

## **A Um modelo de otimização para o problema de dimensionamento e programação de lotes de produção em máquina única**

**Cezaraugusto Gomes Scalcon**

Programa de Pós-graduação em Engenharia de Produção – Universidade Federal de Santa Maria  
Av. Roraima nº 1000 – Cidade Universitária – Bairro Camobi – Santa Maria – RS – 97105 900  
[cezarscalcon@gmail.com](mailto:cezarscalcon@gmail.com)

**Olinto César Bassi de Araújo**

Colégio Técnico Industrial de Santa Maria – Universidade Federal de Santa Maria  
Av. Roraima nº 1000 – Cidade Universitária – Bairro Camobi – Santa Maria – RS – 97105 900  
[olinto@ctism.ufsm.br](mailto:olinto@ctism.ufsm.br)

**Felipe Martins Müller**

Programa de Pós-graduação em Engenharia de Produção – Universidade Federal de Santa Maria  
Av. Roraima nº 1000 – Cidade Universitária – Bairro Camobi – Santa Maria – RS – 97105 900  
[felipe@inf.ufsm.br](mailto:felipe@inf.ufsm.br)

**Marcia Helena Costa Fampa**

Instituto de Matemática - Universidade Federal do Rio de Janeiro  
Av. Athos da Silveira Ramos 149, Centro de Tecnologia – Bloco C  
[fampa@cos.ufrj.br](mailto:fampa@cos.ufrj.br)

**Resumo:** Neste trabalho é proposta uma formulação de programação inteira 0-1 para modelar o problema de programação e dimensionamento de lotes de produção em máquina única. Este problema considera um conjunto de tarefas com diferentes tamanhos e tempos de processamento que devem ser agrupadas em lotes de acordo com a capacidade limitada da máquina. O tempo de processamento de um lote é determinado pelo maior tempo de processamento dentre todas as tarefas que compõem o lote. A medida de desempenho é o tempo total necessário para processar todas as tarefas (*makespan*). A formulação apresentada é mais forte, ou seja, mais próxima da formulação ideal do que aquelas propostas na literatura. Experimentos computacionais demonstram que o modelo é consistente e representa adequadamente o problema tratado.

**PALAVRAS CHAVE:** programação inteira; dimensionamento de lotes; programação da produção.

**Área principal:** Otimização Combinatória, Programação Matemática, PO na Indústria.

**Abstract:** In this paper we proposed 0-1 integer programming formulation to model single batch processing machine. This problem deals with a set of jobs with non-identical sizes and processing times that has to be grouped to form batches according to the limited capacity of the machine. The processing time of a batch is the longest processing time of all jobs in the batch. The performance measure is the total time required to process all jobs (*makespan*). The formulation presented strengthens the model, i.e., it is closer to the optimal formulation than those proposed in the literature. Computational experiments demonstrate that the model is consistent and adequately represents the problem addressed.

**KEYWORDS:** integer programming; lot-sizing; production planning.

**Main area:** Combinatorial Optimization, Mathematical Programming, OR in Industry

## 1. Introdução

Produzir um número crescente de produtos industriais, a fim de atender a demanda por produtos manufaturados, tem impulsionado o desenvolvimento tecnológico e proporcionado a implementação de processos cada vez mais complexos. Este processo permite que os consumidores possam adquirir produtos e serviços com tecnologia sofisticada a preços acessíveis, massificando o uso de determinados itens, como equipamentos eletrônicos, veículos automotivos entre outros. Para suprir esta necessidade são utilizados processos que executam múltiplas tarefas ou várias operações simultâneas que, atualmente, são considerados parte do sistema produtivo da mesma forma que a mais simples ferramenta manual. De fato, nas últimas décadas ocorreu um aumento significativo no interesse em estudar programação de tarefas em sistemas com elementos de processamento em lote. A motivação evidente para tal reside no barateamento dos custos e no aumento da eficiência da produção (Potts e Kovalyov, 2000).

Casos típicos de processamento de tarefas em lote ocorrem na fabricação de circuitos integrados e processadores (Mathirajan e Sivakumar, 2006). A fabricação destes componentes exige uma etapa denominada *burn-in* (teste de confiabilidade), realizada em câmaras térmicas onde os circuitos integrados são submetidos a uma temperatura constante de aproximadamente 120°C. Como a etapa *burn-in* geralmente demanda bem mais tempo quando comparada com outros testes de operação, para agilizar o processo de fabricação são utilizados fornos de lote. Como diversos componentes apresentam diferentes tamanhos e exigem diferentes tempos mínimos de exposição ao calor, a forma como são dimensionados os lotes afeta diretamente a taxa de produção.

