PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

CÂMPUS CURITIBA

CURSO DE PÓS-GRADUAÇÃO – ENGENHARIA DA PRODUÇÃO

EDSON LUIZ DE CAMARGO

UMA ABORDAGEM DE OTIMIZAÇÃO DE GRANDE PORTE PARA O PLANO MESTRE DE PRODUÇÃO

CURITIBA

2010

EDSON LUIZ DE CAMARGO

UMA ABORDAGEM DE OTIMIZAÇÃO DE GRANDE PORTE PARA O PLANO MESTRE DE PRODUÇÃO

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia da Produção e Sistemas, da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de Mestre.

Área de concentração: Gerência de Produção e Logística

Orientador: Prof. Dr. Raimundo José Borges de Sampaio

CURITIBA

2010

EDSON LUIZ DE CAMARGO

UMA ABORDAGEM DE OTIMIZAÇÃO DE GRANDE PORTE PARA O PLANO MESTRE DE PRODUÇÃO

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia da Produção e Sistemas, da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de Mestre. Área de concentração: Gerência de Produção e Logística.

COMISSÃO EXAMINADORA

Prof. Dr. Raimundo José Borges de Sampaio

Prof. Dr. Angelo Márcio Oliveira Sant´Anna

Prof. Dr. Anselmo Chaves Neto

RESUMO

O modelo matemático do problema de Plano Mestre de Produção (MPS), quando agregado ao longo de muitos períodos, torna-se, na prática, quase que numericamente intratável pelo grande número de variáveis de decisão que comporta, bem como pelo grande número de restrições presentes no modelo. Nesse trabalho estudou-se uma modelagem para o problema de MPS usando técnicas de decomposição em blocos, que permitem resolver de maneira simples e eficiente esse problema, com pouco armazenamento de dados e com ganho de tempo computacional, mesmo quando o número de variáveis de decisão e/ou número de restrições são muito grandes. A abordagem empregada tira vantagem do fato de que o modelo matemático do problema de MPS é estruturado e esparso, e torna viável sua solução mesmo no caso de problemas de grande porte. A principal contribuição trazida por este trabalho é como realizar, na prática, a decomposição em blocos, quando o MPS é relaxado para um Problema de Programação Linear.

Palavras-chaves: MPS, Problema de Grande Porte, Decomposição.

ABSTRACT

The mathematical model of the problem of Master Production Schedule (MPS), when aggregated over many periods, becomes, in practice, almost numerically intractable by the large number of decision variables that includes, as well as the large number of restrictions in the model. In the present study is a model for the MPS problem using decomposition techniques in blocks, allowing a simple way to solve this problem efficiently and with little storage and gain in computational time, even when the number of decision variables and / or number of constraints are very large. The approach takes advantage of the fact that the mathematical model of the MPS problem is structured and sparse, and its solution becomes feasible even for large problems. The main contribution made by this work is how to accomplish in practice, the decomposition into blocks, when the MPS is relaxed to a Linear Programming Problem.

Keywords: MPS, Large problem, Decomposition.

SUMÁRIO

1 INTRODUÇÃO 06
1.1 CONTEXTUALIZAÇÃO DO TRABALHO 06
1.2 TEMA E QUESTÃO DE PESQUISA 07
1.3 OBJETIVOS 07
1.4 JUSTIFICATIVA 07
1.5 DEFINIÇÃO DA ABORDAGEM METODOLÓGICA 08
1.6 ESTRUTURAÇÃO DO TRABALHO 09
2 DESENVOLVIMENTO 10
2.1 ARTIGO 1 10
2.2 ARTIGO 2 19
2.3 ARTIGO 3 29
3 CONCLUSÃO 46
REFERÊNCIAS 47

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO TRABALHO

Na área de produção, os sistemas de suporte a tomada de decisões geralmente contêm um Plano Mestre de Necessidades de Materiais (MRP) e um Plano Mestre de Produção (MPS). O MRP consiste de um conjunto de regras, logicamente relacionadas, articuladas para suportar o MPS no que diz respeito às quantidades líquidas de materiais necessários à implementação do planejamento de produção. Já o MPS trata especificamente de quanto e quando produzir de cada um dos produtos demandados. Desse modo, podemos imaginar que o MRP pode ser completamente declarado em termos da lista de materiais (BOM) de cada produto. E essa é uma das principais entradas do MPS. Nos casos onde devem ser considerados um grande número de produtos, e cada produto com um grande número de componentes em sua BOM, a modelagem matemática e a solução do problema via programação inteira fica completamente fora de questão pela complexidade computacional.

Por outro lado, como a execução do modelo é geralmente recorrente, para incorporar sempre novos períodos, faz sentido considerar resolver o problema como se fosse um problema de programação linear de grande porte, e não como um problema de programação linear inteira como, por exemplo, em Godinho Filho & Fernandes (2006); Sastim et al. (2006) e Takey & Mesquita (2006). Nesse caso, deve-se utilizar técnicas de programação linear de grande porte e, principalmente, aquelas de decomposição, dado que o número de variáveis de decisão e o número de restrições no problema de programação linear é muito grande. Esse problema foi parcialmente resolvido em Chu (1995), mas o autor teve que descartar todo um conjunto de restrições relacionadas com a capacidade de produção instalada e, portanto, implicitamente, teve que usar uma hipótese não realista de capacidade de produção infinita para um dado tipo de recurso.

Nesse trabalho propõe-se uma abordagem para o problema de MPS, que incorpora suas restrições, considerando, inclusive, uma capacidade finita de recursos (mão-de-obra).

1.2 TEMA E QUESTÃO DE PESQUISA

Nesse trabalho propõe-se a utilização da decomposição em blocos para resolver problemas de MPS de grande porte, de maneira simples e eficiente, através de um modelo de programação matemática.

1.3 OBJETIVOS

O objetivo geral deste estudo é resolver o problema de MPS de grande porte através de um modelo de programação matemática, aplicando-se técnicas de decomposição. Especificamente, pretende-se obter ganho no tempo computacional com a utilização destas técnicas.

1.4 JUSTIFICATIVA

O modelo de programação linear para solução de MPS é um problema de programação linear de grande porte, esparso e estruturado. Neste caso, sem a relaxação da integralidade das variáveis, torna-se de difícil solução numérica, mesmo em casos relativamente simples, com médio número de variáveis e de períodos no horizonte de planejamento.

Por outro lado, dada a robustez do modelo de programação linear, a solução inteira obtida da solução linear é bastante satisfatória, e em muitos casos, a única disponível. É importante observar que a chave dessa decomposição é não permitir restrições de acoplamento, e desse modo, não permitir problemas principais com variáveis de acoplamento (LASDON, 1968), mas apenas subproblemas independentes. Em Chu (1995), relaxando o conjunto de restrições relativas a disponibilidade de recursos em cada período j=1,2,3, ... T, e, também, relaxando a condição de integralidade, o autor mostrou, através de um número reduzido de experimentos, que a solução inteira obtida da solução linear permanece no entorno da solução ótima, o que justifica a relaxação da condição de integralidade.

Sendo assim, mostra-se relevante proceder um estudo sobre a solução do MPS de grande porte através de um modelo de programação matemática, aplicando-se técnicas de decomposição, buscando, inclusive, ganho no tempo de processamento.

Aliado a isso, a abordagem de decomposição aqui apresentada permite resolver problemas de grande escala do MPS, que desempenha um papel fundamental no campo da Engenharia Industrial, ampliando nossa capacidade de resolver problemas práticos.

1.5 DEFINIÇÃO DA ABORDAGEM METODOLÓGICA

A fim de resolver o problema de MPS de grande porte, considerou-se o modelo matemático adotado em Chu (1995), acrescido da variável de restrição de capacidade.

Sob algumas hipóteses realistas foi então formulado um teorema geral que assegura que a solução do problema obtida por meio da decomposição proposta é também uma solução ótima para o problema.

A partir deste resultado, realizaram-se experimentos numéricos a fim de ilustrar o resultado (MPS de certa empresa), considerando diversos tipos de produtos e seus componentes, para um determinado horizonte de planejamento. Neste problema há um grande número de variáveis de decisão e de restrições funcionais. Além disso, foram supostos os rendimentos de cada produto fabricado em cada um dos períodos no horizonte de planejamento, as exigências correspondentes, o tempo padrão para a fabricação de cada um dos produtos, trabalhos manuais e os recursos disponíveis para cada período também.

Os resultados numéricos estão exibidos em tabelas construídas mostrando a melhor solução para o problema, sem decomposição (1 bloco), e, em seguida, usando a decomposição em dois, três, quatro, seis e doze blocos, bem como mostrando o tempo teórico (2^n, sendo n o número de variáveis, proporcional ao número de blocos) e o tempo real de processamento.

1.6 ESTRUTURAÇÃO DO TRABALHO

Este trabalho está organizado em três capítulos. O primeiro capítulo contém uma introdução ao tema, incluindo a problemática, os objetivos e a justificativa. No segundo capítulo a pesquisa é apresentada em três artigos, selecionados de acordo com a finalidade deste trabalho. Finalmente, no terceiro capítulo, são apresentadas conclusões gerais e algumas questões que ficaram em aberto após a realização deste estudo.

2 DESENVOLVIMENTO

2.1 ARTIGO 1

Este artigo foi aprovado no evento XVII SIMPEP, que aconteceu entre os dias 8 a 10 de novembro de 2010, em Bauru – SP.

UMA ABORDAGEM DE OTIMIZAÇÃO DE GRANDE PORTE PARA O PLANO MESTRE DE PRODUÇÃO

EDSON LUIZ DE CAMARGO ()
elcamarg@hotmail.com
RAIMUNDO J. B. DE SAMPAIO (PUCPR)
de.sampaio@brturbo.com.br

Resumo: *O MODELO MATEMÁTICO DO PROBLEMA DE PLANO MESTRE DE PRODUÇÃO (MPS) QUANDO AGREGADO AO LONGO DE MUITOS PERÍODOS TORNA-SE, NA PRÁTICA, QUASE QUE NUMERICAMENTE INTRATÁVEL PELO GRANDE NÚMERO DE VARIÁVEIS DE DECISÃO QUE COMPORTA, BEM COMO PELO GRANDE NÚMERO DE RESTRIÇÕES PRESENTES NO MODELO. NESSE TRABALHO ESTUDOU-SE UMA MODELAGEM PARA O PROBLEMA DE MPS USANDO TÉCNICAS DE DECOMPOSIÇÃO, QUE PERMITE RESOLVER DE MANEIRA SIMPLES E EFICIENTE ESSE PROBLEMA, MESMO QUANDO O NÚMERO DE VARIÁVEIS DE DECISÃO E/OU NÚMERO DE RESTRIÇÕES SÃO MUITO GRANDES. A ABORDAGEM EMPREGADA TIRA VANTAGEM DO FATO DE QUE O MODELO MATEMÁTICO DO PROBLEMA DE MPS É ESTRUTURADO E ESPARSO, E TORNA VIÁVEL SUA SOLUÇÃO MESMO NO CASO DE PROBLEMAS DE GRANDE PORTE.*

