

ABORDAGENS HEURÍSTICAS APLICADAS AO PROBLEMA GERAL DE DIMENSIONAMENTO E PROGRAMAÇÃO DE LOTES

Nilton Hideki Takagi¹, Lucídio dos Anjos Formiga Cabral², Gilberto Farias³

Universidade Federal da Paraíba¹²³
nilton.takagi@gmail.com¹, lucidio@di.ufpb.br², gilbertofarias@gmail.com³

Resumo – O problema de dimensionamento e sequenciamento de lotes é um problema NP-Completo e aqui tratado de forma integrada. Uma heurística gulosa foi criada para o dimensionamento e uma meta-heurística GRASP foi construída para o sequenciamento. O cenário se concentra em uma fábrica de bebidas, no qual instâncias reais e da literatura foram utilizadas nos testes computacionais. As instâncias delimitam o problema em multi-itens, multilíquidos, multiperíodos e uma linha de produção. Um modelo matemático sobre o problema oriundo da literatura foi analisado e implementado no solver CPLEX e os resultados comparados com a solução desenvolvida. Resultados satisfatórios foram observados, validando a eficiência e robustez da solução.

Palavras chaves – Sequenciamento e dimensionamento de lotes, GRASP, otimização combinatória e indústria de bebidas.

1 Introdução

No Brasil há mais de 800 fábricas de bebidas e que anualmente produzem em torno de 14 bilhões de litros, com faturamento superior a 20 bilhões de reais [1]. A programação da produção nessas indústrias envolve fatores como demanda dos produtos, capacidade da linha de produção (máquina), capacidades dos tanques, tempo de preparação da máquina (*setup*), dependência da sequência de produção, entre outros [2]. Esses fatores trabalham de forma conjunta no processo de dimensionamento e sequenciamento de lotes (*lot sizing*) e devido à quantidade de variáveis envolvidas no processo de tomada de decisão se faz necessária uma ferramenta computacional adequada.

No processo de dimensionamento deve ser levado em consideração a restrição de capacidade do tanque e o lote mínimo a ser produzido. Quanto maior o tamanho do lote, menor é o tempo total de *setup*, pois irá minimizar a intercalação de produtos. Entretanto, quanto maior o tamanho do lote de produção, maior é o risco de haver quebra de estoque em virtude do adiamento da produção de outros itens. Manter o equilíbrio para definir o dimensionamento é fundamental para evitar esse tipo de transtorno [3].

Por um ajuste mecânico ou pela alternância na produção de itens é necessário um tempo de preparação para a limpeza da linha de produção [4,5]. Como a ordem de produção dos itens impacta diretamente no tempo de preparação e a quantidade de soluções possíveis é uma permutação, logo o número de soluções é igual a $n!$ (fatorial de n), onde n é a quantidade de itens a serem produzidos. O problema é considerado na literatura como problema NP-Difícil [6,7]. Soluções utilizando programação linear inteira [8], programação dinâmica [9] e abordagens heurísticas [10,11,12] são encontradas para o problema.

2 Modelos de dimensionamento e sequenciamento de lotes (*lot sizing*)

Na literatura há estudos relacionados ao dimensionamento e sequenciamento de lotes na indústria de bebidas [2,4,5,13,14,15,16,17,18], fundição [19,20], química [3], têxtil [21], produção de ração animal [22,23], entre outras áreas importantes. Há também análises sobre os efeitos de paradas de máquinas e decisões sobre a correção [24].

Os modelos têm como função objetivo minimizar custos de estoque, produção e preparação de máquinas. Nesses modelos, à medida que a quantidade de itens e períodos aumenta, avança o grau de complexidade. Dada a dimensão dos problemas atuais, o uso de métodos exatos encontra dificuldades para determinar a solução em tempo computacional razoável, mesmo para instâncias de média dimensão. Nas instâncias mais complexas, heurísticas são amplamente utilizadas.

O problema da Quantidade Econômica do Pedido (QEP) foi o primeiro modelo envolvendo dimensionamento de lote. O modelo assume tempo de escala contínuo, indicando que sempre haverá recursos disponíveis para a produção de cada item. A taxa de demanda é constante e o tempo de horizonte infinito [25,26,27]. Wagner e Whitin [25] apresentam um algoritmo com uma versão dinâmica para o modelo de dimensionamento de lotes econômico. Abordam um item simples, custos de estoque, custos de preparação entre períodos e com função objetivo de minimizar o custo total de estoque e satisfazendo a demanda. DeMatteis [28] faz uma comparação de desempenho demonstrando uma técnica que calcula o custo do item por período, chamado de parte-períodos. Esse valor é usado para determinar o tamanho do lote. Choi [29] mostra o problema considerando o horizonte de planejamento.

