

UMA HEURÍSTICA DETERMINÍSTICA PARA A RESOLUÇÃO DO PROBLEMA GERAL DE DIMENSIONAMENTO DE LOTES E PROGRAMAÇÃO DA PRODUÇÃO

Claudio Fabiano Motta Toledo

Departamento de Sistemas de Computação, ICMC, USP
Av. Trabalhador são-carlense, 400, Centro, CEP 13566-590, São Carlos, SP
claudio@icmc.usp.br

Renato Resende Ribeiro de Oliveira

Departamento de Ciência da Computação - Universidade Federal de Lavras
Campus Universitário, C.P. 3037, CEP 37200-000, Lavras, MG
renatorro@comp.ufla.br

Márcio da Silva Arantes

Departamento de Ciência da Computação - Universidade Federal de Lavras
Campus Universitário, C.P. 3037, CEP 37200-000, Lavras, MG
marcio@comp.ufla.br

Marluce Pereira

Departamento de Ciência da Computação - Universidade Federal de Lavras
Campus Universitário, C.P. 3037, CEP 37200-000, Lavras, MG
marluce@dcc.ufla.br

Paulo Morelato França

Departamento de Matemática, Estatística e Computação - UNESP
R. Roberto Simonsen, 305, CEP 19060-900, P. Prudente, SP
paulo.morelato@fct.unesp.br

RESUMO

O presente artigo propõe uma heurística baseada na resolução de submodelos gerados para o Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP) com e sem máquinas paralelas. Um modelo matemático é apresentado para este caso do PGDLPP assim como um pseudocódigo para o método proposto. A heurística gera submodelos em três fases distintas, onde o objetivo é utilizar uma quantidade reduzida de variáveis. Instâncias do problema são solucionadas pelo *solver* CPLEX cujas soluções servem para avaliação da heurística. Os resultados revelam um desempenho superior em termos de qualidade das soluções obtidas para o PGDLPP com máquinas paralelas e um desempenho superior em termos de tempo computacional para o PGDLPP com máquinas simples.

PALAVRAS CHAVE. Programação da produção, Dimensionamento de lotes, Heurísticas.

Área principal: Metaheurísticas.

ABSTRACT

The present paper proposes a heuristic based on the solution of sub models generated for the General Lot sizing and Scheduling Problem (GLSP) with and without parallel machines. A mathematical model is presented for this GLSP case as well as pseudo code for the proposed method. The heuristic generates sub models for three different phases, where the objective is to handle a reduced amount of variables. Problem instances are solved by solver CPLEX whose solutions will be benchmarks to evaluate the performance of the heuristic. The results shows a superior performance taking into account the solutions quality fund to GLSP with parallel machines and a superior performance taking into account the computational time for GLSP with single machine.

KEYWORDS. Scheduling, Lot sizing, Heuristics.

Main area: Metaheuristics.

1. Introdução

O presente trabalho propõe uma heurística para resolver do Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP) com máquinas simples e paralelas. O PGDLPP estudado considera penalização para demandas não atendidas, onde uma formulação matemática é utilizada para essa variante do problema. As instâncias geradas para o PGDLPP estudado seguem os critérios propostos por Haase (1996) para o PGDLPP com máquinas simples. Inicialmente, as instâncias geradas são solucionadas usando a ferramenta de modelagem matemática CPLEX. As soluções obtidas nesta etapa servem como critério para avaliação do desempenho da heurística proposta.

Modelos e métodos propostos para solucionar problemas de dimensionamento de lotes e programação da produção podem ser encontrados nas revisões da literatura descritas em Drexl e Kimms (1997) e Karimi *et al.* (2003). Estudos considerando o PGDLPP com máquinas paralelas são apresentados em Kang *et al.* (1999) e Meyr (2002). O primeiro trabalho trata o PGDLPP com máquinas paralelas usando um método baseado em geração de coluna e *branch & bound*. O segundo utiliza a metaheurística *simulated annealing* para determinar a atribuição de produtos às linhas e períodos, em seguida é executado um algoritmo exato responsável por determinar o dimensionamento de lotes e os estoques dos produtos em cada período.

O dimensionamento de lotes com restrição de capacidade é um problema da classe NP-Difícil segundo Bitran e Yanasse (1982). Chen e Thizy (1990) estabelecem que o problema de dimensionamento de lotes multi-item também é um problema NP-Difícil. Esses problemas surgem em diversas situações práticas. Por exemplo, Luche *et al.* (2009) e Clark *et al.* (2010) apresentam modelos de programação inteiro-misto (MIP) para esse tipo de problema no contexto da produção de grãos eletrofundidos e de ração animal.