Palavras-chaves: *MPS, PROBLEMA DE GRANDE PORTE, DECOMPOSIÇÃO.*

AN APPROACH TO OPTIMIZATION OF LARGE SIZE MASTER PLAN FOR THE PRODUCTION

Abstract: *THE MATHEMATICAL MODEL OF THE PROBLEM OF MASTER PRODUCTION SCHEDULE (MPS) WHEN AGGREGATED OVER MANY PERIODS BECOMES, IN PRACTICE, ALMOST NUMERICALLY INTRACTABLE BY THE LARGE NUMBER OF DECISION VARIABLES THAT INCLUDES, AS WELL AS THE LARGE NNUMBER OF CONSTRAINTS PRESENT IN MODEL. THIS WORK IS STUDIED A MODEL FOR THE MPS PROBLEM USING DECOMPOSITION TECHNIQUES, WHICH ALLOWS TO SOLVE SIMPLY AND EFFICIENTLY THIS PROBLEM, EVEN WHEN THE NUMBER OF DECISION VARIABLES AND / OR NUMBER OF CONSTRAINTS ARE VERY LARGE. THE APPROACH TAKES ADVANTAGE OF THE FACT THAT THE MATHEMATICAL MODEL OF THE PROBLEM OF MPS IS STRUCTURED AND SPARSE, AND TAKES ITS VIABLE SOLUTION EVEN FOR LARGE SCALE PROBLEMS.*

Keywords: *MPS, LARGE SCALE PROBLEM, DECOMPOSITION*

1. Descrição do Problema

Geralmente os sistemas de suporte a tomada de decisão na área de produção contém um plano mestre de necessidades de materiais (MRP) e um plano mestre de produção (MPS). O MRP consiste de um conjunto de regras, logicamente relacionadas, articuladas para suportar o MPS no que diz respeito as quantidades líquidas de materiais necessários à implementação do planejamento de produção, enquanto que o MPS trata especificamente de quanto e quando produzir de cada um dos produtos demandados. Desse modo, podemos imaginar que o MRP pode ser completamente declarado em termos da lista de materiais (BOM) de cada produto. E essa é uma das principais entradas do MPS. Nos casos onde devem ser considerados um grande número de produtos, e cada produto, com um grande número de diferentes componentes em sua BOM, a modelagem matemática, e solução do problema, via programação inteira fica completamente fora de questão. Por outro lado, como a execução do modelo é geralmente recorrente, para incorporar sempre novos períodos, faz sentido considerar resolver o problema como se fosse um problema de programação linear de grande porte, e não como um problema de programação linear inteira como, por exemplo, em Godinho Filho & Fernandes (2006); Sastim *et al.* (2006) e Takey & Mesquita (2006). Nesse caso, devemos utilizar técnicas de programação linear de grande porte e, principalmente, aquelas de decomposição, dado que o número de variáveis de decisão e o número de restrições no problema de programação linear agregado é muito grande. Esse problema é parcialmente resolvido em Chu (1995), mas o autor teve que descartar todo um conjunto de restrições relacionadas com a capacidade de produção instalada e, portanto, implicitamente, teve que usar uma hipótese não realista de capacidade de produção infinita para um dado tipo de recurso.

Nesse trabalho nos propomos uma nova decomposição para o problema de MPS, que incorpora todas as restrições usadas na formulação do problema de MPS, de modo que se torna desnecessário qualquer hipótese do tipo daquela assumida em Chu (1995). Nesse sentido, este trabalho representa um avanço na solução do problema de MPS. O restante desse trabalho está organizado do seguinte modo. Na Seção 2, é apresentada a modelagem matemática do problema de programação linear agregado, e é discutida a questão da relaxação da integralidade do problema, para torná-lo, em geral, numericamente tratável. Na Seção 3, é apresentada a nova decomposição por blocos que incorpora todas as restrições do problema e é provado um resultado fundamental sobre essa decomposição. E finalmente na Seção 4, são apresentados e discutidos alguns resultados numéricos relativos a implementação da decomposição, as conclusões obtidas, e algumas questões em aberto.

2. Modelagem Matemática do Problema

Vamos considerar que temos um mix de produtos indexados por i= 1, 2, ..., n**,** e que a variedade de componentes utilizadas em todas as BOM dos n produtos sejam indexadas por k= 1, 2, ..., k. Seja b_{ki} a quantidade de componentes do tipo k utilizadas para produzir uma unidade do produto i, e seja h_i a quantidade padrão de tempo, necessária para produzir uma unidade do produto i, i = 1, 2,..., n**.** Suponha que os períodos de nosso horizonte de planejamento sejam indexados por j = 1, 2, ..., T, e que g_j seja a quantidade de recursos de mão de obra disponível no período j. Ademais, vamos supor que o recurso g_j possa ser utilizado somente no período j, não podendo ser carreado para qualquer período seguinte, ou anterior. Vamos chamar de S_{kj} o estoque disponível de componentes do tipo k no início do período j, e de d_i, a demanda máxima estimada do produto i em j=T. O rendimento aportado por cada unidade de produto i fabricado no período j será chamado de c_{ij}, e as variáveis de decisão de quantos produtos do tipo i fabricar no período j, serão chamadas de x_{ij}. 0 modelo matemático de nosso problema pode então ser formulado como sendo:

$$
\begin{aligned}
&\textit{Maximizar} \sum_{j=1}^{T} \sum_{i=1}^{n} c_{ij} x_{ij} \\
&\textit{sujeito}\text{ a } \sum_{j=1}^{t} \sum_{i=1}^{n} b_{ki} x_{ij} \le \sum_{j=1}^{t} s_{kj} \quad \text{(I)} \\
&\qquad k = 1,2,\ldots,K \text{ e } t = 1,2,\ldots T \\
&\qquad \sum_{j=1}^{T} x_{ij} \le d_i,\ i = 1,2,\ldots,n \quad \text{(II)} \qquad\qquad (1) \\
&\qquad \sum_{i=1}^{n} h_i x_{ij} \le g_j,\ j = 1,\ 2,\ \ldots,T \quad \text{(III)} \\
&\qquad x_{ij} \ge 0,\ \mathrm{i} = 1,2,\ldots,n \text{ e } j = 1,2,\ldots,T
\end{aligned}
$$

Observe que se *T* é um número finito inteiro de períodos, então deveríamos exigir que a variável de decisão X_{jj} fosse inteira. Todavia, como estamos considerando que *T é* apenas o alcance de nosso horizonte de planejamento, que se repete e se estende ao longo do tempo, isto é, cada vez que *T é* alcançado um novo *T* é definido, e assim sucessivamente, faz sentido então, relaxar essa condição de integralidade. De fato, essa decisão encontra suporte nos experimentos numéricos realizados em Chu (1995) e em Sampaio, Vieira e Favaretto (2009), dado que a solução linear ótima está, em geral, no entorno da solução inteira ótima.

Vamos explicitar agora o modelo matemático (I), (II), (III), definido acima. Os sistemas de inequações (I) e (II), são agregados ao longo dos períodos, fazendo-se variar t=1, 2, ..., *T,* isto é, as variáveis que aparecem na desigualdade obtida para t=1 se repetem para t= 2,..., *T,* e é essa agregação que torna esse modelo um problema de grande porte. No sistema de inequações (I), para cada k=1, 2, ...,*K,* $\sum b_{ki}x_{ij}$ fornece os quantitativos líquidos de cada um dos tipos de componentes que aparecem nas BOM dos produtos a serem produzidos nos períodos j=1, 2, ..., *T,* ou seja, a solução de MRP do modelo.

O problema (1), como formulado, é claramente um problema de grande porte, e a matriz das restrições é esparsa e estruturada como veremos adiante. Observe que esse problema tem NxT variáveis de decisão, e KxT+n+T restrições funcionais, além de nxT restrições de positividade. Mesmo casos triviais como o de uma empresa com, por exemplo, 100 produtos, um horizonte de planejamento de 24 períodos e 150 diferentes tipos de componentes nas BOM(s), isso resultaria em um modelo de programação linear com 2400 variáveis de decisão, 3724 restrições funcionais, e 2400 restrições de não negatividade. Do ponto de vista de programação inteira essa dimensão de problema está fora de questão com respeito a possibilidade de obtenção de solução ótima em tempo computacional razoável. Estes dois aspectos: complexidade computacional, e tempo de computação, requerem uma nova abordagem para o problema, de modo que, mesmo com *n* e *T* grandes, o problema continue sendo numericamente tratável.

3. Decomposicão por Blocos

Primeiro vamos discutir como fazer a decomposição por blocos e, após, discutiremos as questões relacionadas com a otimalidade, bem como com o custo computacional de obtenção de uma solução ótima. Inicialmente divide-se o horizonte de planejamento *T* em blocos de tamanho fixo *L*, de acordo com nossa capacidade de computação. O tamanho *L* do bloco pode variar de L= 1 a L= *T,* isto é, os tamanhos de blocos podem variar de uma total decomposição, L=1, a uma ausência de decomposição, L= *T*. É claro que sempre podemos supor *T* como um múltiplo inteiro de *L*. Mantendo-se fixo o tamanho dos blocos a matriz das restrições não sofre qualquer alteração durante todo o processo computacional, de onde pode-se tirar proveito para reduzir o tempo computacional necessário para a solução do problema.

Vamos assumir que,

1. $x^{(j)} = (x_{1j}, x_{2j}, \ldots, x_{nj})^T$, $j = 1,2,\ldots,T$;
2. $B = (b_{ki}), k = 1,2,\ldots,K, i = 1,2,\ldots,n$;
3. $s^{(j)} = (s_{1j}, s_{2j}, \ldots, s_{Kj})^T$, $j = 1,2,\ldots,T$;
4. $d = (d_1, d_2, \ldots, d_n)^T$;
5. $c^{(j)} = (c_{1j}, c_{2j}, \ldots, c_{nj})$, $j = 1,2,\ldots,T$;
6. $h = (h_1, h_2, \ldots, h_n)$;
7. I é a matrizidentidade corresponde nte.