A complexidade do problema de programação e dimensionamento de lotes de produção em máquina única (NP-difícil) foi objeto de estudo em Uzsoy (1994), considerando, separadamente, como critério de otimização o tempo total de conclusão das tarefas (*total completion time*) e o *makespan*. O autor ainda propõe heurísticas e um algoritmo *branch and bound*, sendo esta última técnica aplicada para o caso com critério de tempo total de conclusão das tarefas. Heurísticas para estes problemas também são propostas em Dupont e Jolai Ghazvini (1998) e Jolai Ghazvini e Dupont (1998).

Considerando o critério *makespan*, abordagens baseadas em metaheurísticas podem ser encontradas em Melouk et al. (2004) e Damodarn et al. (2006) e algoritmos exatos em Dupont and Dhaenens-Flipo (2002) e Parsa et al. (2010).

Este trabalho propõe um modelo matemático para o problema de programação e dimensionamento de lotes de produção em máquina única com o objetivo de minimizar o *makespan*. A formulação proposta é mais forte do que aquelas encontradas na literatura, no sentido que a relaxação contínua do problema fornece melhores limitantes e não altera a solução ótima do problema original.

As demais seções deste trabalho são organizadas como segue. Na Seção 2 o problema em estudo é formalmente definido. A Seção 3 apresenta e discute formulações de programação inteira 0-1. Resultados computacionais são analisados na Seção 4 e conclusões são relatadas na Seção 5.

## 2. Descrição do problema

O problema de programação e dimensionamento de lotes de produção em máquina única tratado neste trabalho tem como objetivo minimizar o tempo total de produção ( $C_{\max}$ ). Cada tarefa  $j$  a ser processada tem um tempo de processamento  $p_j$  e um tamanho correspondente  $s_j$ ,  $j = 1, 2, \dots, n$ . O tempo de processamento de um lote  $b$ ,  $b = 1, 2, \dots, m$ , é dado pelo maior tempo de processamento do lote, ou seja,  $C_b = \max\{p_j : x_{jb} = 1\}$ . O tamanho total do lote deve ser menor ou igual à capacidade  $S$  da máquina. Todas as tarefas estão disponíveis no momento 0 (zero) e nenhuma atividade pode ser dividida entre os lotes. A partir da definição de um lote, acrescentar ou retirar uma tarefa não é permitido e o processamento do lote não pode ser interrompido depois que ele é iniciado.

Dado que cada tarefa pode constituir um lote, será considerado que existem tantos lotes quantas tarefas ( $m = n$ ), no entanto, é importante observar que é possível calcular um limitante inferior para o número de lotes.

A Figura 1 corresponde a uma ilustração para uma solução do problema de programação e dimensionamento de lotes em máquina única. Na representação comprimento corresponde ao tempo de processamento e a altura o consumo de recurso. Em (a) uma solução é representada, composta por

quatro lotes. Uma característica importante a ser considerada é que o problema em questão é altamente simétrico, no sentido que uma mesma solução pode ser construída de diferentes formas. A solução em (b) é composta pelos mesmos quatro lotes que em (a), com a diferença que estão dispostos em uma ordem diferente. De fato, qualquer permutação na ordem dos lotes produz o mesmo valor de *makespan*.

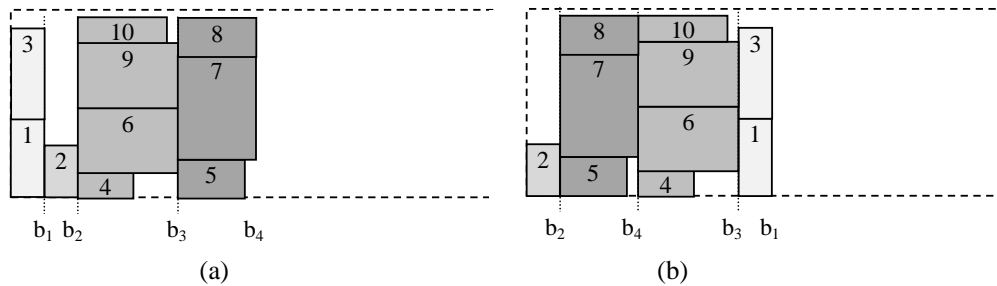


Figura 1. Exemplo de programação de tarefas em lote em máquina única

Para modelar matematicamente esse problema, restrições de quebra de simetrias são importantes para reduzir o espaço de soluções. Na seção a seguir são propostos dois conjuntos de desigualdades válidas e um procedimento de pré-processamento para tratar simetria.