A complexidade desses problemas leva à proposição de métodos heurísticos ou metaheurísticos para sua resolução, onde uma revisão desses métodos é apresentada por Jans e Degraeve (2007) considerando sua aplicação em problemas de dimensionamento de lote e programação da produção. Um Algoritmo Genético (AG) foi utilizado por Toledo *et al.* (2009) para solucionar o PGDLPP. Os autores utilizaram uma representação em árvore ternária para os indivíduos, onde indivíduos são separados em cluster com uma ordem de hierarquia entre eles. O AG proposto foi capaz de encontrar boas soluções, obtendo melhor desempenho em instâncias do PGDLPP com máquinas paralelas.

O presente trabalho propõe uma heurística que soluciona submodelos inteiro-mistos a partir do MIP originalmente estabelecido para o PGDLPP. Esses modelos são gerados considerando critérios que visam otimizar o uso de variáveis binárias, ou seja, os submodelos solucionados tendem a ter um número reduzido de variáveis binárias.

O artigo está organizado em 5 seções. A seção 2 apresenta o modelo matemático para o PGDLPP. A seção 3 descreve a heurística proposta cujo desempenho é avaliado na seção 4. As conclusões deste trabalho seguem na seção 5.

2. Modelo Matemático do PGDLPP

O modelo matemático apresentado a seguir para o PGDLPP segue a formulação descrita em Meyr (2002) para o PGDLPP com máquinas paralelas. A principal diferença está na inserção de variáveis que acumulam as demandas não atendidas. Essas variáveis são incluídas no primeiro período de produção na equação de balanço de estoque. Elas também são penalizadas na função objetivo do modelo. Toledo *et al.* (2007) utiliza penalização de demandas não atendidas em um modelo matemático para um problema de dimensionamento de lotes e programação da produção em fábricas de refrigerantes.

O problema é modelado dividindo o horizonte de planejamento em T macro-períodos, onde cada macro-período t possui um número fixo de micro-períodos S . Os macro-períodos têm tamanho fixo enquanto os micro-períodos podem ter tamanhos variados. O tamanho de cada micro-período s varia de forma proporcional ao tempo de produção do lote de produto a ele atribuído. Um único produto é atribuído a cada micro-período.

A Figura 1 exemplifica essa idéia para três produtos (P1, P2 e P3) com três micro-períodos em cada macro-período de uma linha $L1$.

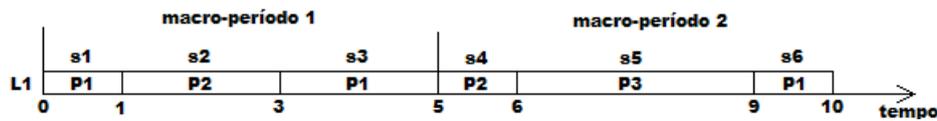


Figura 1. Variação dos micro-períodos.

O tamanho dos micro-períodos $s1$, $s2$, $s3$, $s4$, $s5$ e $s6$ varia de forma proporcional ao tempo de produção. Observe que $P3$ é o produto que consome maior capacidade em $L1$, portanto, o micro período $s5$ é o maior em $L1$. Os micro-períodos $s1$, $s4$ e $s6$ ocupam 1 unidade de tempo, $s2$ ocupa 2 unidades e $s5$ ocupa 3 unidades de tempo. Um micro período poderia ter um produto atribuído com um lote de tamanho zero. Neste caso, o micro-período seria nulo.

O modelo matemático descrito nesta seção é o mesmo proposto em Toledo *et al.* (2009) Assim, foi utilizada a mesma notação para parâmetros e variáveis apresentada a seguir.

Parâmetros:

- J Número de produtos.
- L Número de linhas (máquinas).
- T Número de macro períodos.
- S Número de micro períodos.
- C_t Capacidade em unidades de tempo disponível no macro-período t .
- TP_{lj} Tempo de processamento de uma unidade do produto j na linha l .
- Min_j Lote mínimo do produto j .
- H_j Custo de estoque do produto j .
- CT_{ij} Custo de troca do produto i para o produto j .
- D_{jt} Demanda do produto i no macro-período t .
- $I_{j,0}$ Estoque inicial do produto j no início do horizonte de tempo.
- $y_{lj,0}$ 1, se o produto j está ajustado inicialmente para a linha l ; 0 caso contrário.
- P_j Penalização por unidade de demanda não atendida do produto j .