Assumindo as convenções definidas acima, esse problema de programação linear agregado pode ser escrito de forma matricial como sendo,

$$
\begin{array}{lll}
Max\, c^{(1)}x^{(1)} + c^{(2)}x^{(2)} + \ldots + c^{(T)} & & \\
s/a \quad Bx^{(1)} & \leq s^{(1)} & \\
\quad Ix^{(1)} & \leq d & \\
\quad hx^{(1)} & \leq g_1 & \\
\quad Bx^{(1)} + Bx^{(2)} & \leq s^{(1)} + s^{(2)} & \\
\quad Ix^{(1)} + Ix^{(2)} & \leq d & \\
\quad 0x^{(1)} + hx^{(2)} & \leq g_2 & \\
\quad \vdots & & (2) \\
\quad Bx^{(1)} + Bx^{(2)} + \ldots + Bx^{(T)} & \leq s^{(1)} + s^{(2)} + \ldots + s^{(T)} & \\
\quad Ix^{(1)} + Ix^{(2)} + \ldots + Ix^{(T)} & \leq d & \\
\quad 0x^{(1)} + 0x^{(2)} + \ldots + hx^{(T)} & \leq g_2 & \\
\quad x^{(1)} \geq 0;\ x^{(2)} \geq 0; \ldots x^{(n)} \geq 0 & &
\end{array}
$$

Trata-se claramente de um problema de grande porte, esparso, e estruturado. O modo de resolver esse problema aqui é o seguinte: inicialmente consideramos que os T períodos do horizonte de planejamento sejam particionados em p blocos de tamanho fixo L, e que as quantidades de materiais não utilizados em um período, bem como as demandas não atendidas, possam ser transferidas para o período seguinte, e assim sucessivamente, até o período T. Então a partir do segundo bloco, l = 2, 3, ..., p, ate o p-ésimo bloco de tamanho L, podemos atualizar os parâmetros para o modelo matemático de cada bloco do seguinte modo:

Faça,

$$
s^{(j)}_{k,(l-1)L+1} \leftarrow s^{(j)}_{k,(l-1)(L+1)} + \sum_{l=1}^{p-1} \sum_{j=(l-1)L+1}^{lL} \left(s^{(l)}_{kj} - \sum_{i=1}^{n} b_{ki} \hat{x}^{(l)}_{ij} \right),
$$

$$
k = 1,2,\ldots,K, \quad j = (l-1)L + 1;
$$

$$
s^{(j)}_{kj} \leftarrow s^{(j)}_{kj}, k = 1,2,\ldots,K, \ j = (l-1)L + 2,\ldots,lL.
$$

$$
d^{(l)}_i = d_i - \sum_{l=1}^{(l-1)L} \hat{x}^{(l)}_{ij}, \ i = 1,2,\ldots n
$$

onde os $\hat{x}^{(l)}_{ij}$, l=1,..., (p – 1), que aparecem na formulação do *1*-ésimo bloco são as soluções ótimas obtidas para os (p – 1) blocos anteriores, isto é, na solução ótima, todas as folgas do último período de cada bloco são acrescentadas as disponibilidades do primeiro período do bloco seguinte, e todas as folgas das restrições de demanda, são as novas demandas para o bloco seguinte. O problema a ser resolvido será então,

$$
\begin{aligned}
&Maximizar \sum_{j=(l-1)L+1}^{pL} \sum_{i=1}^{n} c_{ij} x_{ij} \\
&s/a \sum_{j=(l-1)L+1}^{t} \sum_{i=1}^{n} b_{ki} x_{ij} \le \sum_{j=(l-1)L+1}^{t} s_{kt}^{(l-1)} \\
&\quad k = 1,2,\ldots,K \;\; e \;\; t = (l\text{-}1)L + 1,\ldots,lL. \\
&\quad \sum_{j=(l-1)L+1}^{lL} x_{ij} \le d_i^{(l)}, i = 1,2,\ldots,n \qquad\qquad (3) \\
&\quad \sum_{i=1}^{n} h_i x_{ij} \le g_j,\; j = (l-1)L + 1,..lL \\
&\quad x_{ij} \ge 0, i = 1,2,\ldots,n \;\; e \;\; j = (l-1)L + 1,\ldots,lL
\end{aligned}
$$

Uma observação importante aqui é que a matriz das restrições permanece invariável ao longo de cada bloco de *L* períodos, como já tinha sido percebido em Chu (1995). Este fato será da maior importância, tanto no que diz respeito a resolução do problema, quanto no que diz respeito a proposta do modelo de decomposição para o problema. O resultado fundamental desse trabalho é um teorema que relaciona a solução do problema (1) com a solução dada pelas soluções obtidas por meio da decomposição por blocos no horizonte de planejamento de *T* períodos em *k* blocos de tamanho fixo de *L* períodos, sendo *p* x *L*= *T*. Usando as convenções até aqui adotadas, e fazendo ainda,

$$
A = \begin{pmatrix} B \\ I \\ h \end{pmatrix}; b^{(j)} = \begin{pmatrix} \sum_{q=1}^{i} s^{(q)} \\ d \\ g^{(j)} \end{pmatrix}, j = 1,\ldots,T
$$

$$
com\; g^{(j)} = g_j + \sum_{q=1}^{j-1} hx^{(q)},\; j = 1,\ldots,T
$$

onde $\sum_{q=1}^{0} hx(q) = 0$. Cada subproblema (3) pode ser representado matricialmente como sendo o problema:

$$
\begin{aligned}
&Maximizar\;\; c^{(1)}x^{(1)} + c^{(2)}x^{(2)} + \ldots + c^{(p)}x^{(L)} \\
&s/a \quad Ax^{(1)} && \le b^{(1)} \\
&\quad Ax^{(1)} + Ax^{(2)} && \le b^{(2)} \\
&\quad \vdots && \qquad\qquad (4) \\
&\quad Ax^{(1)} + Ax^{(2)} + \ldots + Ax^{(L)} && \le b^{(L)} \\
&\quad x^{(1)} \ge 0, x^{(2)} \ge 0,\ldots, x^{(L)} \ge 0
\end{aligned}
$$

O resultado fundamental desse trabalho é o seguinte teorema que assegura que a decomposição proposta preserva a solução ótima do problema.

Seja x^* uma solução ótima para o problema (1), e seja $\hat{x}^{(l)}$ uma solução ótima para o subproblema (P_l), onde (P_l) é definido como no problema (3), onde) $\hat{x}^{(i)}$, *i* = 1, 2,3, ..., *p* – 1, é uma solução ótima para cada um dos blocos, e l é o número de blocos nos quais foi particionado o horizonte de planejamento *T*. Suponha, além disso, que a seguinte condição de regularidade seja satisfeita,

$$
\sum_{i=1}^{l-1} \hat{x}^{(i)} \le \sum_{i=1}^{l} x^{*(i)}
$$

onde $x^{*(i)}$ é cada um dos blocos da partição do vetor em n blocos, dos quais os (*l*–1) primeiros coincidem com as partições $\hat{x}^{(i)}$, i =1, 2, 3, ..., (*l*-1). Então $\hat{x}$ é uma solução ótima para o problema (1).

Prova: A prova dessa proposição pode ser feita facilmente usando-se a indução matemática, e não será apresentada aqui.

É importante observar que essa proposição assegura que qualquer solução ótima do problema de programação linear agregado pode ser obtida por meio das soluções ótimas dos subproblemas decorrentes da decomposição por blocos proposta para o problema primal. Esse fato permite resolver problemas de programação linear agregados, mesmo quando de grande porte, dado que podemos tomar o número de blocos em conformidade com os recursos computacionais disponíveis. Esse resultado se mantém, mesmo sem a relaxação de integralidade.

4. Resultados Numéricos e Conclusões

4.1 Resultados Numéricos

Para efeito de exposição do uso da abordagem de decomposição em casos práticos, vamos considerar uma empresa que produz cinco tipos de produtos, usa seis tipos de componentes diferentes nas BOM(s) dos produtos, e tem um horizonte de planejamento de doze períodos, Anexo A. Vamos supor que conhecemos os retornos associados a cada produto fabricado em cada um dos períodos do horizonte de planejamento, os tempos padrões de fabricação de cada produto, e as disponibilidades de recursos de mão-de-obra e de cada um dos tipos de componentes por período. Os parâmetros do problema não serão apresentados aqui, mas poderão ser disponibilizados ao interesse de qualquer pesquisador. As tabelas do Anexo A mostram os resultados obtidos para a decomposição em um bloco, dois blocos, três blocos, quatro blocos, seis blocos e doze blocos, respectivamente, onde *i* indica o produto e *j* o período em que foi produzido. Desse modo, a primeira tabela apresenta a solução do problema sem decomposição, a segunda tabela apresenta a solução do problema usando decomposição em dois blocos, e assim sucessivamente. A última tabela apresenta a solução usando-se total decomposição, isto é, cada período é um bloco na qual o problema foi decomposto. Usando-se o método simplex para resolver esse tipo de problema, nós teríamos, no pior caso, que o custo computacional é proporcional a 2^n unidades de tempo, onde n é o número de variáveis do problema. Assim o tempo de computação necessário para obter-se a primeira tabela é proporcional a 1.1529E+018, unidades de tempo. Para obter-se segunda tabela o tempo necessário é de 2.1475E+009, e assim sucessivamente, de modo que o custo computacional da obtenção da última tabela é proporcional a 384 unidades de tempo. Fica claro que a decomposição utilizada tem forte impacto sobre o tempo computacional necessário para obtenção da solução ótima do problema como alegado acima.

4.2 Conclusões

O modelo de programação de produção agregado é um problema de programação linear inteira de grande porte, esparso, e estruturado e, sem a relaxação da integralidade das variáveis torna-se de difícil solução numérica, mesmo em casos relativamente simples, com médio número de variáveis, e médio número de períodos no horizonte de planejamento. Resolver esse tipo de problema sem a relaxação da integralidade é um desafio enorme e na maioria das vezes mal sucedido. Por outro lado, dada a robustez do modelo de programação linear, a solução inteira obtida da solução linear é bastante satisfatória, e em muitos casos, a única disponível. É importante observar que a chave dessa decomposição é não permitir restrições de acoplamento, e desse modo, não permitir problemas principais com variáveis de acoplamento (LASDON, 1968), mas apenas subproblemas independentes. Em Chu (1995), relaxando o conjunto de restrições relativas a disponibilidade de recursos em cada período j=1, 2,3, ... T, e, também, relaxando a condição de integralidade, o autor mostrou, através de um número reduzido de experimentos, que a solução inteira obtida da solução linear permanece no entorno da solução ótima, o que justifica a relaxação da condição de integralidade. Um intensivo estudo do comportamento numérico desse problema no que diz respeito ao tempo computacional necessário para obtenção de solução ótima está em curso, e com resultados sempre muito promissores.

Referências

CHU, S.C.K. *A mathematical programming approach towards optimized master production scheduling.* Int. J. Production Economics 38 (1995) 269-279.

GODINHO FILHO, M.; FERNANDES, F.C.F. *Redução da instabilidade e melhoria de desempenho do sistema MRP.* Prod. v.16 n.1 São Paulo, jan./abr. 2006.

LASDON, L.S. *Duality and Decomposition in Mathematical Programming.* IEEE Transactions on Systems Science and Cybemetics, 4(2),86-100, 1968.

SAMPAIO, R.J.B.; VIEIRA, G.E.; FAVARETTO, F. *An Approach of Mathematical Programming to the Master Production Scheduling Problem,* Technical Report 2009.

SASTIM, O.; KOROGLU, S.; YUZUKIRMIZI, M.; ERSOZ, S. *Using Artificial Intelligence in Material Requirement Planning,* Proceeding of 5th Intemational Symposium on Intelligent Manufacturing System, May 29-31.2006 : 339-345, Sakarya University, Department of Industrial Engineering.

TAKEY, F.M.; MESQUITA, M.A. *Aggregate Planning for a Large Food Manufacturer with High Seasonal Demand*, Brazilian Joumal of Operations and Production Management Volume 3. Number 1. 2006, p. 05-20.

ANEXO A – Tabelas de resultados obtidos no planejamento de 12 períodos

Ordem das Tabelas: 1 bloco (sem decomposição), 2 blocos (Decomposição em 2 Blocos), 3 blocos (Decomposição em 3 Blocos), 4 blocos (Decomposição em 4 Blocos), 6 blocos (Decomposição em 6 Blocos) e 12 blocos (Decomposição em 12 Blocos).