### 3. Modelos matemáticos

O modelo definido a seguir é o que mais comumente aparece na literatura, doravante denominado Modelo 1.

#### Variáveis

$$x_{jb} = \begin{cases} 1 & \text{se a tarefa } j \text{ é processada no lote } b \\ 0 & \text{caso contrário} \end{cases}$$

$$y_b = \begin{cases} 1 & \text{se o lote } b \text{ é utilizado} \\ 0 & \text{caso contrário} \end{cases}$$

$C_b$  : tempo de processamento do lote  $b$

#### Parâmetros

$s_j$  : tamanho da tarefa  $j$

$p_j$  : tempo de processamento da tarefa  $j$

$S$  : capacidade máxima da máquina para processar um lote

### Modelo 1

$$\text{Min } C_{\max} = \sum_{b=1}^m C_b \quad (1)$$

s.a

$$\sum_{b=1}^m x_{jb} = 1 \quad j = 1, 2, \dots, n \quad (2)$$

$$\sum_{j=1}^n s_j x_{jb} \leq S y_b \quad b = 1, \dots, n \quad (3)$$

$$x_{jb} \leq y_b \quad j = 1, 2, \dots, n; b = 1, \dots, n \quad (4)$$

$$p_j x_{jb} \leq C_b \quad j = 1, 2, \dots, n; b = 1, \dots, n \quad (5)$$

$$y_b \in \{0,1\} \quad b = 1, \dots, n \quad (6)$$

$$x_{jb} \in \{0,1\} \quad j = 1, 2, \dots, n; b = 1, \dots, n \quad (7)$$

$$C_b \geq 0 \quad b = 1, \dots, n \quad (8)$$

A função objetivo (1) minimiza o *makespan*,  $C_{\max}$ . O conjunto de restrições (2) garante que uma tarefa  $j$  é atribuída a um único lote  $b$ . O conjunto de restrições (3) garante que a capacidade  $S$  da máquina seja respeitada quando tarefas são atribuídas a um lote. O conjunto de restrições (5) determina o tempo de processamento do lote  $b$ . Os conjuntos de restrições (6), (7) e (8) especificam o tipo das variáveis de decisão. O conjunto de restrições (4) é redundante, uma vez que é uma implicação direta de (3). No entanto, é sabido ser uma desigualdade válida que, quando incluída no modelo, melhora o desempenho de resolvedores baseados em relaxações lineares.

Para tratar a estrutura de simetria ilustrada na seção anterior são propostos dois conjuntos de desigualdades válidas semelhantes àquelas propostas Köhler (2012) para um problema de agrupamento em grafo. O primeiro é formulado como segue.

$$\sum_{b=j}^m x_{jb} = 1 \quad j = 1, 2, \dots, n \quad (9)$$

Esse conjunto de restrições força que cada tarefa seja designada para um lote com índice maior ou igual ao índice da própria tarefa. Desta forma, a tarefa  $n$  só pode ser designada para o lote  $n$ , a tarefa  $n-1$  para os lotes  $n-1$  e  $n$ , e assim por diante.

O segundo conjunto de desigualdades válidas, conforme definido em (10), faz uso do fato que (9) força o índice de um cluster seja igual ao maior índice das tarefas que compõem o cluster. Deste modo, se a tarefa  $k$  não está no lote  $k$ , então o lote  $k$  deve ser considerado vazio.

$$x_{jb} \leq x_{bb} \quad j = 1, 2, \dots, n; b = 1, \dots, n \quad (10)$$

Considerando estes conjuntos de restrições no exemplo da Figura 1, uma solução possível é expressa por  $b_2 = \{2\}$ ,  $b_3 = \{1,3\}$ ,  $b_8 = \{5,7,8\}$ ,  $b_{10} = \{4,6,9,10\}$ , que apresenta o mesmo valor de *makespan*.

O modelo que inclui (9) e (10) é denominado Modelo 2.