Variáveis:

- I_{jt} Estoque do produto j ao final do macro-período
- q_{ljs} Quantidade do produto j produzido no micro-período s da linha l .
- q_j^0 Quantidade de demanda do produto j que não foi produzida.
- y_{ljs} 1 Se o produto j é atribuído à linha l no micro-período s ; 0 caso contrário.
- z_{lijs} $z_{lijs} = 1$ se há troca do produto i para j no micro-período s da linha l ; $z_{lijs} = 0$, caso contrário.

A formulação do PGDLPP segue:

$$\text{minimize} \sum_{j=1}^J \sum_{t=1}^T H_j \cdot I_{jt} + \sum_{i=1}^J \sum_{j=1}^J \sum_{l=1}^L \sum_{s=1}^{TS} CT_{ij} \cdot z_{lijs} + \sum_{j=1}^J P_j \cdot q_j^0 \quad (1)$$

Sujeito à:

$$I_{j1} = I_{j,0} + q_j^0 + \sum_{l=1}^L \sum_{s=1}^S q_{ljs} - D_{j1} \quad j = 1, \dots, J \quad (2a)$$

$$I_{jt} = I_{j,t-1} + \sum_{l=1}^L \sum_{s=(t-1)S+1}^{tS} q_{ljs} - D_{jt} \quad j = 1, \dots, J; t = 2, \dots, T \quad (2b)$$

$$\sum_{j=1}^J \sum_{s=(t-1)S+1}^{tS} TP_{lj} \cdot q_{ljs} \leq C_t \quad l = 1, \dots, L; t = 1, \dots, T \quad (3)$$

$$q_{ljs} \leq \frac{C_t}{TP_{lj}} \cdot y_{ljs} \quad l = 1, \dots, L; j = 1, \dots, J; s = 1, \dots, TS \quad (4)$$

$$q_{ljs} \geq \text{Min}_j \cdot (y_{ljs} - y_{ljs-1}) \quad l = 1, \dots, L; j = 1, \dots, J; s = 1, \dots, TS \quad (5)$$

$$\sum_{j=1}^J y_{ljs} = 1 \quad l = 1, \dots, L; s = 1, \dots, TS \quad (6)$$

$$z_{lijs} \geq y_{li,s-1} + y_{lis} - 1 \quad l = 1, \dots, L; i = 1, \dots, J; j = 1, \dots, J; s = 1, \dots, TS \quad (7)$$

$$I_{jt}, q_{ljs}, q_j^0, z_{lijs} \geq 0; y_{ljs} \in \{0, 1\} \quad (8)$$

A função objetivo minimiza os custos de estoque e os custos de troca envolvendo produtos. Os custos de troca foram considerados como dependentes da sequência dos produtos, mas independentes das linhas. As demandas não atendidas são acumuladas nas variáveis q_j^0 e penalizadas na função objetivo. O estoque de cada macro-período é determinado pelas equações (2a) e (2b). Observe que a demanda não atendida é acumulada na equação de balanço de estoque do primeiro período (2a). Isso permite que as demandas não atendidas nos demais períodos continuem satisfazendo a restrição de balanço de estoque já que são acumuladas em q_j^0 . Considera-se que essas demandas foram produzidas em um período $t = 0$ com capacidade ilimitada. Esse tipo de representação foi utilizada por Toledo (2007) num modelo integrado de dimensionamento de lotes e programação da produção em fábricas de bebidas.

A restrição (3) garante que a capacidade disponível dentro no período t não seja violada. A capacidade disponível por macro período é a mesma para todas as linhas, mas cada linha pode ter diferentes tempos de processamento por produto (TP_{lj}). O tamanho máximo do lote de um produto em um micro período é dado pela restrição (4). Essa restrição também impõe que nada seja produzido, caso não ocorra atribuição do produto ao micro período ($y_{ljs} = 0$). A restrição (5) assegura que um lote mínimo será produzido quando ocorre efetivamente a atribuição de um produto a um micro período ($y_{ljs} = 1$). Por outro lado, se o mesmo produto é atribuído a dois micro períodos consecutivos, não necessariamente haverá produção na segunda atribuição. A equação (6) permite que somente um produto seja produzido em um micro período. As trocas de produtos são determinadas pela restrição (7). A equação (8) apresenta os domínios das variáveis

3. Heurística para o PGDLPP

A heurística proposta tem como objetivo solucionar submodelos gerados a partir do modelo original apresentado na seção 2. Os submodelos são estabelecidos através de critérios que visam

reduzir a quantidade de variáveis binárias do modelo original. Desta forma, espera-se inicialmente acelerar o desempenho do *solver* utilizado na resolução desses submodelos. Todavia, também serão adotados critérios que permitam determinar soluções de boa qualidade quando comparadas aquelas encontradas pelo *solver* na resolução do modelo original.