Produtos	Períodos												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	27.059,676	27.100,299	36.003,461	23.493,149	18.785,538	25.185,925	19.003,686	22.452,899	28.561,750	25.452,998	31.532,163	32.640,635	317.272,178
2	-	49.185,661	35.195,011	23.310,274	17.521,211	23.371,786	11.782,055	27.692,706	27.249,924	25.212,411	30.288,850	39.395,664	310.205,552
3	5.471,050	4.114,086	9.885,237	6.043,786	3.880,038	4.738,724	3.073,963	3.799,411	6.249,100	5.237,588	7.957,675	8.488,839	68.939,496
4	-	-	-	-	-	-	72.814,897	8.522,336	11.138,161	11.211,549	11.870,250	15.352,238	130.909,430
5	5.277,586	18.441,662	17.757,999	11.591,663	8.513,549	11.203,113	8.640,725	9.920,024	12.621,349	13.014,174	13.916,075	19.437,950	150.335,866

Produtos	Períodos												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	27.059,676	11.188,610	51.915,150	-	42.278,686	25.185,925	19.003,686	-	51.014,649	5.755,597	51.229,564	32.640,635	317.272,178
2	22.372,861	26.812,800	15.169,236	43.336,049	17.521,211	23.371,786	18.056,061	21.418,700	-	52.462,335	30.288,850	39.395,664	310.205,552
3	5.471,050	-	13.999,324	6.043,786	3.880,038	4.738,724	3.073,963	-	10.048,511	5.237,588	7.957,675	8.488,839	68.939,496
4	9.547,500	-	-	37.870,211	7.444,086	9.979,513	-	-	-	38.845,633	11.870,250	15.352,238	130.909,430
5	10.311,424	13.407,825	17.757,999	11.591,663	8.513,549	11.203,113	8.640,725	9.920,024	12.621,349	13.014,174	13.916,075	19.437,950	150.335,866

Produtos	Períodos												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	27.059,676	27.100,299	36.003,461	23.493,149	18.785,538	-	44.189,611	22.452,899	28.561,750	25.452,999	31.532,163	32.640,636	317.272,178
2	22.372,861	26.812,800	35.195,011	23.310,274	17.521,211	23.371,786	18.056,061	21.418,700	27.249,924	25.212,411	30.288,850	39.395,664	310.205,552
3	5.471,050	6.256,938	7.742,386	6.043,786	3.880,038	-	7.812,686	3.799,411	6.249,100	5.237,588	7.957,675	8.488,839	68.939,496
4	-	20.990,013	16.003,824	10.423,875	7.444,086	9.979,513	7.973,588	8.522,336	-	22.349,710	11.870,250	15.352,238	130.909,430
5	10.311,424	13.407,825	3.897,026	25.452,635	8.513,549	11.203,113	8.640,725	9.920,024	-	25.635,522	13.916,075	19.437,950	150.335,866

Produtos	Períodos												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	27.059,676	27.100,299	22.142,489	37.354,121	-	43.971,463	19.003,686	22.452,899	28.561,750	25.452,999	31.532,163	32.640,635	317.272,178
2	22.372,861	26.812,800	35.195,011	23.310,274	6.794,606	34.098,392	18.056,061	21.418,700	27.249,924	25.212,411	30.288,850	39.395,664	310.205,552
3	5.471,050	6.256,938	7.742,386	6.043,786	-	8.618,761	3.073,963	3.799,411	6.249,100	5.237,588	7.957,675	8.488,839	68.939,496
4	9.547,500	11.442,513	16.003,824	10.423,875	-	17.423,599	-	-	27.634,085	11.211,549	11.870,250	15.352,238	130.909,430
5	10.311,424	13.407,825	17.757,999	11.591,663	8.513,549	11.203,113	8.640,725	9.920,024	12.621,349	13.014,174	13.916,075	19.437,950	150.335,866

Produtos	Períodos												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	27.059,676	27.100,299	36.003,461	23.493,149	18.785,538	25.185,925	19.003,686	22.452,899	28.561,750	25.452,999	31.532,163	32.640,635	317.272,178
2	22.372,861	26.812,800	35.195,011	23.310,274	17.521,211	23.371,786	18.056,061	21.418,700	27.249,924	25.212,411	30.288,850	39.395,664	310.205,552
3	5.471,050	6.256,938	7.742,386	6.043,786	3.880,038	4.738,724	3.073,963	3.799,411	6.249,100	5.237,588	7.957,675	8.488,839	68.939,496
4	-	20.990,013	-	26.427,699	-	17.423,599	-	16.495,924	-	22.349,710	11.870,250	15.352,238	130.909,430
5	10.311,424	13.407,825	17.757,999	11.591,663	8.513,549	11.203,113	8.640,725	9.920,024	12.621,349	13.014,174	13.916,075	19.437,950	150.335,866

Produtos	Períodos												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	27.059,676	27.100,299	22.142,489	37.354,121	18.785,538	25.185,925	19.003,686	22.452,899	28.561,750	25.452,999	31.532,163	32.640,635	317.272,178
2	22.372,861	26.812,800	35.195,011	23.310,274	17.521,211	23.371,786	18.056,061	21.418,700	27.249,924	25.212,411	30.288,850	39.395,664	310.205,552
3	5.471,050	6.256,938	7.742,386	6.043,786	3.880,038	4.738,724	3.073,963	3.799,411	6.249,100	5.237,588	7.957,675	8.488,839	68.939,496
4	9.547,500	11.442,513	16.003,824	10.423,875	7.444,086	9.979,513	7.973,588	8.522,336	11.138,161	11.211,549	11.870,250	15.352,238	130.909,430
5	10.311,424	13.407,825	17.757,999	11.591,663	8.513,549	11.203,113	8.640,725	9.920,024	12.621,349	13.014,174	13.916,075	19.437,950	150.335,866

2.2 ARTIGO 2

Este artigo foi aprovado no evento: 2010 Industrial Engineering Research Conference (IERC), que aconteceu em Cancun - México.

An Approach of Mathematical Programming to the Master Production Scheduling Problem

R.J.B. de Sampaio, G.E. Vieira, F. Favaretto, E.L. de Camargo

Department of Industrial Engineering,
Pontifical Catholic University of Parana,
80215-901 Curitiba PR, Brazil

Abstract

The Material Requirement Planing (MRP) and Master Production Scheduling (MPS) systems have gained a great deal of attention lately, see for instance [1], [5], [4], [3], and the references therein. However, they still need a lot of more research as soon as its aggregation aspects is the concern since it easily becomes a very large scale programming problem, hardly to be solved. In this paper a complete approach of decomposition, including all the constraints, is developed to deal with MPS problem, which mainly take into account its aggregation aspects, while preserving its mathematical tractability.

Keywords MPS, Large Scale Problem, Decomposition.

1 The Problem

In general the Decision Support Systems (DSS) in production engineering include a Material Requirement Planing (MRP) and a Master Production Scheduling (MPS) system. The MRP contain a set of rules, logically organized, articulated to support the MPS, related to the net quantities of material required to perform the production schedule. In this way we can imagine that the MRP may be completed declared in terms of Bill of Material (BOM) of each product. The mathematical problem resulting from the modelling of such a problem result in what we call aggregate optimization model [2]. For the cases where we should consider a large quantity of products, and for each product, a large number of components, the mathematical modelling and solution to this problem, using integer programming, is completely out of practical issue, since the complexity of the problem result in unacceptable computational time to reach an optimal solution [1]. However, since the execution of the model is periodically repeated to incorporate new periods, and if we think of it as one activity that must be repeated many times, it make sense to consider to solve this problem as a large scale linear programming problem, thus relaxing its integrality aspect. In this case, we must strongly use technics of decomposition to have its processing time acceptable. In [1], this problem was partially solved, but the author had to relax some constraints related to capacity, which brings some degree of unrealistic approach to this very real problem. In this work we propose a new decomposition schema for the MPS problem which consider all the constraints from the problem, and in this sense generalize the model presented in [1]. It is important to emphasize that the MPS defines the final products quantities, as well as when and where they will be made, and consequently, a good material planning scheduling is essential within the supply chain management. This work is organized as follows. In Section 2, we present the problem of aggregate model using linear programming, and the question of integer relaxation is discussed. In Section 3, we present a new block decomposition schema, and the main theorem of this paper is fully proved. Finally, in Section 4, the results are summarized, and some open questions are pointed out.

2 Model and Structure of the Problem

Lets suppose we have to consider the following product-mix problem. The products are indexed by $i = 1, 2, ..., n$, the component parts are indexed by $k = 1, 2, ...,m$, and the periods of planning horizon are indexed by $j = 1, 2, ..., T$. Let b_{ki} be the number of components of type k used to produce one unit of product i, and let h_i be the standard time required to produce one unit of product i. Furthermore, suppose that g_j is the labor resource (in units of standard time) available for period j. An important feature of resources g_j is that it can not be carried out to the next period. Lets call s_{kj} the available supply of components k at period j, and d_i, the maximum demand for product i at $j=T$. The profit income from each unit of product i produced at period j will be called x_{ij}. The mathematical modelling of the problem can be declared as,

$$
\begin{aligned}
&\textit{Maximize} \quad \sum_{j=1}^{T}\sum_{i=1}^{n} c_{ij} x_{ij} \\
&\textit{s/t} \quad \sum_{j=1}^{t}\sum_{i=1}^{n} b_{ki} x_{ij} \le \sum_{j=1}^{t} s_{kj} \ \ (Eq\text{-}A) \\
&\quad t = 1,2,\dots,T \quad e \quad k = 1,2,\dots,m \\
&\quad \sum_{j=1}^{T} x_{ij} \le d_i, \ i = 1,2,\dots,n \ \ (Eq-B) \qquad\qquad (1) \\
&\quad \sum_{i=1}^{n} h_i x_{ij} \le g_j, \ j = 1,2,\dots,T \ \ (Eq-C) \\
&\quad x_{ij} \ge 0, i = 1,2,\dots,n; \textit{and} \ \ j = 1,2,\dots,T
\end{aligned}
$$

It can be seen that if T is a finite integer number of periods, we should require the decision variable x_{ij} is to be integer. However, since we only will consider T as being the planning horizon, which will be repeated again and again, thus for practical uses it means that T last for ever, and then it make sense to relax the integrality condition to consider the problem as a large scale linear programming problem. In fact this strong assumption is also supported by some numerical experiments related by [1].

Now let say a word about the mathematical model, that is, the set of equations (Eq-A), (Eq-B), (Eq-C), stated above. The system of equations (Eq-A), define the aggregation of the model, since for t = 1, 2, ..., T, it aggregate the production as well as the available stock of components s_{kj} . This aggregation turns the problem to be of large scale, but also sparse and structured. The set of equations (Eq-B), state that the production must not overpass the maximum predicted demand, and (Eq-C) gives bound for the resources of available labor at each period j. It is worth to say that such a resource can not be carried out from one period to another.

In the system of equations (Eq-A), for each k = 1, 2, ...m, the $\sum_i b_{ki} x_{ij}$ gives the net requirements of each component k, necessary to produce the product i at period j, which means that it gives the MRP solution to the model.