### Modelo 2

$$\text{Min} \quad (1)$$

s.a

$$(3), (4), (5), (6), (7), (8)$$

$$\sum_{b=j}^m x_{jb} = 1 \quad j = 1, 2, \dots, n$$

$$x_{jb} \leq x_{bb} \quad j = 1, 2, \dots, n ; b = 1, \dots, n$$

O conjunto de restrições (2) não é utilizado, uma vez estas restrições são dominadas pelas restrições do conjunto (9).

Ainda, é possível lançar mão de uma rotina de pré-processamento para efetuar uma ordenação das tarefas de acordo com tamanho (ordem crescente). Esse procedimento simples, associado a (9) e (10), faz com que as tarefas com maior probabilidade de determinarem o *makespan* de cada lote tenham menor possibilidade de designação para diferentes lotes. O modelo com pré-processamento é denominado Modelo 3.

### 4. RESULTADOS COMPUTACIONAIS

Para os testes computacionais foi utilizado um computador com processador Intel QuadCore Xeon X3360 2.83 GHz e o otimizador CPLEX 12.1, com configuração padrão, foi utilizado para resolver as três formulações.

O conjunto de instâncias teste foi gerado de acordo com Parsa et al. (2010). Quatro fatores são considerados para gerar as instâncias: número de tarefas, intervalo do tamanho das tarefas, intervalo do tempo de processamento e capacidade da máquina. O tempo de processamento e o tamanho das tarefas são gerados aleatoriamente a partir de uma distribuição uniforme, de acordo com os valores dos fatores apresentados na Tabela 1.

Tabela 1 – Valores dos fatores considerados para gerar instâncias aleatórias

Fatores	Valores
Número de tarefas ( $n$ )	20, 40, 60, 80, 100
Tamanho das tarefas ( $s_j$ )	[1,10] ; [4,8] ; [1,5] ; [2,4]
Tempo de processamento dras tarefas ( $p_j$ )	[1,10] ; [1,5]
Capacidade da máquina ( $S$ )	5, 10

A Tabela 2 apresenta seis diferentes tipos de problemas combinando os quatro intervalos de tamanho de tarefas e os dois níveis de capacidade para a máquina. Para cada tipo, os cinco diferentes números de tarefas e dois níveis de tempo de processamento são utilizados, possibilitando  $6 \times 2 \times 5 = 60$  possíveis combinações. Para cada combinação foram geradas 10 instâncias testes, o que totaliza 600 instâncias testes. Em todos os testes o tempo computacional máximo foi arbitrariamente limitado em 1800 segundos.

Tabela 2 – Tipos de instâncias

Tipo da instância	Tamanho das tarefas ( $s_j$ )	Capacidade da máquina ( $B$ )
1	[1,10]	10
2	[4,8]	10
3	[1,5]	10
4	[2,4]	10
5	[1,5]	5
6	[2,4]	5

Os resultados para os três modelos, considerando tempos de processamento  $p_i \in [1,10]$  e  $p_j \in [1,5]$ , são apresentados nas tabelas 3 e 4, respectivamente. Nessas tabelas, as colunas apresentam o tipo da instância, o número de tarefas, o tempo de processamento, o número de soluções comprovadamente ótimas e o *gap* final, este último definido por  $gap = 100(\text{Sol}^* - \text{Sol}_L) / \text{Sol}_L$  ( $\text{Sol}^*$  é a melhor solução obtida e  $\text{Sol}_L$  o limitante inferior). Para os modelos 1 e 2 a coluna que apresenta o número de soluções ótimas mostra dois valores, o número de soluções que o modelo provou serem ótimas e o número de soluções ótimas encontradas. O Modelo 3 encontrou todos os ótimos que os demais modelos foram hábeis para encontrar. Os valores, a exceção do número de ótimos, são médias das dez instâncias de cada configuração.

A partir das tabelas 3 e 4, é possível verificar que o Modelo 3 apresenta o melhor desempenho médio em todos os quesitos avaliados. O Modelo 1, usualmente utilizado na literatura como base de comparação, apresenta o pior desempenho e não raro o processamento é interrompido devido ao limite de tempo.

O pior desempenho, considerando todos os três modelos, ocorreu para os experimentos com as instâncias do tipo 3 e 4. As tarefas que compõem essas instâncias apresentam tamanhos dentro dos intervalos  $[1,5]$  e  $[2,4]$  e capacidade máxima da máquina igual a 10.