No Problema de Dimensionamento de Lotes Capacitado (PDLC) foi proposta na sua formulação uma restrição de limite de capacidade e a possibilidade de produzir mais de um item [26,27]. Özdamar e Barbarosoglu [10] abordam o problema com uma meta-heurística integrando conceitos de *Simulated Annealing* e a metodologia clássica de relaxação lagrangiana. A

solução decompõe o problema em pequenos subproblemas e intensifica a busca utilizando *Simulated Annealing*. Comparando os resultados com Tempelmeier e Derstroff [30] o tempo de processamento foi superior, entretanto ocorreram estratégias que atingiram média de resultados mais próximas da solução ótima, chegando a uma melhora de mais 60% em um dos casos.

Partindo do PDLC aonde os períodos são considerados como macro-períodos, no Problema Discreto de Dimensionamento e Programação de Lotes (PDDPL) cada período é subdividido em diversos períodos (micro-períodos). As demandas e capacidades serão avaliadas nesses micro-períodos. Nesse problema temos a hipótese do “tudo ou nada”. Ou seja, toda a capacidade deverá ser utilizada ou nada é produzido [7]. Também é considerado um problema NP-Difícil [8,9].

Na produção de um período, nem toda a capacidade disponível é utilizada. No Problema Proporcional de Dimensionamento e Programação de Lotes (PPDPL) a capacidade ociosa pode ser aproveitada para um segundo item. A divisão proporcional da capacidade disponível do período entre dois itens motivou o nome desse problema [27,7]. Uma variação do PPDPL utilizando uma máquina simples e de multi-nível é modelada [31].

O Problema Geral de Dimensionamento e Programação de Lotes (PGDPL) consiste em determinar os dimensionamento de lotes contínuos de diversos produtos e programar os lotes em uma única máquina com capacidade limitada [32,27]. Fleischmann [32] apresenta uma solução através da heurística *Threshold Accepting*. Fabricio [15] resolve através de um modelo de programação linear inteira.

3 Modelo matemático do problema e solução abordada

3.1 Modelo matemático

O PGDPL foi abordado utilizando a heurística *relax and fix* [18] no qual consta o modelo matemático com a seguinte formulação:

Tamanho do problema:

J : Número de itens (bebidas);

L : Número de sabores dos líquidos;

T : Número de macro-períodos;

N : Total de sub-períodos;

Índices:

$i, j \in \{1, \dots, J\}; k, l \in \{1, \dots, L\}; t \in \{1, \dots, T\}; s \in \{1, \dots, N\};$

Conjuntos:

S_t : Número de cada sub-período t ;

α : Itens que podem ser produzidos;

β : Sabores que podem ser produzidos;

γ : Itens que precisam do xarope l para ser produzido;

Dados:

d_{jt} : Demanda do item j no período t ;

h_j : Custo unitário de estoque para o item j ;

g_j : Custo unitário de atraso para o item j ;

s_{ij}^H : Custo da troca do item i para o j ;

b_{ij}^H : Tempo da troca do item i para o j ;

a_j^H : Tempo de produção de uma unidade do item j ;

K_t^H : Tempo de capacidade total da máquina no período t ;

K^L : Capacidade total dos tanques em litros;

q_l^L : Lote mínimo do sabor l para garantir a homogeneidade do líquido;

r_{lj} : Quantidade necessária do sabor do líquido l para produzir uma unidade do item j ;

y_{j0}^H : 1, se a máquina é inicialmente configurada para o item j ; 0, em caso contrário;

I_{j0}^+ : Estoque inicial do item j ;

I_{j0} : atraso inicial do item j ; 0, se não houver atraso.

Variáveis:

I_{jt}^+ : Estoque ao final da produção do item j no período t ;

I_{jt}^- : Atraso do item j ao final do período t ;

x_{js}^H : Quantidade produzida do item j no sub-período s ;

y_{ls}^l : 1, se o tanque está indicado para o sabor do liquido l no sub-período s ; 0, caso contrário;

y_{js}^H : 1, se a máquina está indicado para o sabor do liquido l no sub-período s ; 0, caso contrário;

z_{ijs}^H : 1, se haverá troca do item i para j no sub-período s ; 0, caso contrário;

Após a descrição dos parâmetros, apresenta-se o modelo matemático e as restrições vinculadas aos estágios de envase e preparação do líquido:

$$\text{Min}Z = \sum_{j=1}^J \sum_{t=1}^T (h_j I_{jt}^+ + g_j I_{jt}^-) + \sum_{s=1}^N \sum_{i \in \alpha} \sum_{j \in \alpha} (s_{ij}^H z_{ijs}^H) \quad (1)$$

s.a:

Estágio de envase:

$$I_{jt}^+ - I_{jt}^- = I_{j(t-1)}^+ + \sum_{s \in S_t} x_{js}^H - d_{jt} - I_{j(t-1)}^- \quad \forall j, t \quad (2)$$

$$\sum_{j \in \alpha} \sum_{s \in S_t} (a_j^H x_{js}^H) + \sum_{i \in \alpha} \sum_{j \in \alpha} \sum_{s \in S_t} (b_{ij}^H z_{ijs}^H) \leq K_t^H \quad \forall t \quad (3)$$

$$x_{js}^H \leq \frac{K_t^H}{a_j^H} y_{js}^H \quad j \in \alpha, \forall t, s \in S_t \quad (4)$$

$$\sum_{j \in \alpha} y_{js}^H = 1 \quad \forall s \quad (5)$$

$$z_{ijs}^H \geq y_{i(s-1)}^H + y_{js}^H - 1 \quad i, j \in \alpha; \forall s \quad (6)$$

$$\sum_{i \in \alpha} \sum_{j \in \alpha} z_{ijs}^H \leq 1 \quad \forall s \quad (7)$$

Estágio de preparação do líquido:

$$q_l^l y_{ls}^l \leq \sum_{j \in \gamma_l} r_{lj} x_{js}^H \leq K^l y_{ls}^l \quad l \in \beta; s \in S_t; \forall t \quad (8)$$

$$I_{jt}^+, I_{jt}^-, x_{js}^H \geq 0; y_{js}^H, y_{ls}^l, z_{ijs}^H \in \{0,1\} \quad \forall i, j, l, s, t \quad (9)$$

A função objetivo (1) minimiza custos de estoque, atraso e preparação. A restrição (2) é o balanceamento do estoque. A restrição (3) é relativa à capacidade. A restrição (4) garante que a produção do item no lote será produzida apenas se a máquina estiver preparada. A restrição (5) determina que um único item seja atribuído a cada sub-período. A restrição (6) controla a troca de máquina em cada sub-período. A restrição (7) limita a uma única ocorrência de troca ou não de itens no sub-período s . A restrição (8) garante que o tanque está preparado para o liquido l e está entre o mínimo necessário para deixar

o líquido homogêneo e a capacidade máxima. Finalmente a restrição (9) garante a integralidade das restrições e valores não negativos.

3.2 Solução abordada

3.2.1 Heurística gulosa para o dimensionamento de lotes

Como a premissa do dimensionamento é atender a demanda, antes de definir os produtos que serão enviados para a etapa do sequenciamento, há uma análise da demanda de cada produto no período atual e o seu respectivo estoque inicial. Se o estoque inicial for suficiente para atender a demanda, então não se produz o item no período em questão.

Duas abordagens são realizadas para definir o tamanho dos lotes. Uma considerando apenas o período atual, no qual a definição do dimensionamento dos lotes é calculada para suprir a demanda atual. Outra abordagem analisa a demanda nos períodos seguintes e antecipa a produção caso tenha capacidade disponível. A antecipação evita o custo de configuração (*setup*) de produção nos períodos seguintes.

3.2.2 Meta-heurística GRASP para o sequenciamento

O Greedy Randomized Adaptive Search Procedure (GRASP) ou Procedimento de busca adaptativa gulosa e aleatória é um método iterativo que consiste de duas fases: uma fase de construção, na qual uma solução é gerada elemento a elemento, e uma fase de busca local, na qual um ótimo local na vizinhança da solução construída é pesquisado. A melhor solução encontrada ao longo de todas as iterações GRASP realizadas é retornada como resultado [33].

```
Algoritmo Grasp (Sequencia,  $\alpha$ , Filtro)
1. MelhorTempo  $\leftarrow \infty$ ;
2. Para (de 1 até Filtro) faça
3.   Construcao( $g(\cdot)$ ,  $\alpha$ , SequenciaC);
4.   Calcula TempoConstrucao(SequenciaC);
5.   Se (TempoConstrucao < MelhorTempo) então
6.     Armazena a melhor sequencia;
7.     MelhorTempo  $\leftarrow$  TempoConstrucao;
8.   Fim se;
9. fim para;
10. Sequencia  $\leftarrow$  melhor sequencia;
11. BuscaLocal(Sequencia);
12. Retorne Sequencia;
fim Grasp;
```

Figura 1 – Pseudocódigo GRASP

```
Algoritmo Construcao ( $g(\cdot)$ ,  $\alpha$ , SequenciaC)
1. Inicializa SequenciaC;
2. Inicializa o conjunto C de candidatos baseado na Sequencia;
3. Enquanto (C diferente de nulo) faça
4.    $g(t_{min}) = \min \{g(t) \mid t \in C\}$ ;
5.    $g(t_{max}) = \max \{g(t) \mid t \in C\}$ ;
6.   LCR =  $\{t \in C \mid g(t) \leq g(t_{min}) + \alpha (g(t_{max}) - g(t_{min}))\}$ ;
7.   Selecione, aleatoriamente, um elemento t LCR;
8.   SequenciaC  $\leftarrow$  SequenciaC  $\cup$  {t};
9.   Atualize o conjunto C de candidatos;
10. fim enquanto;
11. Retorne SequenciaC;
fim Construcao;
```