A Figura 2 mostra um pseudocódigo para a heurística proposta.

```

1: function GREEDYMODELS(tempoTotal)
2:   // Início da Fase 1
3:   for All  $l \in [1..L], t \in [1..T]$  do
4:     if  $t == 1$  then
5:        $S_t^l = \frac{J}{L}$ ;
6:     else
7:        $S_t^l = \frac{J}{2 * L}$ ;
8:     end if
9:   end for
10:  modelo = CriarSubModelo( $S_t^l$ );
11:  ExecutarCPLEX(modelo,  $\frac{\text{tempoTotal}}{6}$ );
12:  // Fim da Fase 1 e Início da Fase 2
13:  for All  $l \in [1..L], t \in [1..T]$  do
14:     $S_t^l = S_t^l + 1$ ;
15:  end for
16:  modelo = CriarSubModeloComSoluçãoInicial( $S_t^l$ , modelo);
17:  tempoRestante = tempoTotal – tempoDecorrido;
18:  ExecutarCPLEX(modelo,  $\frac{\text{tempoRestante}}{5}$ );
19:  // Fim da Fase 2 e Início da Fase 3
20:  RemoveMicroPeriodosNãoUtilizados( $S_t^l$ , modelo);
21:  for All  $l \in [1..L], t \in [1..T]$  do
22:     $S_t^l = S_t^l + 1$ ;
23:  end for
24:  modelo = CriarSubModeloComSoluçãoInicial( $S_t^l$ , modelo);
25:  tempoRestante = tempoTotal – tempoDecorrido;
26:  ExecutarCPLEX(modelo, tempoRestante);
27:  // Fim da Fase 3
28:  return modelo;
29: end function
  
```

Figura 2. Pseudocódigo da heurística desenvolvida.

A heurística apresenta características gulosas na seleção dos micro-períodos que geram os submodelos. Por isso, foi chamada de *Greedy Models Heuristic* (GMH). O método é dividido em três fases que solucionam submodelos diferenciados pelos critérios que estabelecem a quantidade de variáveis envolvidas. O parâmetro S_t^l define o número de micro-períodos em cada macro-período t na máquina l . Os significados dos parâmetros T e L são os mesmos definidos na notação da seção 2.

Na fase 1 de GMH, a quantidade de micro-períodos para cada máquina no primeiro macro-período é igual ao número de produtos dividido pelo número de máquinas. Para os demais macro-períodos, esse valor é definido como a metade do valor estabelecido no primeiro macro-período. Esses critério foi adotado já que normalmente os primeiros macro-períodos produzem mais que os últimos macro-períodos.

Além disso, a quantidade de micro-períodos em cada macro-período impacta diretamente no

número de variáveis e restrições do modelo. A maioria das variáveis estão indexadas por micro-períodos: q_{ljs} , y_{ljs} e z_{ljs} . Um submodelo inicial é então criado e solucionado pelo *solver* com um limite de tempo igual à 1/6 do tempo total de execução. A fase 1 tem o objetivo de obter rapidamente uma solução inicial para o problema.

A fase 2 começa após o término da execução do *solver* para o submodelo da fase 1. As quantidades de micro-períodos para todas as máquinas e macro-períodos são incrementadas em 1 unidade na fase 2. Assim, um submodelo novo é gerado com mais um micro-período em cada macro-período, ou seja, com mais variáveis indexadas pelo novo micro-período. Esse submodelo é solucionado pelo *solver* usando a solução obtida na fase 1 como ponto de partida. A resolução desse submodelo é limitada em 1/5 do tempo de execução que ainda resta ($tempoTotal - tempoDecorrido$). A fase 2 tem o objetivo de melhorar a solução da fase 1 e de identificar as máquinas e macro-períodos onde há maior demanda por produção.

A fase 3 também parte da solução obtida na fase 2 após algumas análises. Todos os micro-períodos não utilizados na solução da fase 2, $y_{ljs} = 0$ para todo $j \in [1..J]$, são retirados da quantidade total de microperíodos da máquina l no macro-período t (S_{lt}). Em seguida, as quantidades de micro-períodos para todas as máquinas e macro-períodos são novamente incrementadas em 1 unidade.