A diferença entre o maior tamanho das tarefas e a capacidade da máquina pode ser investigada como uma possível explicação para este desempenho pior. De fato, para este caso, o número máximo de lotes pode ser limitado por  $n / \lfloor S / s_{\max} \rfloor$ , em que  $s_{\max}$  é o maior tamanho das tarefas.

Tabela 3 – Resultados dos modelos 1,2, e 3 para os seis tipos de instâncias com  $p_i \in [1,10]$

Inst	n	Modelo 1			Modelo 2			Modelo 3		
		T(s)	#O	G(%)	T(s)	#O	G(%)	T(s)	#O	G(%)
1	20	241,2	9 (10)	0,13	0,1	10 (10)	0,00	0,0	10	0,00
	40	1800,0	0 (9)	8,57	187,4	10 (10)	0,00	0,6	10	0,00
	60	1651,1	1 (9)	10,96	304,7	10 (10)	0,00	1,7	10	0,00
	80	1800,0	0 (6)	10,84	583,3	9 (10)	0,08	14,6	10	0,00
	100	1800,0	0 (0)	67,45	1213,3	5 (9)	1,40	85,1	10	0,00
2	20	194,4	9 (10)	0,13	0,0	10 (10)	0,00	0,0	10	0,00
	40	1630,0	1 (10)	6,01	0,2	10 (10)	0,00	0,2	10	0,00
	60	1800,0	0 (10)	13,64	29,4	10 (10)	0,00	0,9	10	0,00
	80	1800,0	0 (9)	12,90	128,4	10 (10)	0,00	1,1	10	0,00
	100	1800,0	0 (1)	61,80	326,0	10 (10)	0,00	2,4	10	0,00
3	20	234,1	9 (10)	1,89	168,2	10 (10)	0,00	1,0	10	0,00
	40	1800,0	0 (1)	49,06	1639,5	1 (2)	6,81	1462,1	3	3,97
	60	1800,0	0 (0)	71,81	1800,0	0 (0)	23,56	1800,0	0	16,67
	80	1800,0	0 (0)	81,58	1800,0	0 (0)	33,84	1800,0	0	22,49
	100	1800,0	0 (0)	86,29	1800,0	0 (0)	31,30	1800,0	0	20,28
4	20	205,9	10 (10)	0,00	309,7	10 (10)	0,00	2,3	10	0,00
	40	1800,0	0 (1)	51,66	1800,0	0 (2)	15,05	1696,9	2	8,62
	60	1800,0	0 (0)	68,59	1800,0	0 (0)	30,31	1800,0	0	24,20
	80	1800,0	0 (0)	81,41	1800,0	0 (0)	35,56	1800,0	0	27,80
	100	1800,0	0 (0)	87,31	1800,0	0 (0)	41,24	1800,0	0	32,07
5	20	555,6	7 (10)	0,46	0,1	10 (10)	0,00	0,0	10	0,00
	40	1800,0	10 (10)	7,14	65,5	10 (10)	0,00	1,1	10	0,00
	60	1800,0	0 (9)	8,16	543,0	9 (9)	0,31	181,4	9	0,12
	80	1276,6	4 (10)	6,33	493,5	10 (10)	0,00	1,8	10	0,00
	100	1800,0	0 (1)	71,37	906,4	6 (9)	0,31	192,7	9	0,06
6	20	731,4	6 (10)	2,28	0,0	10 (10)	0,00	0,0	10	0,00
	40	1267,9	3 (10)	3,99	0,3	10 (10)	0,00	0,2	10	0,00
	60	1306,6	3 (10)	3,54	31,2	10 (10)	0,00	0,7	10	0,00
	80	1578,2	3 (10)	3,29	270,8	10 (10)	0,00	1,9	10	0,00
	100	1800,0	0 (1)	66,02	572,6	10 (10)	0,00	5,8	10	0,00

Tabela 4 – Resultados dos modelos 1,2, e 3 para os seis tipos de instâncias com  $p_i \in [1,5]$