Figura 2 – Pseudocódigo da fase da Construção

A Figura 1 exibe o pseudocódigo da meta-heurística GRASP. Foi utilizado o recurso de filtro, no qual realiza refinamentos na etapa da construção para entregar um melhor resultado para a fase da busca local.

O pseudocódigo da fase de construção é demonstrado na Figura 2. O valor do alfa nesse momento regula o nível de aleatoriedade da função. Quanto maior o alfa, maior se torna a lista de candidatos e mais aleatório se torna o sequenciamento dessa etapa.

```
Algoritmo BuscaLocal (Sequencia);
1. MelhorSequencia  $\leftarrow$  Sequencia;
2. Enquanto (movimentos < (|Sequencia| * 300)) faça
3.   MelhorSequencia, Sequencia  $\leftarrow$  {Swap(1,1) | Swap(1,1) < Sequencia};
4.   MelhorSequencia, Sequencia  $\leftarrow$  {Swap(2,1) | Swap(2,1) < Sequencia};
5.   MelhorSequencia, Sequencia  $\leftarrow$  {Swap(2,2) | Swap(2,2) < Sequencia};
6.   Se (não melhorou nos swaps) então
7.     Perturba Sequencia;
8.   Fim se;
9.   Atualiza movimentos;
10. fim enquanto;
11. Retorne MelhorSequencia;
fim BuscaLocal;
```

Figura 3 – Pseudocódigo da fase da Busca Local

A Figura 3 mostra o pseudocódigo da fase da busca local, no qual realiza movimentos de vizinhança até o ponto de parada, definido nessa solução em 300 vezes a quantidade de itens da sequência.

Na fase da busca local utilizamos a estratégia de swap [35], no qual há trocas de elementos. No swap (1,1), Figura 4, há mudança de apenas um elemento da sequência alterando apenas uma posição. Após a troca é verificado se a sequência melhorou o resultado em comparação com a melhor sequência até o momento. Em caso positivo, esta passa a ser a melhor sequência e serve de comparação para as demais trocas. É realizado o processo até que todos os itens sejam trocados.

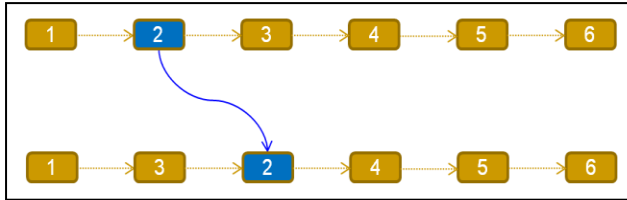


Figura 4 – swap (1,1)

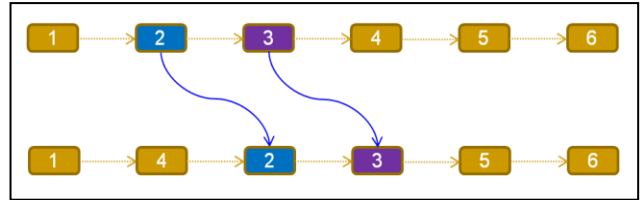


Figura 5 – swap (2,1)

Após todos os elementos trocados, a informação se houve algum resultado de melhoria é armazenado e será utilizado no ponto de parada. Na Figura 5 temos o movimento swap (2,1) realizado em seguida. Nele são trocados dois elementos de uma vez e deslocados uma posição na sequência. O swap (2,2), Figura 6, é realizado após o swap (2,1). Nele são trocados dois elementos de uma vez e deslocados duas posições na sequência.

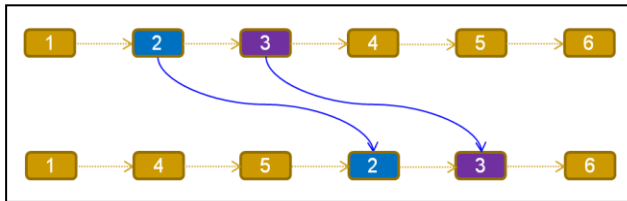


Figura 6 – swap (2,2)

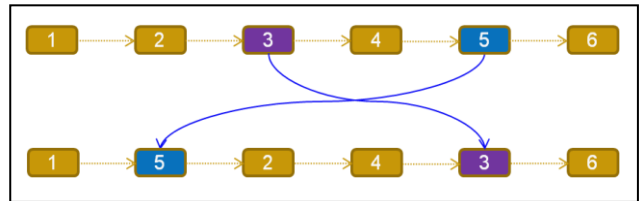


Figura 7 – Perturbação da sequência

Após todos os swaps terem sido avaliados e verificado que não houve nenhuma melhoria, o algoritmo prevê uma perturbação, Figura 7, com o objetivo de encontrar alguma nova vizinhança que melhore a sequência dos itens. O movimento é aplicado em torno de 10% dos elementos, pois uma perturbação maior pode deixar longe do melhor resultado já obtido. Após a perturbação, inicia-se novamente todo o processo desde o primeiro swap.