The way the problem is stated my suggest the possibility of a standard algorithm carry to the last period, $j = T$, the maximum quantity of requiring production, and then becoming infeasible. Against this filling we have to consider two arguments. First, the existence of solution (or not existence) for problem (1) depend only on the fulfillment of the hypothesis of Weierstrass theorem [7], which say that all continuous function defined on a closed and bounded set attains its maximum and minimum in the set,and the second: even when the solution exist can an standard algorithm always reach it? The answer for this last question comes from the fact that the problem is a maximizer problem, and due the signal of the constraints, the algorithm always tries to produce as much as it is possible by the resources.

The problem (1) as formulate is certainly a large scale linear programming problem, even for some simple cases, since the problem has $n\times T$ decision variables, $m\times T + n + T$ linear constraints, and $n\times T$ variable constraints, where n, m, and T, stand respectively by the number of products, the number of different components, and the number of periods. Even for a simple case of an enterprize which manufactory a 100 different products, using 200 different components, for a planing horizon of 24 months, would give rise a problem with 2400 decision variables, 5100 linear constraints, and 2400 variable constraints. It is ease to imagine the trouble this would cause if we have to solve it as an integer problem. This kind of computational complexity brought by the scale of the problem almost always require an appropriate approach to keep the numerical tractability of the problem in the right perspective. And here it means decomposition.

The approach of decomposition that will be considered here brings some stimulating results from the point of view of the problem solution, and also from the point of view of computational feasibility as well. The application of the technic perform as a reduction in the planing horizon amount of periods, and so, diminishing arbitrarily, the problem scale. As a matter of fact, the question raised up by decomposition is if this such approach modify the solution of the primitive aggregate linear problem. This will be the main question addressed in the next section.

1. $x^{(j)} = (x_{1j}, x_{2j}, ..., x_{nj})^T$, $j = 1, 2, ..., T$;
2. $B = (b_{ki})$, $k = 1, 2, ...,m$, $i = 1, 2, ..., n$;
3. $s^{(j)} = (s_{1j}, s_{2j}, ..., s_{mj})^T$, $j = 1, 2, ..., T$;
4. $d = (d_1, d_2, ..., d_n)^T$;
5. $c^{(j)} = (c_{1j}, c_{2j}, ..., c_{nj})$ $j = 1, 2, ..., T$;
6. $h = (h_1, h_2, ..., h_n)$;
7. I is the correspondent identity matrix.

It is worth to explicit here an particular arbitrary case where the planing horizon is taken to be $T = 2$, the number of components are $m = 2$, and the total type os products are $n = 2$. This will allow us to have a better understanding about the problem we are addressing, its sparsity, and it structure as well. In this case the model is the following,

$$
\begin{array}{ll}
\textit{Maximize} & c_{11}x_{11}+c_{12}x_{12}+c_{21}x_{21}+c_{22}x_{22}; \\
\textit{s/t} & \\
b_{11}x_{11}+b_{12}x_{21} & \le s_{11}; \\
b_{21}x_{11}+b_{22}x_{21} & \le s_{21}; \\
b_{11}x_{11}+b_{12}x_{21}+b_{11}x_{12}+b_{12}x_{22} & \le s_{11}+s_{12}; \\
b_{21}x_{11}+b_{22}x_{21}+b_{21}x_{12}+b_{22}x_{22} & \le s_{21}+s_{22}; \\
x_{11} & \le d_1; \\
x_{11} \quad +x_{12} & \le d_1; \\
x_{21} & \le d_2; \\
x_{21} \quad +x_{22} & \le d_2; \\
h_1x_{11}+h_2x_{21} & \le g_1; \\
h_1x_{12}+h_2x_{22} & \le g_2; \\
x_{11}\ge 0; x_{12}\ge 0; x_{21}\ge 0; x_{22}\ge 0 &
\end{array}
$$

It is quite obvious that the constraint matrix is structured and sparse. The approach of decomposition to be used here take advantage from this structure in the way described bellow, which include that one given in [1]. Assuming the above notation, this aggregate linear programming can be stated as,

$$
\begin{array}{lll}
\textit{Maximize} & c^{(1)}x^{(1)}+c^{(2)}x^{(2)} & \\
\textit{s/t} & Bx^{(1)} & \le s^{(1)}; \\
 & Bx^{(1)}+Bx^{(2)} & \le s^{(1)}+s^{(2)}; \\
 & Ix^{(1)} & \le d; \\
 & Ix^{(1)}+Ix^{(2)} & \le d; \\
 & hx^{(1)} & \le g_1; \\
 & \quad hx^{(2)} & \le g_2; \\
 & x^{(1)}\ge 0; x^{(2)}\ge 0 &
\end{array}
$$

or more specifically,

$$
\begin{array}{lll}
\textit{Maximize} & c^{(1)}x^{(1)}+c^{(2)}x^{(2)} & \\
\textit{s/t} & Bx^{(1)} & \le s^{(1)}; \\
 & Ix^{(1)} & \le d; \\
 & hx^{(1)} & \le g_1; \\
 & Bx^{(1)}+Bx^{(2)} & \le s^{(1)}+s^{(2)}; \\
 & Ix^{(1)}+Ix^{(2)} & \le d; \\
 & 0x^{(1)}+hx^{(2)} & \le g_2; \\
 & x^{(1)}\ge 0; x^{(2)}\ge 0 &
\end{array} \tag{2}
$$

In general we can approach this problem in the following way. First, consider that the T periods are divided into blocks of size L, and that the components not utilized in one period my be utilized in the next, and that the demands not achieved can be transferred to the next period L, and so on until the last block. Then the mathematical model for two blocks will be,

1. For the first block of L periods.

- Assign,

$$s_{kj}^{(1)} \leftarrow s_{kj}, k = 1,2,\dots,m, \; j = 1,2,\dots,L;$$
$$d_i^{(1)} \leftarrow d_i, i = 1,2,\dots,n,$$

- Solve,

$$\begin{aligned} Maximize \quad & \sum_{j=1}^{L}\sum_{i=1}^{n} c_{ij}x_{ij} \\ s/t \quad & \sum_{j=1}^{t}\sum_{i=1}^{n} b_{ki}x_{ij} \le \sum_{j=1}^{t} s_{kj}^{(1)} \\ & t = 1,2,\dots,L \;\; e \;\; k = 1,2,\dots,m. \\ & \sum_{j=1}^{L} x_{ij} \le d_i^{(1)}, i = 1,2,\dots,n \\ & \sum_{i=1}^{n} h_i x_{ij} \le g_j, \;\; j = 1,2,\dots,L \\ & x_{ij} \ge 0, i = 1,2,\dots,n; \;\; j = 1,2,\dots,L \end{aligned}$$

2. For the last block of L periods.

- Assign,

$$\begin{aligned} & s_{k,L+1}^{(2)} \leftarrow s_{k,L+1} + \sum_{j=1}^{L}(s_{kj}^{(1)} - \sum_{i=1}^{n} b_{ki}\hat{x}_{ij}), \\ & \quad k = 1,2,\dots,m, \;\; j = 1,2,\dots,L; \\ & s_{k,j}^{(2)} \leftarrow s_{kj}, k = 1,2,\dots,m, \qquad j = L+2,\dots,2L \\ & d_i^{(2)} \leftarrow d_i - \sum_{j=1}^{L}\hat{x}_{ij}, i = 1,2,\dots,n \end{aligned}$$

- Solve,

$$\begin{aligned} Maximize \quad & \sum_{j=L+1}^{2L}\sum_{i=1}^{n} c_{ij}x_{ij} \\ s/t \quad & \sum_{j=L+1}^{t}\sum_{i=1}^{n} b_{ki}x_{ij} \le \sum_{j=L+1}^{t} s_{kj}^{(2)} \\ & t = L+1,\dots,2L; k = 1,2,\dots,m. \\ & \sum_{j=L+1}^{2L} x_{ij} \le d_i^{(2)}, i = 1,2,\dots,n \\ & \sum_{i=1}^{n} h_i x_{ij} \le g_j, \;\; j = L+1,\dots,2L \\ & x_{ij} \ge 0, i = 1,2,\dots,n \quad e \;\; j = L+1,\dots,2L, \end{aligned}$$

where the $\hat{x}_{ij}$ that appear in the formulation of last block is a solution we have got in the first block. In general, for the p^{th} block, p≥2, the subproblems can be formulate like: Assign,

$$\begin{aligned} & s_{k,(p-1)L+1}^{(p)} \leftarrow s_{k,(p-1)L+1} + \sum_{l=1}^{p-1} \sum_{j=(l-1)L+1}^{lL} (s_{kj}^{(l)} - \sum_{i=1}^{n} b_{ki}\hat{x}_{ij}^{(l)}), \\ & \quad k = 1,2,\dots,m; \qquad j = pL+2,\dots,pL; \\ & s_{k,j}^{(p)} \leftarrow s_{kj}, k = 1,2,\dots,m, \qquad j = pL+2,\dots,pL \\ & d_i^{(p)} = d_i - \sum_{l=1}^{(p-1)L} \hat{x}_{ij}^{(l)}, i = 1,2,\dots,n \end{aligned}$$

where the $\hat{x}_{ij}^{(l)}$, $l = 1,\dots,(p-1)$, that appear in the formulation of p^{th} block are the solutions we have got for the first $(p-1)^{th}$ blocks. Then the problem to be solved will be,

$$\begin{aligned} &Maximize \quad \sum_{j=(p-1)L+1}^{pL}\sum_{i=1}^{n} c_{ij}x_{ij} \\ &s/t \quad \sum_{j=(p-1)L+1}^{t}\sum_{i=1}^{n} b_{ki}x_{ij} \le \sum_{j=(p-1)L+1}^{t} s_{kj}^{(p-1)}, \\ &t=(p-1)L+1,\ldots,pL \quad e \quad k=1,2,\ldots,m \\ &\sum_{j=(p-1)L+1}^{pL} x_{ij} \le d_i^{(p)}, i=1,2,\ldots,n \\ &\sum_{i=1}^{n} h_i x_{ij} \le g_j, \; j=(p-1)L+1,\ldots,pL \\ &x_{ij}\ge 0, i=1,2,\ldots,n; \; j=(p-1)L+1,\ldots,pL. \end{aligned}$$

A main observation to be made here is that the matrix of constraints remain unchangeable as long as the size of periods L is kept constant. This fact is important as long the solution of the problem is the concern and also because the approach of decomposition used here, as well. The main result we present here will be a theorem which relate the solution to problem (1) and the solutions obtained through the subproblems coming from the proposed decomposition. Before the statement of general theorem lets see it version for a particular case where the T periods are divided into two blocks of size L. This will help us to have a deep intuition on the process of decomposition and how to generalize it, as well. Using the notation adopted so far, and making,

$$A=\begin{pmatrix} B \\ I \\ H \end{pmatrix}; b^{(j)}=\begin{pmatrix} \sum_{q=1}^{j} s^{(q)} \\ d \\ g^{(j)} \end{pmatrix}, j=1,\ldots,T,$$

$$where \;\; g^{(j)} = g_j + \sum_{q=1}^{j-1} hx^{(q)}, j=1,\ldots,T,$$

and $\sum_{q=1}^{0} hx^{(q)}=0$, we my represent the problem (2) as being the problem,

$$\begin{aligned} &Maximize \quad c^{(1)}x^{(1)}+c^{(2)}x^{(2)} \\ &s/t \quad Ax^{(1)} \le b^{(1)} \\ &Ax^{(1)}+Ax^{(2)} \le b^{(2)} \\ &x^{(1)}\ge 0, x^{(2)}\ge 0 \end{aligned}$$

Lets call problem (P_1), the problem,

$$\begin{aligned} &Maximize \quad c^{(1)}x^{(1)} \\ &s/t \quad Ax^{(1)} \le b^{(1)} \\ &x^{(1)} \ge 0, \end{aligned}$$

and lets call problem (P_2) the problem,

$$\begin{aligned} &Maximize \quad c^{(2)}x^{(2)} \\ &s/t \quad Ax^{(2)} \le b^{(2)} - A\hat{x}^{(1)} \\ &x^{(2)} \ge 0, \end{aligned}$$

where $\hat{x}^{(1)}$ is an optimal solution to problem (P_1). As we can see, the problem (2) is a particular instance of problem (1), and the subproblems (P_1) and (P_2) are the decomposition of problem (2) into two blocks of size L. The main result of this work is a generalization of the following proposition,

Proposition 1 *Let (x^*, y^*) be an optimal solution to problem (2), and let* $\hat{x}$ *be an optimal solution to problem (P_1), where* $\hat{x} \le x^*, y^*$. *If* $\hat{y}$ *is an optimal solution to problem (P_2),then (* $\hat{x}$ *,* $\hat{y}$ *) is an optimal solution to problem* (2).