Inst	N	Modelo 1			Modelo 2			Modelo 3		
		T(s)	#O	G(%)	T(s)	#O	G(%)	T(s)	#O	G(%)
1	20	486,6	9 (10)	0,26	0,1	10 (10)	0,00	0,0	10	0,00
	40	1444,7	2 (10)	7,18	148,7	10 (10)	0,00	0,3	10	0,00
	60	1800,0	0 (8)	14,82	714,1	8 (10)	0,49	3,1	10	0,00
	80	1800,1	0 (7)	12,23	449,5	9 (10)	0,37	12,3	10	0,00
	100	1800,3	0 (1)	39,39	1088,6	5 (9)	1,55	263,8	9	0,45
2	20	217,1	9 (10)	0,20	0,0	10 (10)	0,00	0,0	10	0,00
	40	922,3	5 (10)	3,04	0,2	10 (10)	0,00	0,1	10	0,00
	60	1693,3	1 (10)	11,78	52,5	10 (10)	0,00	0,6	10	0,00
	80	1800,1	0 (10)	11,91	153,0	10 (10)	0,00	1,1	10	0,00
	100	1800,1	0 (1)	39,79	312,3	10 (10)	0,00	2,2	10	0,00
3	20	757,0	7 (10)	5,61	71,5	10 (10)	0,00	0,6	10	0,00
	40	1800,0	0 (3)	59,66	1786,8	1 (4)	11,39	1232,7	4	4,26
	60	1800,0	0 (0)	74,05	1800,0	0 (0)	17,85	1800,0	0	12,72
	80	1800,0	0 (0)	82,44	1800,0	0 (0)	21,35	1800,0	0	14,31
	100	1800,1	0 (0)	86,80	1800,0	0 (0)	27,54	1800,1	0	16,67
4	20	880,3	6 (10)	8,38	190,5	10 (10)	0,00	1,4	10	0,00
	40	1800,0	0 (2)	53,02	1686,3	1 (3)	10,61	1328,8	3	5,39
	60	1800,0	0 (0)	71,13	1800,0	0 (0)	22,62	1800,0	0	15,42
	80	1800,0	0 (0)	80,86	1800,0	0 (0)	26,58	1800,0	0	21,09
	100	1800,1	0 (0)	87,51	1800,1	0 (0)	32,53	1800,1	0	21,59
5	20	186,8	9 (10)	0,26	0,1	10 (10)	0,00	0,0	10	0,00
	40	1621,8	1 (10)	8,86	95,6	10 (10)	0,00	0,6	10	0,00
	60	1459,7	2 (10)	6,87	371,0	10 (10)	0,00	1,5	10	0,00
	80	1670,6	1 (10)	8,41	819,9	8 (10)	0,49	171,8	10	0,00
	100	1790,1	1 (2)	38,91	409,9	9 (10)	0,22	6,9	10	0,00
6	20	281,8	9 (10)	0,25	0,0	10 (10)	0,00	0,0	10	0,00
	40	732,5	6 (10)	2,29	0,3	10 (10)	0,00	0,1	10	0,00
	60	1141,6	4 (10)	3,75	43,2	10 (10)	0,00	0,4	10	0,00
	80	1040,1	6 (10)	1,43	204,7	10 (10)	0,00	0,4	10	0,00
	100	1760,9	3 (6)	36,75	436,4	10 (10)	0,00	0,7	10	0,00

A Tabela 5 exibe resultados comparativos para os modelos 1 e 3, uma vez que este último produziu os melhores resultados. Nessa tabela são apresentados o número total médio de nós avaliados e o tempo médio para encontrar a melhor solução reportada pelo respectivo modelo. Esses resultados demonstram o efeito da simetria quando se observa a grande diferença no número de nós avaliados pelos modelos e, por conseguinte, o tempo para encontrar uma solução ou provar que é ótima. Para instâncias com 100 tarefas eventualmente o Modelo 1 avalia menos nós que o Modelo 3, mas isso ocorre porque o tempo de processamento de cada nó é maior no primeiro caso. É possível comprovar esse fato a partir do tempo para encontrar a solução ótima, ou a melhor solução (incumbente), que é menor para o Modelo 3.



Tabela 5 – Número de nós e tempo para encontrar a melhor solução dos modelos 1 e 3.