4 Instâncias, estratégias e infra-estrutura utilizada

Dados de duas fábricas foram utilizados, denominadas de Fábrica A e B [18]. Nas instâncias da Fábrica A foram realizadas as execuções com 8 itens, 6 sabores, 5 períodos, 20 subperíodos e os custos de estoque, produção, *setup* e atraso. Na Fábrica B existem 27 itens, 10 sabores, 5 períodos, 104 subperíodos e os custos foram estimados de acordo com os valores da Fábrica A.

Tabela 1 – Estratégias utilizadas

Estratégia	α	Filtro	Estratégia	α	Filtro	Estratégia	α	Filtro
Est01	0,2	Sem filtro	Est06	0,5	Sem filtro	Est11	0,8	Sem filtro
Est02	0,2	25	Est07	0,5	25	Est12	0,8	25
Est03	0,2	50	Est08	0,5	50	Est13	0,8	50
Est04	0,2	100	Est09	0,5	100	Est14	0,8	100
Est05	0,2	200	Est10	0,5	200	Est15	0,8	200

Conforme a Tabela 1, foram criadas 15 estratégias com alterações das variáveis α e filtro do GRASP. Como a variável α indica o nível de gulosidade, definimos valores próximos a um nível totalmente guloso, 0,2, um nível intermediário, 0,5, e um nível próximo ao totalmente aleatório, 0,8. O filtro foi definido com valores 1 (sem filtro), 25, 50, 100 e 200 iterações. A variação do filtro teve como finalidade verificar o tempo de processamento e os resultados que a fase de construção irá entregar a fase de busca local.

Para estudar o comportamento das estratégias e dado o comportamento estocástico da meta-heurística foram executadas 10 vezes cada estratégia para as diferentes instâncias.

A meta-heurística desenvolvida foi executada em um notebook com processador Intel Core 2 Duo, modelo T8100 com 2.10Ghz, 4 GB de memória. O sistema operacional utilizado foi o Windows XP, 32 bits. A aplicação foi desenvolvida na linguagem C, utilizando o Dev-C++, versão 4.9.9.2.

Para comparação com os resultados da meta-heurística, o modelo matemático foi implementado no CPLEX e as execuções foram realizadas em um desktop com processador AMD ATHLON, com 2.4 Ghz, 2 GB de memória. O sistema operacional foi o Linux Ubuntu e o CPLEX [34] na versão 10 foi utilizado.

5 Resultados computacionais

Os resultados foram divididos em dois cenários, um considerando as melhorias nos tempos de preparação com foco no sequenciamento. Nessa etapa separamos os resultados obtidos em cada fábrica. Para a Fábrica A realizamos testes considerando apenas o período atual, com foco na eficiência do GRASP proposto. Realizamos testes utilizando todas as estratégias definidas, Est01 a Est15. Para a Fábrica B, realizamos testes considerando também períodos posteriores na busca de minimizar o tempo de preparação da linha de produção nos períodos seguintes.

A outra parte dos resultados foi relacionada aos custos de *setup*, estoque e atraso de produção (*backorder*). Nesse cenário são comparados os resultados da meta-heurística desenvolvida com os resultados obtidos no CPLEX. Foi criada para a Fábrica A uma instância menor baseada na original, uma com 3 itens e 2 períodos para auxiliar na comparação de resultados.

No cenário considerando as antecipações de demandas de períodos seguintes, a Figura 8 demonstra que mesmo com pequenas variações na qualidade da solução entregue pela fase de construção para a fase seguinte de busca local utilizando os recursos de *swaps* converge para resultados similares. A Figura 9 demonstra os resultados da Fábrica A no qual o *setup* 1 é analisado somente o período atual e no *setup* 2 os períodos seguintes para o dimensionamento da produção. A minimização do custo de preparação das linhas de produção no *setup* 2 obteve média de 46,35% neste gráfico. Nos valores do *setup* 2 há poucas variações, demonstrando que o sequenciamento está convergindo para resultados próximos. Todas as instâncias testadas neste cenário tiveram comportamento similar, sempre com redução de acordo com a disponibilidade das linhas de produção.