Proof. If (x^*, y^*) is an optimal solution to problem (2) then,

1. $Ax^* \le b^{(1)}$, and $x^* \ge 0$;
2. $Ax^* + Ay^* \le b^{(2)}$, $y^* \ge 0$, and,
3. $\forall(x, y)$ feasible, $c^{(1)}x^* + c^{(2)}y^* \ge c^{(1)}x + c^{(2)}y$.

Let suppose that $\hat{x} \ge 0$ is an optimal solution to problem (P_1), with $\hat{x} \le x^* + y^*$. Latter we discuss this regularity condition which is essential to support the hypothesis that the existence of solution to problem (P_1) does not imply the nonexistence of solution to problem (2).Then,

1. $A\hat{x} \le b^{(1)}$, $\hat{x} \ge 0$, and (I)
2. $\forall x$ feasible, $c^{(1)}\hat{x} \ge c^{(1)}x$

Observe that since $\hat{x}$ is a feasible solution to problem (P_1), then $\hat{x}$ is also feasible for the first set of constraints of problem (2). Now let $\hat{y} \ge 0$ be an optimal solution to problem (P_2), then,

1. $A\hat{y} \le b^{(2)} - A\hat{x}$, $y \ge 0$, and (II)
2. $\forall y$ feasible, $c^{(2)}\hat{y} \ge c^{(2)}y$

Putting (I) and (II) together, we have that ($\hat{x}$, $\hat{y}$)is a feasible solution to problem (2) and then $c^{(1)}x^*+c^{(2)}y^* \ge c^{(1)}\hat{x} + c^{(2)}\hat{y}$. Beside, since x^* is a feasible solution to problem (P_1), then $c^{(1)}\hat{x} \ge c^{(1)}x^*$, and since y^* is a feasible solution to problem (P_2), then $c^{(2)}\hat{y} \ge c^{(2)}y^*$, and then $c^{(1)}x^* + c^{(2)}y^* \le c^{(1)}\hat{x} + c^{(2)}\hat{y}$, from where we conclude that ($\hat{x}$, $\hat{y}$) is also an optimal solution to problem (2). Now we justify the required assumption of regularity condition $\hat{x} < x^* + y^*$.
Let suppose by the contrary that $\hat{x} > x^* + y^*$ then, $\forall\, c^{(1)} \ge 0$, and $c^{(1)} \ne 0$, $c^{(1)}\hat{x} > c^{(1)}x^* + c^{(1)}y^*$, where $c^{(1)} \ge 0$ means that all the components of the vector are nonnegative, and at least one is positive.
Furthermore,

$$c^{(1)}\hat{x} + c^{(2)}\hat{y} > c^{(1)}x^* + c^{(1)}y^* + c^{(2)}\hat{y} \ge c^{(1)}x^* + c^{(1)}y^* + c^{(2)}y^* > c^{(1)}x^* + c^{(2)}y^*,$$

from where we conclude that (x^*, y^*) can not be an optimal solution to problem (2).
The general result is the following,

Proposition 2 *Let x^* be an optimal solution to problem (1) and let $\hat{x}^{(n)}$ be an optimal solution to subproblem (P_n), defined by,*

$$\begin{array}{ll} Maximize & c^{(n)}x^{(n)} \\ s/t & Ax^{(n)} \le b^{(n)} - \sum_{i=1}^{n-1} A\hat{x}^{(i)} \\ & x^{(n)} \ge 0, \end{array} \qquad (3)$$

where $\hat{x}^{(i)}, c^{(i)}, b^{(i)}$, $i=1,2,\ldots,n-1$, are respectively an optimal solution, the cost vector, and the vector of resources, to the corresponding subproblems. Furthermore, suppose the following regularity condition is satisfied,

$$\sum_{i=1}^{n-1} \hat{x}^{(i)} \le \sum_{i=1}^{n} x^{*(i)}$$

then, $\hat{x} = (\hat{x}^{(1)}, \ldots, \hat{x}^{(n)})^T$, is an optimal solution to problem (1).

Proof. The proof of this proposition will be made using mathematical induction. First it must be proved that the result is true for two blocks (Proposition 1), and then, supposing the result is true for any k blocks, we will prove it is true for k + 1. Thus the proposition is true for any integer number of blocks k, $2 \leq k \leq T$.

By the induction hypothesis the planing horizon T will become divided into k blocks, and if $\tilde{x}^{(n)}, n=1,2,\ldots,k$, is a given optimal solution to the subproblems (P_n), n=1, 2, ...,k, then $\tilde{x}^{(n)} = (\tilde{x}^{(1)}, \tilde{x}^{(2)}, \ldots, \tilde{x}^{(k)})$, is an optimal solution to problem (1). Furthermore, the following regularity condition is in order,

$$\sum_{i=1}^{k-1} \tilde{x}^{(i)} \leq \sum_{i=1}^{k} x^{*(i)}$$

where $x^{*(i)}$ is the corresponding optimal solution to each one of the blocks from partition of x^* into k blocks, in such a way that the $(k-1)^{\text{th}}$ first blocks partitions coincide with those partitions of $\tilde{x}^{(i)}$, i = 1, 2, ..., k-1.

Lets suppose now that $\hat{x} = (\hat{x}^{(1)}, \hat{x}^{(2)} \ldots, \hat{x}^{(k+1)})$ is a vector of optimal solutions to the subproblems (P_n), n = 1, 2, ..., k+1, satisfying the corresponding regularity condition,

$$\sum_{i=1}^{k} \hat{x}^{(i)} \leq \sum_{i=1}^{k+1} x^{*(i)}$$

where $x^{*(i)}$, i = 1, 2, ..., k + 1, are the corresponding partition of vector $x^{*(i)}$ into k + 1 blocks, in such a way that the first k blocks coincide with the partition of $\tilde{x}^{(i)}$, i = 1, 2, ..., k. We will prove that one such a vector $\hat{x}$ is an optimal solution to problem (1).

First, lets suppose that the vector $\tilde{x}$, which by hypothesis of induction is an optimal solution to problem (1), be partitioned into k + 1 blocks, overall similar to the partitioning of the vector $\hat{x}$, then $\tilde{x} = (\tilde{x}^{(1)}, \tilde{x}^{(2)}, \ldots, \tilde{x}^{(k+1)})$, and

$$\sum_{i=1}^{k} \hat{x}^{(i)} \leq \sum_{i=1}^{k+1} \tilde{x}^{(i)}$$

The resulting blocks from the partitioning of $\tilde{x}$ into (k+1) blocks clearly gives feasible solutions to the subproblems (P_n), n=1,2, ..., k+1. However, since $\hat{x}^{(n)}$, n = 1, 2, ..., k+1, are optimal solutions to subproblems (P_n), n = 1, 2, ..., k+1, then,

$$\sum_{i=1}^{k+1} c^{(i)} \hat{x}^{(i)} \geq \sum_{i=1}^{k+1} c^{(i)} \tilde{x}^{(i)}$$

Thus, given that $\tilde{x}$ is an optimal solution to problem (1), then so is $\hat{x}$. The requirement for fulfilments of regularity condition is similar to that one of Proposition 1.

It is important to observe that this proposition assure that any optimal solution to the aggregate linear programming problem can be obtained by the rule of proposed decomposition, and that this fact allow to solve very large scale aggregate linear programming problems.

4 Numerical Experiments and Conclusions

4.1 Numerical Experiments

In order to show how the approach of decomposition perform numerically, lets consider a Master Production Scheduling for an Company which manufacture three kind of products, using three kind of components, for a planing horizon of six periods. Furthermore, lets suppose we know the incomes from each product manufactured in each one of the periods in the planing horizon, the stand time for manufacture each one of the products, and the available handwork resources at each period. The tables below show the optimal solution for the problem with and without decomposition, where Product stands for the product to be manufactured and Period stands for the period where it is manufactured.

Table 1 – 1 Block (without decoposition)

Product / Period	1	2	3	4	5	6	Total
1	0	12	0	0	20	8	40
2	16.43	1.75	6.25	3.57	52	0	80
3	5.71	0	0	34.29	0	0	40

Table 2 – 2 Blocks

Product / Period	1	2	3	4	5	6	Total
1	0	0	0	40	0	0	40
2	25	0	0	5	3.57	46.43	80
3	0	10	10	0	14.29	5.71	40

Table 3 – 3 Blocks

Product / Period	1	2	3	4	5	6	Total
1	0	0	0	0	0	40	40
2	25	0	0	12.5	7.5	35	80
3	0	10	10	20	0	0	40

Table 4 – 6 Blocks

Product / Period	1	2	3	4	5	6	Total
1	0	0	0	0	0	40	40
2	25	0	0	12.5	12.5	30	80
3	0	10	10	20	0	0	40

4.2 Conclusions

In general the Master Production Scheduling model is a large scale structured and sparse integer programming problem, which is hardly complex to be solved without integrality relaxation to linear programming problem. Even in modest cases, solve this integer problem is quite challenging, and we fail in most of the cases [1]. However in many cases we my consider this problem as being a large scale linear programming problem, and in these cases the problem can be solved using polynomial time algorithms.
As soon we consider to solve the problem as being a large scale linear programming problem, decomposition is in order. First, because we reduce the needs of computational storage, and second, because we reduce computational time, which is obvious in general for decomposition approach. So, for resume, the approach of decomposition presented here allow to solve large scale problems of MPS, which perform a key role on the field of industrial engineering, enlarging our capacite to solve practical problems. Since this algorithm is polinomial time in the variables, it is easy to understanding the gain in terms of complexity brought by decomposition. Un intensive study about

References

[1] Sydney C. K. Chu, *A mathematical programming approach towards optimized master production scheduling*, Int. J. Production Economics 38(1995) 269-279.

[2] Sydney C. K. Chu, *Optimal master production scheduling in a flexible manufacturing system: the case of total aggregation*, Proc. of the First Conf. on the Operational Research Society of Hong Kong, pp 103-108, 1991.