Inst	N	[1,10]				[1,5]			
		Modelo 1		Modelo 3		Modelo 1		Modelo 3	
		Nós	Tempo I (s)	Nós	Tempo I (s)	Nós	Tempo I (s)	Nós	Tempo I (s)
1	20	128284,6	1,3	2,0	1,0	796819,3	0,6	0,0	0,0
	40	490418,9	58,0	121,8	0,5	433238,3	27,5	48,9	0,2
	60	111008,9	485,5	609,8	1,4	132692,1	482,0	406,7	1,9
	80	20255,8	1325,7	2237,3	7,5	44883,7	985,0	238,7	1,9
	100	394,9	779,8	2813,1	15,3	2351,5	851,4	2419,4	18,6
2	20	167265,0	0,7	1,0	1,0	512713,5	0,5	3,0	3,0
	40	631222,3	45,3	112,1	0,1	464006,1	26,6	45,6	0,1
	60	130188,8	324,2	222,9	0,6	111600,6	227,6	133,6	0,4
	80	17902,5	1256,4	199,8	0,8	36201,0	739,5	132,4	0,7
	100	976,0	930,7	275,2	1,6	1194,8	1343,5	258,3	1,4
3	20	303808,8	13,0	459,6	0,5	1257184,4	0,5	337,4	0,5
	40	216297,7	623,0	120131,5	317,3	221048,0	355,2	4548,9	20,0
	60	57965,7	1279,8	118828,1	755,2	105467,6	1078,2	10064,8	106,2
	80	14415,8	910,8	46009,1	832,4	38845,5	944,4	10462,5	237,4
	100	4826,8	1202,0	25394	916,5	8010,8	1181,8	10668,3	426,6
4	20	365556,1	3,7	1757,4	1,1	1372957,9	1,1	369,3	0,5
	40	238880,0	1124,3	65760,6	194,9	261852,4	678,0	10424,7	31,4
	60	60838,0	1516,0	144692,4	969,2	99290,2	1357,0	24161	172,9
	80	14433,8	1198,3	59896,2	955,7	35903,1	1378,3	11929	264,3
	100	3814,5	1369,3	29031,4	910,3	7363,6	1069,1	16838,6	527,3
5	20	920134,2	0,7	11,1	3,0	311928,7	0,5	7,4	3,0
	40	571035,5	40,1	29,6	0,2	654636,0	32,9	145,9	0,5
	60	243105,4	378,0	3078,1	4,7	181448,3	177,7	237,4	2,1
	80	32019,9	859,2	154,9	2,5	49261,3	758,7	429,6	3,1
	100	1596,4	1416,5	1801,3	11,3	2018,1	1320,3	139,4	4,4
6	20	1276645,0	1,1	4,5	2,0	557400,4	0,4	4	4,0
	40	717905,4	28,8	42,7	3,1	445392,1	15,0	51,7	6,1
	60	192822,0	178,5	110,7	3,5	219118,8	136,7	32,8	4,2
	80	47645,4	963,4	87,5	3,9	45986,0	482,2	27,3	6,1
	100	397,1	862,7	164,1	4,9	6276,7	1598,9	64,1	4,4

A Tabela 6 compara o valor de função objetivo do problema relaxado para os Modelos 2 e 3. Nesse experimento é possível verificar o impacto do pré-processamento. A coluna Mod 3 exibe o valor médio de função objetivo e a coluna D o desvio percentual determinado pela fórmula  $(Sol_2 - Sol_3) / Sol_3 \cdot 100$ , em que  $Sol_x$  é o valor da solução obtida com o modelo  $x$ . Pode-se verificar que em todos os casos o Modelo 3 fornece soluções no mínimo 25% e algumas vezes até 100% melhores. O resultados do Modelo 1 não foram incluídos nessa comparação, uma vez que, devido ao método utilizado na geração das instâncias teste, a relaxação linear deste modelo invariavelmente fornece um valor de função objetivo igual ao maior tempo de processamento.

Tabela 6 – Valor da função objetivo do nó raiz do Modelo 3 e desvio D em relação ao Modelo 1

n	Inst	[1,10]		[1,5]		Inst	[1,10]		[1,5]	
		Mod 3	D(%)	Mod 3	D(%)		Mod 3	D(%)	Mod 3	D(%)
20	1	59,2	43,3	31,3	26,7	4	35,4	94,7	18,6	70,1
		120,3	45,9	68,1	43,7		71,2	99,7	34,7	75,4
		189,0	59,2	101,8	53,2		95,1	126,3	52,7	87,9
		232,3	59,8	131,9	55,1		126,5	109,0	72,5	94,3
		294,0	56,7	164,2	55,4		161,9	124,1	91,3	112,9
20	2	66,7	42,2	36,6	29,8	5	73,6	46,9	38,9	47,4
		130,5	40,1	72,7	35,4		124,6	36,2	69,2	30,8
		196,0	59,8	109,0	41,5		190,4	43,9	108,9	39,6
		265,5	47,4	145,8	42,2		264,1	39,8	147,0	44,3
		329,7	49,4	174,1	38,9		317,3	53,0	177,2	37,4
20	3	35,1	83,9	18,1	57,7	6	64,7	38,7	37,6	59,9
		65,8	76,0	34,9	96,5		133,0	48,5	70,1	25,7
		100,5	130,2	52,1	93,9		200,2	49,6	107,7	43,3
		135,4	154,4	73,9	96,2		262,0	47,6	139,1	36,6
		163,9	123,1	90,7	103,9		333,1	44,0	178,3	48,9