Todavia, devido a remoção prévia dos micro-períodos não utilizados, as quantidades S_{lt} não são mais idênticas para cada linha em cada macro-período. Assim, o acréscimo de um novo micro-período segue um critério guloso: destina-se a maior parte dos micro-períodos para as máquinas e macro-períodos onde há mais produtos alocados para produção.

Um novo submodelo é criado a partir dessas novas quantidades de micro-períodos. O *solver* é então executado para este novo submodelo partindo da solução encontrada na fase 2. Esta terceira e última fase da GMH é executada por todo o tempo ainda disponível na tentativa de melhorar o máximo possível a qualidade da solução final.

4. Resultados Computacionais

A heurística desenvolvida foi testada em instâncias geradas da mesma forma que a descrita em Haase (1996). As instâncias geradas contemplam o caso do PGDLPP com máquina simples (uma única linha de produção) e máquinas paralelas (várias linhas de produção).

A Tabela 1 mostra a configuração de parâmetros para cada conjunto de instâncias gerado. A coluna UC(%) indica a porcentagem de utilização da capacidade das máquinas. A notação utilizada é a mesma que a definida na seção 2.

O restante dos parâmetros para geração das instâncias foram definidos como $D_{jt} \in [0..100]$, $H_j = 1$, $TP_{lj} \in [0, 5..1, 5]$, $CT_{ij} \in [100..200] \forall i \neq j$, $P_j = H_j + \max(S_{ij})$, $Min_j = 1$, e $C_t = 50 \cdot J$. Um total de 10 instâncias foi gerado para cada conjunto, totalizando 40 instâncias de máquina simples e 30 instâncias de máquinas paralelas.

Conjunto	L	T	S	J	UC(%)	Tipo
MS1	1	6	4	4	80	Simple
MS2	1	6	4	4	90	Simple
MS3	1	5	4	4	90	Simple
MS4	1	5	5	5	90	Simple
MP1	2	6	4	4	80	Paralelas
MP2	3	6	6	6	80	Paralelas
MP3	4	6	8	8	80	Paralelas

Tabela 1. Parâmetros das instâncias geradas

Todos os testes foram executados em um computador Intel Core 2 Duo 2,66 GHz com 2GB de memória RAM. O *solver* IBM ILOG CPLEX 12.1 com a ferramenta de modelagem GAMS foi

utilizado para solucionar o modelo inteiro-misto do PGDLPP.

A *Concert Technology* do CPLEX para linguagem Java foi utilizada na implementação do método GMH. Inicialmente, o *solver* CPLEX é utilizado para determinar as soluções considerando o modelo completo proposto na seção 2. Essas soluções serão a base de comparação para avaliação do desempenho da heurística proposta. Todas as execuções foram limitadas em 1 hora. Os resultados para máquina simples e máquinas paralelas são apresentados nas próximas seções.

4.1. Instâncias de Máquina Simples

A Tabela 2 mostra os resultados obtidos para os quatro conjuntos de instâncias de máquina simples. O tempo de execução não atinge o limite de 3600 segundos quando o solver encontra o resultado ótimo para o modelo padrão ou quando GMH encontra o ótimo para o submodelo da Fase 3. Os desvios (do tempo computacional e do valor da solução) são calculados utilizando a seguinte equação:

$$Devio(\%) = 100,0 \cdot \frac{V^{GMH} - V^{CPLEX}}{V^{CPLEX}} \quad (9)$$

O parâmetro V^{GMH} indica o valor obtido (do tempo ou da solução) por GMH e V^{CPLEX} o valor obtido pelo CPLEX.

Pode-se observar que, em termos de qualidade das soluções, GMH é levemente superior ao CPLEX aplicado ao modelo padrão. Há praticamente um empate nos resultados obtidos pelo CPLEX e GMH como indica a coluna Desvio Médio Solução. A diferença maior de desempenho aparece no tempo de execução. O método GMH é superior ao CPLEX padrão, uma vez que consegue resultados com qualidade equivalente em um tempo menor. GMH foi mais rápido que o CPLEX em todas as instâncias de máquina simples. Em média, GMH consegue reduzir em até 84,69% o tempo de execução.

Na Tabela 3 são mostrados os valores absolutos das soluções encontradas ao final da execução de GMH e do CPLEX. GMH obteve 3 resultados melhores que o CPLEX, 35 resultados iguais e apenas 2 resultados piores. A Tabela 4 utiliza os modelos gerados para a instância MS4-9 por cada abordagem para exemplificar a quantidade de variáveis envolvida. A instância MS4-9 foi selecionada por apresentar o maior desvio da solução em relação ao CPLEX.