[3] Ozgur Sastim, Serkan Koroglu, Mustafa Yuzukirmizi, Suleyman Ersoz, *Using Artificial Intelligence in Material Requirement Planning*, Proceeding of 5th International Symposium on Intelligent Manufacturing System, May 29-31, 2006 : 339-345, Sakarya University, Department of Industrial Engineering.

[4] Moacir Godinho Filho, Flavio Cesar Faria Fernandes, *Redução da instabilidade e melhoria de desempenho do sistema MRP*, Prod. v.16 n.1 So Paulo jan./abr. 2006.

[5] Flávia M. Takey, Marco A. Mesquita, *Aggregate Planning for a Large Food Manufacturer with High Seasonal Demand*, Brazilian Journal of Operations and Production Management Volume 3, Number 1, 2006, pp. 05-20.

[6] L. S. Lasdon, *Duality and Decomposition in Mathematical Programming*, IEEE Transactions on Systems Science and Cybernetics, 4(2),86-100, 1968.

[7] Arch W. Naylor and George R. Sell, *Linear Operator Theory in Engineering and Science*, Spring-Verlag, New York, 1982.

2.3. ARTIGO 3

Este artigo está em processo de submissão a um periódico.

An Approach of Mathematical Programming to Large Master Production Scheduling Problem

R. J. B. de Sampaio*, Wenyu Sun†, E. L. de Camargo ‡
Pontifical Catholic University of Paraná - PPGEPS

Abstract

Only occasionally the Material Requirement Planing (MRP) and Master Production Scheduling (MPS) systems gain an approach of exact Mathematical Programming Optimization (MPO), see the survey [3] and references therein ([5],[10],[14],[21],[24],[40],[45]), for single-stage, and references ([5],[6],[18],[28],[29],[30],[34]), for multiple stage problems. It is eventually because the MPS is a very large integer programming problem, hardly to be solved, even when its integrality is relaxed. However, it is a key problem in industrial engineering and in fact deserve a lot of more attention. The mathematical model for this problem easily becomes a burden scale programming problem, even when applied to medium enterprizes ([4], [6]). In this paper the approach of mathematical decomposition used to solve the MPS mathematical model allow to solve optimally large scale MPS problems at the only cost of solving some small simple linear programming problems, which turn into our perspective to solve problems we did not could even consider before. The main contribution brought by this paper is how actually decomposition perform in practice, when the MPS is relaxed for Linear Programming Problem.

Keywords: MPS, Aggregate Large Scale Problem, Model Decomposition.

1 The Problem

In general, Decision Support Systems (DSS) in production engineering include a MRP and a MPS system. The MRP contain a set of rules, logically organized, articulated to support the MPS, related to the net quantities of material required to perform the production schedule, and in this way we can imagine that the MRP may be completed declared in terms of the Bill of Material (BOM) of each product. The mathematical problem resulting from the modelling of such a problem result in what it was called an aggregate optimization model [7]. For the cases where we should consider a large quantity of products, and for each product, a large number of components, the mathematical modelling and solution to this problem, using integer programming, is completely out of practical issue, since the complexity of the problem require an unacceptable computational time to reach an optimal solution [6]. However, since the execution of the model is periodically repeated to incorporate new periods, and if we think of it as one activity that must be repeated many times, it make sense to consider to solve this problem as a large scale linear or nonlinear programming problem, thus relaxing its integrality aspect, see ([4],[5],[6]). In the case of large scale linear programming problem we must strongly use technics of mathematical decomposition to have its computer processing time, acceptable. In [6], this problem was partially solved, but the author had to relax the set of constraints related to capacity. In this work we propose a mathematical decomposition schema for the MPS problem which consider all the constraints, and in this sense generalize the model presented in [6]. It is important to emphasize that the MPS defines the final products quantities, as well as, when and where they will be made and, consequently, a good material planning scheduling issued, which is essential within the supply chain management ([8], [11], [10]). This work is organized as follows. In Section 2, Model and Structure of the Problem, we present the problem of aggregating model for MPS using linear programming, and the question of integer relaxation is shortly discussed. In Section 3, Block Decomposition, we present a new mathematical block decomposition schema, and the main theorem of this paper is fully proved. Finally, in Section 4, Numerical Experiments and Conclusions, the results are summarized, and some open questions are pointed out.

* Corresponding author: de.sampaio@brturbo.com.br
†Nanjing Normal University - China
‡Graduate Student at PPGEPS – PUCPR

2 Model and Structure of the Problem

Lets suppose we have to consider the following product-mix problem. The products are indexed by i = 1, 2, ..., n, the component parts are indexed by k = 1, 2, ...,K, and the periods of planning horizon are indexed by j = 1, 2, ..., T. Let b_{ki} be the number of components of type k used to produce one unit of product i, and let h_i be the standard time required to produce one unit of product i. Furthermore, suppose that g_j is the labor resource (in units of standard time) available for period j. Lets suppose that an important feature of resources g_j is that they can not be carried out to the next period. Lets call s_{kj} the available supply of components k at period j, and d_{ij}, the maximum demand for product i at period j. The income profit from each unit of product i produced at period j will be called c_{ij}, and the decision variable of how many products of the type i to be produced at period j will be called x_{ij} . The mathematical modelling of this problem can be declared as,

$$
\begin{aligned}
&\textit{Maximize} \quad \sum_{j=1}^{T}\sum_{i=1}^{n} c_{ij}x_{ij} \\
&s/t \quad \sum_{j=1}^{t}\sum_{i=1}^{n} b_{ki}x_{ij} \le \sum_{j=1}^{t} s_{kj},\ k=1,2,\ldots,K;\ t=1,2,\ldots,T \\
&\sum_{j=1}^{t} x_{ij} \le \sum_{j=1}^{t} d_{ij},\ i=1,2,\ldots,n;\ t=1,2,\ldots,T \\
&\sum_{i=1}^{n} h_i x_{ij} \le g_j,\ j=1,2,\ldots,T \\
&x_{ij} \ge 0, i=1,2,\ldots,n;\ j=1,2,\ldots,T
\end{aligned}
\tag{1}
$$

It can be seen that since T is a finite integer number of periods, then we should require the decision variable x_{ij} to be integer. However, since we only will consider T as being the planning horizon, which will be repeated shifted on, thus for practical uses it means that T will ever be reached, and then it make sense to relax the integrality condition to consider the problem as a large scale linear programming problem. In fact this strong assumption is also supported by some numerical experiments related in ([6],[4]). Now let say a word about the mathematical model of problem (1). The first and second set of constraints define the structure and sparsity of the mathematical model, since for t = 1, 2, ..., T, it aggregate the production and the available stock of components s_{kj} , from j = 1 to j = t, as well as the demands. This aggregation aspects turns out the problem to be of large scale, but also sparse and structured. The second set of constraints states that the production must not overpass the maximum predicted demand for each period, and finally the last set of constraints gives bound for the resources of available labor work for each period j. It is worth to say that such a resource can not be carried out from one period to another.

In the first set of constraints for each k = 1, 2, ...K, the $\sum_i b_{ki}x_{ij}$ gives the net requirements of each component k necessary to produce the demand of product i at period j, which means that adding on j = 1, 2, ..., T, gives the MRP solution to the mathematical model.

The problem (1) as formulated is certainly a large scale linear programming problem, and even for some simple cases, it tends to be a large scale problem since it has $n \times T$ decision variables, $K \times T + n + T$ linear constraints, and $n \times T$ constraints of positivity, where n, K, and T, stand respectively by the number of products, the number of different components, and the number of periods over the planing horizon. It is ease to imagine that if the number of products and components are large it is too hard to solve this problem for optimality as an integer problem. This kind of computational complexity always require an appropriate approach to keep the numerical tractability of the problem in the right perspective. And here it means mathematical decomposition.

The approach of mathematical decomposition that will be considered here brings some stimulating results from the point of view of the problem solution, and also from the point of view of computational feasibility as well. The application of this technic performs as a reduction in the planing horizon, and so, diminishing the problem scale. As a matter of fact the question raised up by the mathematical decomposition used here is if this such approach modify the optimal solution of the mathematical model or not, and this will be the main question addressed in the next section.

3 Block Decomposition

First we discuss how to make the block decomposition, and then, we discuss the questions related to optimality and computational complexity as well. To discuss block decomposition we begin by dividing the planing horizon T, into periods of fix size L, accordingly available computational capacity. The size of the blocks may vary from L = 1 to L = T, that is, from a total decomposition (L = 1) to an absence of decomposition (L = T). It is obvious that we can always suppose T as a multiple of L, and in this case the jacobian matrices of the constraint systems

will remains of the same size during all the process. Furthermore, as we are going to see, these matrices remain unchangeable during all the blocks, which allow reducing computational complexity by saving matrix decompositions.
From now on lets use the following notation,

1. $x^{(j)} = (x_{1j}, x_{2j}, ..., x_{nj})^T$, $j = 1, 2, ..., T$;
2. $B = (b_{ki})$, $k = 1, 2, ..., K$, $i = 1, 2, ..., n$;
3. $s^{(j)} = (s_{1j}, s_{2j}, ..., s_{Kj})^T$, $j = 1, 2, ..., T$;
4. $d^{(j)} = (d_{1j}, d_{2j}, ..., d_{nj})^T$, $j = 1, 2, ..., T$;
5. $c^{(j)} = (c_{1j}, c_{2j}, ..., c_{nj})$ $j = 1, 2, ..., T$;
6. $h = (h_1, h_2, ..., h_n)$;
7. I is the correspondent identity matrix of size n.