Os resultados de todas as execuções considerando todas as estratégias, na instância reduzida da Fábrica A foram similares. Temos nessa instância 114 variáveis e o CPLEX resolveu com a solução ótima em 3.925,54. A meta-heurística desenvolvida chegou a um resultado de 3.925,77 convergindo para o valor ótimo. A diferença de 0,23 (0,01%) aconteceu por conta de arredondamento do ponto flutuante, demonstrando que para pequenas instâncias a solução encontra o melhor resultado.

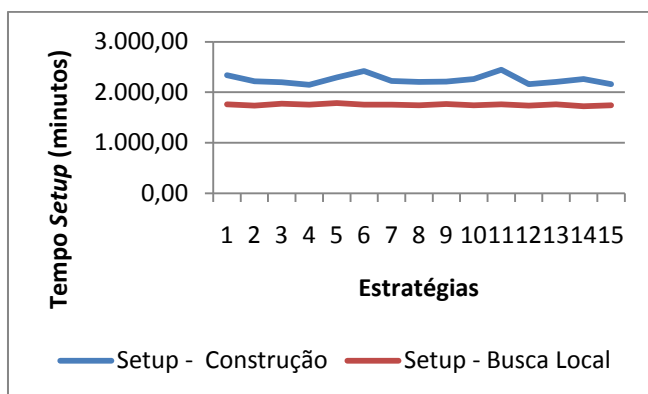


Figura 8 – Comparativo entre construção e busca local

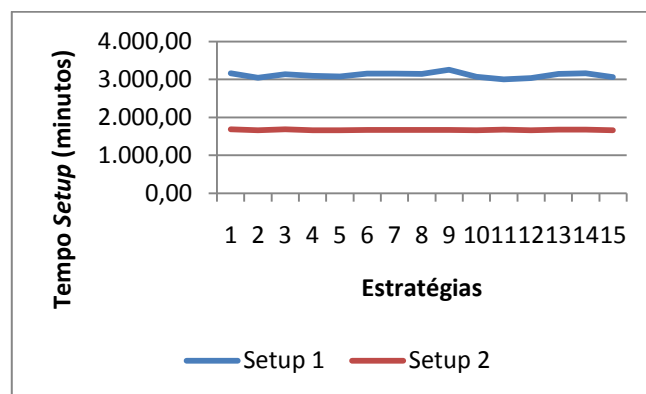


Figura 9 – Comparativo entre tempos de setup da Fábrica A

A Tabela 2 mostra os resultados da instância real da Fábrica A, no qual temos 3.520 variáveis. Após 6 horas de processamento no CPLEX o resultado da função objetivo foi de 27.847,90, composto por 19.194,79 de custo de *setup*, 7.895,20 de custo de atraso e 758,01 de custo de estoque. A meta-heurística demorou menos de 1 segundo para cada instância e os valores ficaram com GAP variando entre 4,52% a 5,36%, com média geral de 4,94%. Os valores da tabela são a média de 10 execuções de cada estratégia e o melhor resultado individual da solução alcançou um custo de 28.816,74, com GAP de 3,48%.

As instâncias da Fábrica B possuem 27 itens e 10 líquidos a serem analisados em 5 períodos. Nessas instâncias há 82.742 variáveis. Pelo tamanho da instância, o CPLEX não conseguiu encontrar nenhuma solução inteira após 6 horas de processamento. Para todas as instâncias trabalhadas nessa fábrica a meta-heurística desenvolvida encontrou uma solução em menos de 1 segundo. Comparando a média geral com as médias de todas as estratégias, a variação ficou inferior a 2% para mais ou para menos, demonstrando pequena oscilação nas soluções encontradas.

Tabela 2 – Resultados da meta-heurística na Fábrica A (instância real)

Estratégia	Setup	Atraso	Estoque	Total	GAP
Est01	21.859,11	7.291,00	131,92	29.282,03	5,15%
Est02	21.859,29	7.291,00	131,92	29.282,21	5,15%
Est03	21.801,36	7.291,00	131,92	29.224,28	4,94%
Est04	21.801,36	7.291,00	131,92	29.224,28	4,94%
Est05	21.684,94	7.291,00	131,92	29.107,86	4,52%
Est06	21.859,47	7.291,00	131,92	29.282,39	5,15%
Est07	21.859,29	7.291,00	131,92	29.282,21	5,15%
Est08	21.684,58	7.291,00	131,92	29.107,50	4,52%
Est09	21.859,47	7.291,00	131,92	29.282,39	5,15%
Est10	21.801,36	7.291,00	131,92	29.224,28	4,94%
Est11	21.742,88	7.291,00	131,92	29.165,80	4,73%
Est12	21.859,65	7.291,00	131,92	29.282,57	5,15%
Est13	21.742,88	7.291,00	131,92	29.165,80	4,73%
Est14	21.916,69	7.291,00	131,92	29.339,61	5,36%
Est15	21.684,41	7.291,00	131,92	29.107,33	4,52%
Média	21.801,12	7.291,00	131,92	29.224,04	4,94%

