o objetivo de melhorar a solução gerada pela heurística construtiva foi proposta uma heurística baseada na metaheurística VND, na qual utilizou-se três estruturas de vizinhança.

Com este trabalho foi possível validar o uso da heurística da literatura para definir o valor de UB para o problema aqui estudado, comparando-se o tempo de execução e o  $GAP_{UB}$  da heurística com o  $GAP_{ModeloRelaxado}$ . Foi possível notar uma maior eficiência da heurística proposta para o conjunto de instâncias utilizado, visto que conseguiu-se obter soluções boas em um tempo de execução reduzido se comparado à formulação de programação linear inteira mista.

Como trabalhos futuros, pretende-se explorar novos e mais complexos conjuntos de instâncias e a aplicação de outra metaheurística, como a *Iterated Local Search* (ILS), a fim de encontrar resultados com baixos valores de GAP e para conjunto de instancias mais abrangentes para o problema estudado no presente trabalho.



Tabela 1: Tabela de Resultados

Instâncias		PLIM				UB		HC		HM	
ID	abc	GAP	t(s)	GAP-r	t-r(s)	GAP	t(s)	GAP	t(s)	GAP	t(s)
1	000	0.000	1.856	0.016	0.094	0.015	0.001	0.144	0.001	0.045	0.111
2	001	0.000	1.052	0.000	0.125	0.000	0.001	0.024	0.001	0.000	0.065
3	002	0.000	0.635	0.004	0.133	0.003	0.001	0.038	0.001	0.005	0.087
4	003	0.000	2.160	0.000	0.137	0.000	0.001	0.009	0.001	0.002	0.088
5	010	0.000	0.683	0.037	0.166	0.037	0.001	0.105	0.001	0.047	0.081
6	011	0.000	0.606	0.002	0.161	0.002	0.001	0.008	0.001	0.002	0.071
7	012	0.010	200.492	0.018	0.187	0.018	0.001	0.069	0.001	0.054	0.083
8	013	0.004	200.443	0.004	0.173	0.004	0.001	0.010	0.001	0.004	0.072
9	020	0.000	0.595	0.039	0.186	0.038	0.001	0.104	0.001	0.041	0.102
10	021	0.000	0.668	0.001	0.211	0.001	0.001	0.025	0.001	0.004	0.084
11	022	0.000	0.346	0.000	0.227	0.000	0.001	0.046	0.001	0.000	0.079
12	023	0.006	201.068	0.006	0.218	0.006	0.001	0.039	0.001	0.022	0.071
13	100	0.034	402.074	0.036	0.434	0.036	0.001	0.104	0.001	0.048	0.438
14	101	0.000	2.214	0.000	0.430	0.000	0.001	0.024	0.002	0.000	0.300
15	102	0.064	602.042	0.064	0.473	0.064	0.001	0.083	0.002	0.068	0.324
16	103	0.015	401.095	0.046	0.474	0.044	0.001	0.101	0.002	0.060	0.344
17	110	0.005	202.460	0.017	0.632	0.017	0.001	0.082	0.002	0.057	0.375
18	111	0.019	201.418	0.021	0.545	0.021	0.001	0.066	0.001	0.039	0.348
19	112	0.000	1.790	0.000	0.696	0.000	0.001	0.008	0.002	0.001	0.249
20	113	0.012	201.447	0.013	0.683	0.013	0.001	0.044	0.002	0.030	0.274
21	120	0.000	1.801	0.001	0.781	0.001	0.001	0.029	0.001	0.009	0.367
22	121	0.000	1.827	0.024	0.680	0.024	0.001	0.113	0.001	0.028	0.309
23	122	0.000	1.328	0.001	0.789	0.001	0.001	0.019	0.002	0.005	0.235
24	123	0.000	2.409	0.001	0.812	0.001	0.001	0.027	0.002	0.003	0.294
25	200	0.000	69.018	0.000	1.162	0.000	0.001	0.087	0.002	0.003	1.238
26	201	0.000	11.500	0.001	1.137	0.001	0.001	0.066	0.002	0.010	0.939
27	202	0.037	617.134	0.055	1.154	0.055	0.001	0.102	0.002	0.060	1.249
28	203	0.017	204.015	0.017	1.131	0.017	0.001	0.023	0.002	0.018	0.504
29	210	0.000	15.243	0.006	1.540	0.006	0.001	0.084	0.002	0.021	1.069
30	211	0.000	14.639	0.000	1.512	0.000	0.001	0.009	0.002	0.000	0.595
31	212	0.003	206.070	0.003	1.556	0.003	0.001	0.022	0.002	0.009	0.782
32	213	0.005	405.143	0.007	1.520	0.007	0.001	0.080	0.002	0.028	1.053
33	220	0.007	404.472	0.099	2.037	0.094	0.001	0.162	0.002	0.089	1.252
34	221	0.000	7.351	0.000	1.908	0.000	0.001	0.001	0.002	0.000	0.696
35	222	0.006	10.688	0.007	1.978	0.007	0.001	0.046	0.002	0.036	0.589
36	223	0.000	4.795	0.003	1.961	0.003	0.001	0.013	0.002	0.004	0.620
37	300	0.117	1000.000	0.118	3.385	0.117	0.002	0.287	0.003	0.132	6.802
38	301	0.007	821.560	0.008	3.577	0.007	0.002	0.091	0.003	0.018	5.666
39	302	0.096	1000.000	0.117	3.660	0.117	0.002	0.206	0.003	0.113	7.101
40	303	0.023	389.793	0.024	3.607	0.024	0.002	0.102	0.003	0.051	5.851
41	310	0.004	472.561	0.004	5.251	0.004	0.002	0.041	0.003	0.008	4.231
42	311	0.001	234.323	0.001	5.398	0.001	0.002	0.033	0.003	0.010	3.339
43	312	0.031	635.837	0.031	5.492	0.031	0.002	0.051	0.003	0.034	6.310
44	313	0.000	221.466	0.000	5.256	0.000	0.002	0.007	0.003	0.001	2.844
45	320	0.007	643.708	0.016	6.977	0.016	0.002	0.080	0.003	0.029	5.348
46	321	0.009	377.930	0.012	6.976	0.012	0.002	0.048	0.003	0.016	5.336
47	322	0.005	449.173	0.022	7.148	0.022	0.002	0.060	0.003	0.023	5.140
48	323	0.000	41.751	0.000	7.062	0.000	0.002	0.000	0.003	0.000	1.762
49	400	0.312	1000.000	0.312	11.353	0.312	0.002	0.309	0.004	0.204	14.003
50	401	0.071	1000.000	0.072	11.531	0.072	0.002	0.135	0.004	0.051	17.967
51	402	0.289	1000.000	0.303	11.730	0.302	0.002	0.298	0.004	0.215	13.659
52	403	0.120	1000.000	0.121	11.534	0.121	0.002	0.146	0.005	0.074	16.583
53	410	0.302	1000.000	0.318	7.969	0.318	0.002	0.216	0.004	0.138	18.658
54	411	0.029	806.528	0.029	7.944	0.029	0.002	0.028	0.005	0.009	8.566
55	412	0.135	1000.000	0.135	8.519	0.135	0.002	0.197	0.004	0.084	20.104
56	413	0.015	837.368	0.015	7.461	0.015	0.002	0.012	0.005	0.001	11.228
57	420	0.154	1000.000	0.156	24.056	0.156	0.002	0.121	0.004	0.042	21.773
58	421	0.052	1000.000	0.067	23.891	0.066	0.002	0.062	0.004	0.029	9.912
59	422	0.446	1000.000	0.446	25.496	0.446	0.002	0.124	0.005	0.072	14.937
60	423	0.333	946.087	0.333	24.168	0.333	0.002	0.041	0.005	0.016	9.712
61	500	0.823	1000.000	0.824	25.620	0.824	0.003	0.462	0.005	0.259	52.769
62	501	0.320	1000.000	0.335	25.490	0.335	0.003	0.295	0.006	0.135	49.933
63	502	0.806	1000.000	0.806	27.393	0.806	0.003	0.334	0.006	0.242	38.414
64	503	0.468	1000.000	0.469	27.546	0.469	0.003	0.283	0.006	0.176	48.209
65	510	0.541	1000.000	0.554	17.576	0.554	0.003	0.258	0.006	0.140	49.668
66	511	0.174	1000.000	0.197	17.324	0.197	0.003	0.170	0.006	0.079	44.332
67	512	0.732	1000.000	0.732	16.652	0.732	0.003	0.265	0.006	0.195	46.943
68	513	0.137	1000.000	0.137	16.385	0.137	0.003	0.112	0.006	0.053	34.852
69	520	0.816	1000.000	0.816	59.598	0.816	0.003	0.205	0.006	0.105	45.291
70	521	0.230	1000.000	0.239	50.825	0.239	0.003	0.086	0.006	0.027	32.385
71	522	0.548	1000.000	0.560	56.603	0.560	0.003	0.180	0.006	0.106	42.311
72	523	0.432	1000.000	0.432	58.365	0.431	0.003	0.057	0.006	0.017	30.837



## Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por estimular o desenvolvimento deste trabalho através da bolsa de estudos e também à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo apoio financeiro para a participação no evento.

## Referências

- Agnētis, A., Aloulou, M. A., e Fu, L.-L. (2014). Coordination of production and interstage batch delivery with outsourced distribution. *European Journal of Operational Research*, 238(1):130–142.
- Chang, Y.-C. e Lee, C.-Y. (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research*, 158(2):470–487.
- Cheng, B.-Y., Leung, J. Y.-T., e Li, K. (2015). Integrated scheduling of production and distribution to minimize total cost using an improved ant colony optimization method. *Computers & Industrial Engineering*, 83:217–225.
- Damodaran, P. e Chang, P.-Y. (2008). Heuristics to minimize makespan of parallel batch processing machines. *The International Journal of Advanced Manufacturing Technology*, 37(9):1005–1013.
- Fuchigami, H. Y. e Rangel, S. (2014). Uma análise de estudos de casos em sequenciamento da produção. In *Anais do XLVI SBPO*, p. 159–170, Salvador, Bahia. SOBRAPO.
- Gendreau, M. e Potvin, J.-Y. (2010). *Handbook of metaheuristics*, volume 2. Springer.
- Hansen, P. e Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467.
- Leung, J. Y.-T. e Chen, Z.-L. (2013). Integrated production and distribution with fixed delivery departure dates. *Operations Research Letters*, 41(3):290–293.
- Li, K., Jia, Z.-h., e Leung, J. Y.-T. (2015). Integrated production and delivery on parallel batching machines. *European Journal of Operational Research*, 247(3):755–763.
- Mathirajan, M. e Sivakumar, A. I. (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *The International Journal of Advanced Manufacturing Technology Research*, 29(9-10):990–1001.
- Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & Operations research*, 24(11):1097–1100.