Percebe-se que a quantidade de variáveis q_j^0 e I_{jt} permanece constante em todas as fases, pois essas variáveis não são indexadas por micro-períodos. Os valores nas linhas das variáveis q_{ljs} e y_{ljs} são idênticos já que estas variáveis possuem a mesma indexação. O número de variáveis binárias (y_{ljs}) iniciou com 52% da quantidade total, chegando a 68% na segunda fase e terminando com 80% do número total de variáveis.

Conjunto	Tempo Médio CPLEX	Tempo Médio GMH	Desvio Médio Tempo	Desvio Médio Solução
MS1	2001,91	306,54	-84,69%	0,00%
MS2	2018,36	584,72	-71,03%	0,00%
MS3	221,28	73,34	-66,86%	0,00%
MS4	3600	1519,47	-57,79%	-0,04%

Tabela 2. Resultados para instâncias de máquina simples

4.2. Instâncias de Máquinas Paralelas

A Tabela 5 mostra os resultados obtidos para os três conjuntos de instâncias de máquinas paralelas. O tempo de execução atingiu o limite de 3600 segundos para todas as instâncias de máquinas

Instância	Solução CPLEX	Solução GMH	Instância	Solução CPLEX	Solução GMH
MS1-0	1703,64	1704,24	MS3-0	1405,02	1405,02
MS1-1	1500,88	1500,88	MS3-1	1443,00	1443,00
MS1-2	1450,94	1450,94	MS3-2	1393,07	1393,07
MS1-3	1567,70	1567,70	MS3-3	1383,75	1383,75
MS1-4	1495,22	1495,22	MS3-4	1670,52	1670,52
MS1-5	1652,00	1652,00	MS3-5	1420,35	1420,35
MS1-6	1545,00	1545,00	MS3-6	1599,29	1599,29
MS1-7	1535,79	1535,79	MS3-7	1385,34	1385,34
MS1-8	1585,08	1585,08	MS3-8	1326,13	1326,13
MS1-9	1561,06	1561,06	MS3-9	1442,04	1442,04
MS2-0	1477,75	1477,75	MS4-0	1938,70	1938,70
MS2-1	1594,74	1594,74	MS4-1	1930,14	1930,14
MS2-2	1953,45	1953,45	MS4-2	2029,00	2029,00
MS2-3	1462,42	1462,42	MS4-3	1730,51	1730,51
MS2-4	1747,28	1747,28	MS4-4	1972,86	1881,86
MS2-5	1717,43	1717,43	MS4-5	1773,44	1830,83
MS2-6	1669,74	1669,74	MS4-6	2053,96	2038,54
MS2-7	1680,83	1680,83	MS4-7	1615,32	1615,32
MS2-8	1811,05	1811,05	MS4-8	1806,00	1806,00
MS2-9	1661,52	1661,52	MS4-9	2040,82	1874,82

Tabela 3. Valor das soluções para máquina simples

Variáveis	CPLEX	GMH-1	GMH-2	GMH-3
q_j^0	5	5	5	5
I_{jt}	25	25	25	25
q_{ljs}	125	65	85	100
y_{ljs}	125	65	85	100
z_{ljs}	625	325	425	500
Total	905	485	625	730

Tabela 4. Número de variáveis para MS4-9

paralelas. Isso se justifica pela complexidade elevada dessas instâncias. Logo, nem o *solver* e nem GMH conseguiram encontrar o ótimo. Os desvios (do tempo e da solução) são calculados utilizando a equação (9).

Como o tempo de execução neste caso foi igual para o CPLEX e para GMH, os desvios médios das soluções foram divididos para as três fases de GMH. Pode-se observar que GMH consegue melhorar os resultados obtidos pelo CPLEX nos três conjuntos de instância ao final da terceira fase. Essa melhoria é mais expressiva nas instâncias com maior complexidade, atingindo até 49,48% de melhoria no conjunto MP3.

Porém, os resultados de GMH começam a superar o *solver* CPLEX mesmo levando em conta apenas os resultados obtidos na segunda fase. Nesta fase GMH consegue melhoria média de até 47,53%. Observando a Fase 1, nota-se que mesmo com 1/6 do tempo total de execução GMH obtém uma melhoria de de 10,77% no conjunto de instâncias de maior porte.

A Tabela 6 apresenta os valores absolutos das soluções obtidas pelo CPLEX e pelas três fases de GMH. Ao final da Fase 3, GMH obteve 27 resultados melhores que os resultados obtidos por CPLEX e apenas 3 resultados iguais aos do CPLEX. Ao final da segunda fase, GMH consegue 26 resultados melhores que o CPLEX e apenas 4 resultados piores, mesmo com limite de tempo de 1/3 do total. Ainda ao final da primeira fase, GMH obtém 7 resultados melhores que o CPLEX e 23 resultados piores.