A solução ótima na instância reduzida da fábrica A e os resultados próximos e com margem pequena ao que o CPLEX encontrou na instância real, demonstram a robustez e a eficácia da meta-heurística desenvolvida. Não há como obter gaps de otimalidade para as instâncias da Fábrica B, mas pela pequena variação nos resultados é possível inferir que os resultados satisfatórios serão obtidos, similarmente aos já encontrados na Fábrica A.

6 Conclusão

A solução desenvolvida foi executada utilizando estratégias baseadas em alterações das variáveis da meta-heurística GRASP, com abordagens analisando demandas dos períodos para a definição da antecipação, ou não, da produção dos itens. Concentrando os esforços no sequenciamento, os resultados alcançaram melhora de 32,12% em média nos tempos de setup, chegando a atingir 51,40% de ganho no melhor dos casos.

Um modelo matemático da literatura foi utilizado e implementado no solver CPLEX. Resultados ótimos obtidos em instâncias reduzidas foram igualmente alcançados pela solução. Instâncias moderadas e originais de uma das fábricas foram submetidas ao solver CPLEX, e após 6 horas de processamento alcançaram soluções viáveis, porém sem a garantia de otimalidade. A solução para a mesma instância realizou o processamento em menos de um segundo mantendo uma diferença de 4,94% do modelo matemático em média. Na instância da Fábrica B o modelo matemático não encontrou nenhuma solução viável, e a solução desenvolvida disponibilizou resultados robustos, com pouca variação e com menos de 1 segundo de processamento. Para sistemas de apoio a decisão, a solução proposta se demonstrou eficiente, pois além de resultados robustos, possui baixo custo computacional.

Como pesquisa futura há a perspectiva de se utilizar a solução desenvolvida com o modelo matemático em uma solução híbrida. Dependendo da complexidade de cada período pode ser alternado quem irá resolvê-lo, ora direcionando alguns períodos para serem refinados pelos mecanismos de busca local e outros poucos períodos para serem refinados pela resolução do modelo matemático associado. Outro trabalho interessante como pesquisa futura seria manter a fase de sequenciamento sendo realizado pela meta-heurística desenvolvida e a fase de dimensionamento sendo resolvido pelo modelo matemático.

7 Referências

- [1] ABIR, Dados de Mercado 2008, **Associação Brasileira das Indústrias de Refrigerantes e de Bebidas Não Alcoólicas**, <http://abir.org.br> (acessado em 23/05/2011).
- [2] D. Ferreira, Tese de Doutorado: Abordagens para o problema integrado de dimensionamento e sequenciamento de lotes da produção de bebidas, **Universidade Federal de São Carlos**, (2006).
- [3] A. M. Santos, C. R. V. Carvalho, Planejamento e sequenciamento da produção de uma máquina aplicado a uma indústria de processo repetitivo em lotes, **XXVI Enegep**, (2006).