A Tabela 7 apresenta resultados agregados para todos os tipos de instâncias. O objetivo é apresentar o tempo médio para encontrar a melhor solução reportada a partir dos modelos 1 e 3 e o *gap* entre essas soluções determinado por  $100 (Sol_1 - Sol_3) / Sol_3$ , em que  $Sol_x$  é o valor da solução obtida com o modelo  $x$ . Pode-se observar que o Modelo 1 fornece resultados inferiores e com maior custo computacional do que aqueles obtidos com o Modelo 3. Os piores resultados do Modelo 1 ocorrem justamente nas instâncias mais difíceis, ou seja, as instâncias do tipo 3 e 4.

Tabela 7 – Dados agregados para os seis tipos de instância

Inst	[1,10]			[1,5]		
	T(s) I 1	T(s) I 3	G*(%)	T(s) I 1	T(s) I 3	G*(%)
1	530,0	5,1	3,314	469,3	4,5	2,547
2	511,5	0,8	1,862	467,5	1,1	0,964
3	805,7	564,38	7,957	712,0	158,1	7,128
4	1042,3	606,24	8,395	896,7	199,2	7,291
5	538,9	4,34	2,072	458,0	2,6	1,509
6	406,9	3,48	2,174	446,6	5,0	0,734

## 5. Conclusões

Neste trabalho foi proposta uma nova formulação de programação inteira 0-1 para modelar o problema de programação e dimensionamento de lotes de produção em máquina única. A formulação proposta faz uso de dois conjuntos de desigualdades válidas e é mais forte do que aquela comumente utilizada na literatura como base de comparação para outros métodos. Adicionalmente, foi utilizado, em conjunto com as desigualdades válidas, um procedimento de pré-processamento que ordena as variáveis segundo o tamanho.

Os resultados computacionais para o conjunto de instâncias teste gerados demonstraram o desempenho superior da formulação proposta em todos os quesitos avaliados, em que os melhores resultados foram obtidos com o pré-processamento em conjunto com as desigualdades válidas.

Como trabalho futuro os autores pretendem considerar na formulação limitantes para o número de lotes.

## 6. Referências

- Damodarn, P., Manjeshwar, P. e Srihari, K.** (2006). Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms, *International Journal of Production Economics*, 103, 882-891.
- Dupont, L. e Jolai Ghazvini, F.** (1998). Minimizing makespan on a single batch processing machine with non-identical job sizes, *European Journal of Automation Systems*, 32, 431-440.
- Dupont, L. e Dhaenens-Flipo, C.** (2002). Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure, *Computers & Operations research*, 29, 807-819.
- Jolai Ghazvini F. e Dupont, L.** (1998). Minimizing mean flow time on a single batch processing machine with non-identical job sizes, *International Journal of Production Economics*, 55, 273-280.
- Köhler, V.**, *Programação Matemática Aplicada ao Problema de Clusterização com Aplicação em Engenharia de Software*. Tese de Doutorado, Universidade Federal do Rio de Janeiro- COPPE/UFRJ, 2012.
- Melouk, S., Damodarn, P. e CHANG, P.Y.** (2004). Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing, *International Journal of Production Economics*, 87, 141-147.
- Mathirajan, M., e Sivakumar, A. I.** (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *International Journal of Advance Manufacturing*, 29, 990-1001.
- Parsa, N. R., Karimi, B. e Kashan, A. H.** (2010). A branch and price algorithm to minimize makespan on a single batch processing machine with non-identical job sizes, *Computers & Operations Research*, 37, 1720- 1730.
- Potts, C. N. e Kovalyov, M. Y.** (2000). Scheduling with batching: a review, *European Journal of Operational Research*, 120, 228-249.
- Uzsoy, R.** (1994). A single batch processing machine with non-identical job sizes, *International Journal of Production Research*, 32, 1615-1635.