Conjunto	Tempo Médio	Tempo Médio	Desvio Médio	Desvio Médio da Solução		
	CPLEX	GMH	Tempo	Fase 1	Fase 2	Fase 3
MP1	3600	3600	0,00%	140,68%	-0,79%	-2,63%
MP2	3600	3600	0,00%	41,29%	-17,52%	-18,83%
MP3	3600	3600	0,00%	-10,77%	-47,53%	-49,48%

Tabela 5. Resultados para instâncias de máquinas paralelas

Instância	Solução	Solução de GMH		
	CPLEX	Fase 1	Fase 2	Fase 3
MP1-0	1400,00	1903,65	1340,76	1340,76
MP1-1	1285,05	2180,19	1325,91	1285,05
MP1-2	1445,11	7229,85	1395,11	1395,11
MP1-3	1437,60	3547,54	1460,84	1437,60
MP1-4	1507,00	4606,14	1385,00	1385,00
MP1-5	1358,61	1924,54	1463,49	1354,83
MP1-6	1576,11	5292,35	1569,30	1542,30
MP1-7	1421,74	1996,92	1475,90	1421,74
MP1-8	1511,29	3879,93	1463,06	1463,06
MP1-9	1367,35	2377,07	1301,00	1301,00
MP2-0	3149,80	3832,00	2529,94	2529,94
MP2-1	3143,27	3779,00	2601,95	2595,99
MP2-2	3698,20	3931,00	2663,16	2663,16
MP2-3	3457,56	6776,75	2948,00	2869,00
MP2-4	3239,20	5953,88	2624,35	2507,00
MP2-5	2948,13	3960,21	2515,00	2515,00
MP2-6	3030,59	3613,20	2321,00	2321,00
MP2-7	3179,09	3665,92	3002,74	2821,81
MP2-8	3282,27	3929,00	2671,00	2671,00
MP2-9	3241,20	6364,09	2777,31	2734,31
MP3-0	7648,94	6677,84	4214,90	3953,62
MP3-1	7700,64	7105,56	4658,13	4475,42
MP3-2	9889,44	6321,00	3928,59	3679,59
MP3-3	28650,60	6357,67	4155,00	3796,05
MP3-4	8689,96	7809,92	4188,40	4061,00
MP3-5	10823,90	6743,92	4223,62	4161,62
MP3-6	6569,38	6197,08	3933,33	3933,34
MP3-7	6015,17	6445,00	4070,00	3937,36
MP3-8	6528,40	10887,63	4539,18	4258,18
MP3-9	6473,25	6870,67	4567,03	4481,70

Tabela 6. Valor das soluções para máquinas paralelas

Como exemplo do comportamento do GMH para máquinas paralelas, a Tabela 7 apresenta o número de variáveis utilizadas pelo CPLEX e por GMH para a instância MP3-3. Essa instância também foi selecionada por apresentar o maior desvio das soluções de GMH em relação ao CPLEX. É notável a redução no número de variáveis do modelo. Pode-se notar que as variáveis binárias iniciam com apenas 14,58% do total, aumentam para 27,08% na segunda fase e terminam com 35,41% do total das variáveis binárias.

Os melhores resultados do método GMH se justificam pelo fato do método trabalhar com um conjunto reduzido de variáveis como ilustrado na Tabela 7. O método começa com poucas variáveis visando obter uma solução inicial rápida. Em seguida, mais variáveis são atribuídas ao modelo e, por último, mais variáveis são atribuídas somente às linhas e períodos onde se utilizou efetivamente as variáveis anteriormente acrescentadas. Esse comportamento adaptativo é a base de trabalho do

GMH para obter boas soluções a partir de modelos com reduzido número de variáveis.

Variáveis	CPLEX	GMH-1	GMH-2	GMH-3
q_j^0	8	8	8	8
I_{jt}	48	48	48	48
q_{ljs}	1536	224	416	544
y_{ljs}	1536	224	416	544
z_{lij}	12288	1792	3328	4352
Total	15416	2296	4216	5496

Tabela 7. Número de variáveis para MP3-3

5. Conclusão

O presente trabalho propôs uma heurística baseada na resolução de submodelos para solucionar o Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP) com e sem máquinas em paralelo, e com penalização por demandas não atendidas. Um modelo matemático foi apresentado e instâncias para o PGDLPP com máquinas simples e paralelas foram geradas. Essas instâncias foram solucionadas inicialmente usando o *solver* CPLEX e os resultados obtidos foram utilizados na avaliação do desempenho da heurística. O método proposto foi chamado de *Greedy Models Heuristic* (GMH) já que solucina o problema gerando submodelos com um número reduzido de microperíodos e de acordo com a necessidade de produção das máquinas.