- [4] D. Ferreira, R. Morabito, S. Rangel, Solution approaches for the soft drink integrated production lot sizing and scheduling problem, **European Journal of Operational Research**, 196 (2009), 697-706.
- [5] D. Ferreira, A. Clark, B. Almada-Lobo, R. Morabito, Uma reformulação mono estágio de um modelo de programação da produção de bebidas dois estágios com sincronia, **XLI SBPO 2009 - Pesquisa Operacional na Gestão do Conhecimento**, 3267-3276, (2009).
- [6] N. Brahimi, S. Dauzere-Peres, N. Najid, A. Nordli, A review of Single Item Lot Sizing Problems, **IEEE SMC - WA1K3**, (2002).
- [7] A. Drexel, A. Kimms, Lot sizing and scheduling - Survey and extensions, **European Journal of Operational Research**, 99 (1997), 221-235.
- [8] V. S. Hoesel, A. Kolen, A linear description of the discrete lot-sizing and scheduling problem, **European Journal of Operational Research**, 75 (1994), 342-353.
- [9] M. Salomon, M. M. Solomon, L. N. V. Wassenhove, Y. Dumas, S. Dauzère-Pérès, Solving the discrete lotsizing and scheduling problem with sequence dependent set-up costs and set-up times using the Travelling Salesman Problem with time windows. **European Journal of Operational Research**, 100 (1997), 494-513.
- [10] L. Özdamar, G. Barbarosoglu, Integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem, **Int. J. Production Economics**, 68 (2000), 319-331.
- [11] D. Gupta, T. Magnusson, The capacitated lot-sizing and scheduling problem with sequence-dependent Setup costs and Setup times, **Computers & Operations Research**, 32 (2005), 727-747.
- [12] D. Dai, C. Chu, S. Yang, The Optimal Pricing and Multi-Item Capacitated Lot Sizing Problem. **6th World Congress on Intelligent Control and Automation**, (2006).
- [13] C. F. M. Toledo, P. M. França, R. Morabito, Proposta de um modelo conjunto de programação da produção e dimensionamento de lotes aplicado a uma indústria de bebidas, **XXII Enegep**, (2002).
- [14] J. G. Castro, N. D. Pizzolato, A programação de lotes econômicos de produção (ELSP) com tempos e custos de Setup dependentes da sequência: Um estudo de caso. **Revista Gestão Industrial**, v.01 n.03 (2005), 60-70.
- [15] A. S. F. Fabricio, L. A. F. Cabral, A. Subramanian, Problema de dimensionamento de lotes e sequenciamento da produção: Um estudo de caso em uma indústria de bebidas. **XXVII Enegep**, (2007).
- [16] C. F. M. Toledo, P. M. França, K. A. Rosa, Evaluating Genetic Algorithms with Different Population Structures on a Lot Sizing and Scheduling Problem, **SAC'08**, Fortaleza-Ceará-Brasil, (2008).
- [17] D. Ferreira, R. Morabito, S. Rangel, Um modelo de otimização inteira mista e heurísticas relax and fix para a programação da produção de fábricas de refrigerantes de pequeno porte, **Produção**, v.18 n.1 (2008), 76-88.
- [18] D. Ferreira, R. Morabito, S. Rangel, Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants, **Computers & Operations Research**, 37 (2010), 684-691.
- [19] S. A. Araujo, M. N. Arenales, A. R. Clark, Dimensionamento de lotes e programação do forno numa fundição de pequeno porte, **Gestão e Produção**, v11 n2 (2004), 165-176.
- [20] S. A. Araujo, M. N. Arenales, Planejamento e Programação da Produção numa Fundição Cativa Automatizada de Grande Porte, **Associação Portuguesa de Investigação Operacional**, 24 (2004), 197-210.
- [21] C. Silva, J. M. Magalhães, Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry, **Computers & Industrial Engineering**, 50 (2006), 76-89.
- [22] E. A. V. Toso, R. Morabito, Otimização no dimensionamento e sequenciamento de lotes de produção: Estudo de caso numa fábrica de rações. **Gestão e Produção**, v12 n2 (2005), 203-217.
- [23] E. A. V. Toso, R. Morabito, A. R. Clark, Lot sizing and sequencing optimisation at an animal-feed plant, **Computer & Industrial Engineering**, 57 (2009), 813-821.
- [24] H. Kuhn, A dynamic lot sizing model with exponential machine breakdowns, **European Journal of Operational Research**, 100 (1997), 514-536.
- [25] H. M. Wagner, T. M. Whitin, Dynamic Version of the Economic Lot Size Model, **Management Science**, 5 (1958), 89-96.
- [26] R. Jans, Z. Degraeve, Modeling Industrial Lot Sizing Problems: A Review. ERIM Report Series in Management ERS-2005-049-LIS, **Erasmus University Rotterdam**, (2005).
- [27] C. F. M. Toledo, Tese de Doutorado: Problema Conjunto de Dimensionamento de Lotes e Programação da Produção, Faculdade de Engenharia Elétrica e Computação, **Universidade Estadual de Campinas**, (2005).
- [28] J. J. DeMatteis, An economic lot-sizing technique, **IBM System Journal**, 7 (1968).
- [29] Y. Choi, I. Jeong, Heuristics for Dynamic Lot-Sizing with Plan Setup Cost in a Rolling Horizon Environment, **Eighth International Conference on Intelligent Systems Design and Applications**, (2008).
- [30] H. Tempelmeier, M. Derstroff, A Lagrangean-based heuristic for dynamic multi-item multi-level constrained lot sizing with setup times, **Management Science**, 42 (1996), 738-757.
- [31] A. Kimms, Multi-level, single-machine lot sizing and scheduling (with initial inventory), **European Journal of Operational Research**, 89 (1996), 86-99.
- [32] B. Fleischmann, H. Meyr, The general lotsizing and scheduling problem, **University of Augsburg OR Spektrum**, (1996).
- [33] M. J. F. Souza, Notas de aula da disciplina de inteligência computacional para otimização, **Universidade Federal de Ouro Preto**, (2005).
- [34] ILOG, **Using the CPLEX Callable Library**, (2006).
- [35] M. B. Teitz, P. Bart, Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph, **Operations Research**, 16 (1968), 955-961.