Na primeira fase, uma quantidade de variáveis é determinada seguindo um critério de atribuição de produtos a linhas e períodos. Na segunda fase, um micro-período é acrescentado ao modelo o que repercute diretamente na quantidade de variáveis. Afinal, a maioria das variáveis do modelo estão indexadas por produtos, micro-períodos e linhas. Por último, as variáveis não utilizadas na fase 2 são removidas dos modelos na fase 3 e mais um micro-período é acrescentado. Porém, esse micro-período é adicionado a partir da nova quantidade de períodos estabelecida para cada linha e macro-período.

Os testes computacionais nas instâncias do PGDLPP com máquina simples demonstram que GMH alcança as mesmas soluções obtidas pelo CPLEX em um tempo computacional menor. A redução no tempo de execução obtida por GMH ficou acima de 50% em todos os conjuntos de instância do PGDLPP com máquina simples. No PGDLPP com máquinas paralelas, o *solver* CPLEX e o GMH foram executados por 1 hora. O método proposto obteve soluções melhores na grande maioria das instâncias solucionadas. O percentual de melhoria alcançada variou de cerca de 2%, para instâncias menos complexas, até quase 50% para instâncias mais complexas. A heurística proposta oferece a vantagem de solucionar submodelos de forma exata, obtendo soluções de boa qualidade dentro de um tempo computacional razoável.

Como trabalhos futuros, pretende-se avaliar o método em instâncias da literatura, assim como comparar seu desempenho em relação a outros métodos heurísticos (Algoritmos Genéticos, Busca Tabu, entre outros). Também será objeto de trabalhos futuros a avaliação da abordagem proposta em um conjunto de instâncias reais ou mais complexas do que as solucionadas neste artigo.

Referências

- Bitran, G. R. e Yanasse, H. H., (1982), Computational complexity of the capacitated lot size problem, *Management Science*, 28 (10), 1174-1186.
- Chen, W. H. e Thizy, J. M., (1990), Analysis of relaxations for the multi-item capacitated lot-sizing problem, *Annals of Operations Research*, 26, 29-72.

- Clark, A., Morabito, R. e Toso, E.,** (2010), Production setup-sequencing and lot-sizing at an animal nutrition plant through ATSP subtour elimination and patching, *Journal of Scheduling*, 13 (2), 111-121.
- Drexel, A. e Kimms, A.,** (1997), Lot sizing and scheduling - survey and extensions, *European Journal of Operational Research*, 99, 221-235.
- Haase, K.,** (1996), Capacitated lot-sizing with sequence dependent setup costs, *OR Spektrum*, 18, 51-59.
- Jans, R. e Degraeve, Z.,** (2007), Metaheuristics for dynamic lot sizing: A review and comparison of solution approaches, *European Journal of Operational Research*, 177, 1855-1875.
- Kang, S., Malik, K. e Thomas, L. J.,** (1999), Lotsizing and scheduling on parallel machines with sequence-dependent setup costs, *Management Science*, 45, 273-289.
- Karimi, B., Fatemi Ghomi, S. M. T. e Wilson, J. M.,** (2003), The capacitated lot sizing problem: a review of models and algorithms, *Omega*, 31, 365-378.
- Luche, J. R. D., Morabito, R. e Pureza, V.,** (2009), Combining process selection and lot sizing models for production scheduling of electrofused grains, *Asia-Pacific Journal of Operational Research*, 26 (3), 421-443.
- Meyr, H.,** (2002), Simultaneous lotsizing and scheduling on parallel machines, *European Journal of Operational Research*, 139, 277-292.
- Toledo, C. F. M., França, P. M., Morabito, R. e Kimms, A.** (2007), Um modelo de otimização para o problema integrado de dimensionamento de lotes e programação da produção em fábricas de refrigerantes, *Pesquisa Operacional*, 27, 155-186.
- Toledo, C. F. M., Filho, J. E. F. J., Simeone, F. e Rosa, G. P.,** (2009), Metaheurísticas aplicadas ao problema geral de dimensionamento de lotes e programação da produção, In: *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional*, Porto Seguro, Brasil.