This will allow to write the problem (1) above in a matricial format, which is easier to manipulate. Just to show how the structure of the problem looks like, lets explicit here a particular case where the planing horizon is taken to be $T = 2$, the number of different components are $K = 2$, and the total type of products are $n = 2$. This will allow us to have a better understanding about the problem we are addressing, its structure, and its sparsity as well. In this case the mathematical model (1) becomes,

$$\begin{array}{lll}
\textit{Maximize} & c_{11}x_{11}+c_{12}x_{12}+c_{21}x_{21}+c_{22}x_{22}; & \\
\textit{s/t} & b_{11}x_{11}+b_{12}x_{21} & \leq s_{11}; \\
 & b_{21}x_{11}+b_{22}x_{21} & \leq s_{21}; \\
 & b_{11}x_{11}+b_{12}x_{21}+b_{11}x_{12}+b_{12}x_{22} & \leq s_{11}+s_{12}; \\
 & b_{21}x_{11}+b_{22}x_{21}+b_{21}x_{12}+b_{22}x_{22} & \leq s_{21}+s_{22}; \\
 & x_{11} & \leq d_{11}; \\
 & x_{11} \quad +x_{12} & \leq d_{11}+d_{12}; \\
 & x_{21} & \leq d_{21}; \\
 & x_{21} \quad +x_{22} & \leq d_{21}+d_{22}; \\
 & h_1x_{11}+h_2x_{21} & \leq g_1; \\
 & h_1x_{12}+h_2x_{22} & \leq g_2; \\
 & x_{11}\geq 0; x_{12}\geq 0; x_{21}\geq 0; x_{22}\geq 0 &
\end{array}$$

It is quite obvious that the constraint matrix is structured and sparse, and using the mentioned notation the problem can be arranged as,

$$\begin{array}{lll}
\textit{Maximize} & c^{(1)}x^{(1)}+c^{(2)}x^{(2)} & \\
\textit{s/t} & Bx^{(1)} & \leq s^{(1)}; \\
 & Ix^{(1)} & \leq d^{(1)}; \\
 & hx^{(1)} & \leq g_1; \\
 & Bx^{(1)}+Bx^{(2)} & \leq s^{(1)}+s^{(2)}; \\
 & Ix^{(1)}+Ix^{(2)} & \leq d^{(1)}+d^{(2)}; \\
 & 0x^{(1)}+hx^{(2)} & \leq g_2; \\
 & x^{(1)}\geq 0; x^{(2)}\geq 0 &
\end{array} \tag{2}$$

The model itself suggest how to decompose. First, consider that the T periods from the planing horizon are divided into blocks of fixed size L. Second, assume that the components not utilized in one block of L periods can be utilized in the next block and that the demands not achieved can be transferred to the next block of period L, as well.
Then, the mathematical model for two blocks will be,

1. For the first block of L periods, assign,

$$s^{(j)} \leftarrow (s_{1j}, s_{2j}, \ldots, s_{Kj})^T, \; j = 1,2,\ldots,L;$$
$$d^{(j)} \leftarrow (d_{1j}, d_{2j}, \ldots, d_{nj})^T, \; j = 1,2,\ldots,L,$$

and then solve,

$$\begin{aligned} &\textit{Maximize} && \sum_{j=1}^{L} \sum_{i=1}^{n} c_{ij} x_{ij} \\ &s/t && \sum_{j=1}^{t} \sum_{i=1}^{n} b_{ki} x_{ij} \le \sum_{j=1}^{t} s^{(j)}, \; k = 1,2,\ldots,K; \; t = 1,2,\ldots,L \\ &&& \sum_{j=1}^{t} x_{ij} \le \sum_{j=1}^{t} d^{(j)}, \; t = 1,2,\ldots,L \\ &&& \sum_{i=1}^{n} h_i x_{ij} \le g_j, \; j = 1,2,\ldots,L \\ &&& x_{ij} \ge 0, i = 1,2,\ldots,n; \; j = 1,2,\ldots,L \end{aligned}$$

2. For the second block of L periods, assign,

$$s^{(L+1)} \leftarrow s^{(L+1)} + (s^{(L)} - B\hat{x}^{(1)}),$$
$$s^{(j)} \leftarrow (s_{1j}, s_{2j}, \ldots, s_{Kj})^T, \; j = L+2, \ldots, 2L$$
$$d^{(j)} \leftarrow d^{(j)} + (d^{(L)} - \hat{x}^{(1)}), \; j = L+1, \ldots, 2L$$

and then solve,

$$\begin{aligned} &\textit{Maximize} && \sum_{j=L+1}^{2L} \sum_{i=1}^{n} c_{ij} x_{ij} \\ &s/t && \sum_{j=L+1}^{t} \sum_{i=1}^{n} b_{ki} x_{ij} \le \sum_{j=L+1}^{t} s^{(j)}, \; t = L+1, \ldots, 2L; k = 1,2,\ldots,K. \\ &&& \sum_{j=L+1}^{t} x_{ij} \le \sum_{j=L+1}^{t} d^{(j)}, \\ &&& i = 1,2,\ldots,n; t = L+1, \ldots, 2L; \\ &&& \sum_{i=1}^{n} h_i x_{ij} \le g_j, \; j = L+1, \ldots, 2L \\ &&& x_{ij} \ge 0, i = 1,2,\ldots,n; \; j = L+1, \ldots, 2L, \end{aligned}$$

where $\hat{x}^{(1)}$ is an optimal solution for the first block. In general for the p^{th} block, $p \ge 2$, the subproblems can be formulate likewise; Assign,

$$s^{((p-1)L+1)} \leftarrow s^{((p-1)L+1)} + (s^{((p-1)L)} - \sum_{l=1}^{p-1} B\hat{x}^{(l)}),$$
$$s^{(j)} \leftarrow (s_{1j}, s_{2j}, \ldots, s_{Kj})^T, \; j = (p-1)L+2, \ldots, pL.$$
$$d^{(j)} \leftarrow d^{(j)} + (d^{((p-1)L)} - \hat{x}^{((p-1)L)}), \; j = (p-1)L+1, \ldots, pL,$$

where the $\hat{x}^{((p-l)L)}$ is an optimal solution for the $(p-1)^{\text{th}}$ block. Then, the problem to be solved will be,

$$\begin{aligned} &\textit{Maximize} && \sum_{j=(p-1)L+1}^{pL} \sum_{i=1}^{n} c_{ij} x_{ij} \\ &s/t && \sum_{j=(p-1)L+1}^{t} \sum_{i=1}^{n} b_{ki} x_{ij} \le \sum_{j=(p-1)L+1}^{t} s_{kt}^{(p-1)}, \\ &&& k = 1,2,\ldots,m; t = (p-1)L+1, \ldots, pL. \\ &&& \sum_{j=(p-1)L+1}^{pL} x_{ij} \le \sum_{j=(p-1)L+1}^{pL} d_{ij}^{(j)}, i = 1,2,\ldots,n \\ &&& \sum_{i=1}^{n} h_i x_{ij} \le g_j, \; j = (p-1)L+1, \ldots, pL \\ &&& x_{ij} \ge 0, i = 1,2,\ldots,n; \; j = (p-1)L+1, \ldots, pL. \end{aligned}$$

A main observation to be made here is that the matrix of constraints remain unchangeable as long as the size of periods L is kept constant.
An important result we present here is a theorem which relate the solution of problem (1) and the solutions obtained throughout the subproblems coming from the proposed decomposition. Before the statement of general theorem lets see its version for a particular case where the T periods are divided into two blocks of fixed size L. This will help us to shed some light on the process of decomposition as well as over the technic will be used to prove the general theorem. Using the notation adopted so far, and assuming,

$$A = \begin{pmatrix} B \\ I \\ H \end{pmatrix}; b^{(j)} = \begin{pmatrix} \sum_{q=1}^{j} s^{(q)} \\ d^{(j)} \\ g^{(j)} + \sum_{q=1}^{j-1} hx^{(q)} \end{pmatrix}, j = 1, \ldots, T,$$

where $\sum_{q=1}^{0} hx^{(q)} = 0$, problem (2) may be represented as,

$$\begin{array}{lll} \textit{Maximize} & c^{(1)}x^{(1)} + c^{(2)}x^{(2)} & \\ s/t & Ax^{(1)} & \le b^{(1)} \\ & Ax^{(1)} + Ax^{(2)} & \le b^{(2)} \\ & x^{(1)} \ge 0, x^{(2)} \ge 0 & \end{array}$$

Lets call (P_1), the problem,

$$\begin{array}{lll} \textit{Maximize} & c^{(1)}x^{(1)} & \\ s/t & Ax^{(1)} \le b^{(1)} & (P_1) \\ & x^{(1)} \ge 0, & \end{array}$$

and lets call (P_2),

$$\begin{array}{lll} \textit{Maximize} & c^{(2)}x^{(2)} & \\ s/t & Ax^{(2)} \le b^{(2)} - A\hat{x}^{(1)} & (P_2) \\ & x^{(2)} \ge 0, & \end{array}$$

where $\hat{x}^{(1)}$ is an optimal solution to problem (P_1). As it can be seen, problem (2) is a particular instance of problem (1), and the subproblems (P_1) and (P_2) are the decomposition of problem (2) into two blocks of size L. The main result of this work is a generalization of the following proposition,

Proposition 1 *Let (x^*, y^*) be an optimal solution to problem (2), and let $\hat{x}$ be an optimal solution to problem (P_1), where $\hat{x} \le x^*, y^*$. If $\hat{y}$ is an optimal solution to problem (P_2),then ($\hat{x}$, $\hat{y}$) is an optimal solution to problem* (2).

Proof. If (x^*, y^*) is an optimal solution to problem (2) then,

1. $Ax^* \le b^{(1)}$, and $x^* \ge 0$;
2. $Ax^* + Ay^* \le b^{(2)}$, $y^* \ge 0$, and,
3. $\forall (x, y)$ feasible, $c^{(1)}x^* + c^{(2)}y^* \ge c^{(1)}x + c^{(2)}y$.

Let suppose that $\hat{x} \ge 0$ is an optimal solution to problem (P_1), with $\hat{x} \le x^* + y^*$. Then,

1. $A\hat{x} \le b^{(1)}$, $\hat{x} \ge 0$, $\hat{x} \le x^* + y^*$, and (I)
2. $\forall x$ feasible, $c^{(1)}\hat{x} \ge c^{(1)}x$

Observe that since $\hat{x}$ is a feasible solution to problem (P_1), then $\hat{x}$ is also feasible for the corresponding set of constraints of problem (2). Now let $\hat{y} \ge 0$ be an optimal solution to problem (P_2), then,

1. $A\hat{y} \le b^{(2)} - A\hat{x}$, $\hat{y} \ge 0$, and (II)

2. $\forall\, y$ feasible, $c^{(2)}\hat{y} \geq c^{(2)} y$

Combining (I) and (II) we have that ($\hat{x}$, $\hat{y}$) is a feasible solution to problem (2) and $c^{(1)}x^{*}+c^{(2)}y^{*} \leq c^{(1)}\hat{x} + c^{(2)}\hat{y}$, since (x^{*}, y^{*}) is a feasible solution for the problems (P_1) and (P_2), respectively. Since by hypothesis (x^{*}, y^{*}) is an optimal solution to problem (2), $c^{(1)}x^{*} + c^{(2)}y^{*} \geq c^{(1)}\hat{x} + c^{(2)}\hat{y}$, and then $c^{(1)}x^{*} + c^{(2)}y^{*} = c^{(1)}\hat{x} + c^{(2)}\hat{y}$, from where we conclude that ($\hat{x}$, $\hat{y}$) is also an optimal solution to problem (2). Now we justify the required regularity condition $\hat{x} < x^{*} + y^{*}$.

Let suppose by the contrary, that $\hat{x} > x^{*} + y^{*}$, then, $\forall\, c^{(1)} \geq 0$, and $c^{(1)} \neq 0$, which means that all the components of the vector are nonnegative, and at least one is positive, $c^{(1)}\hat{x} > c^{(1)}x^{*} + c^{(1)}y^{*}$. Since $c^{(2)}\hat{y}^{(2)} > 0$, then, $c^{(1)}\hat{x} + c^{(2)}\hat{y}^{(2)} > c^{(1)}x^{*} + c^{(1)}y^{*} + c^{(2)}\hat{y}^{(2)} \geq c^{(1)}x^{*} + c^{(1)}y^{*} + c^{(2)}y^{*} > c^{(1)}x^{*} + c^{(2)}y^{*}$, from where we conclude that (x^{*}, y^{*}) can not be an optimal solution to problem (2). This contradiction complete the proof. The general result is the following,

Proposition 2 *Let x^{*} be an optimal solution to problem (1) and let $\hat{x}^{(n)}$ be an optimal solution to subproblem (P_n), n= 1, 2, ...,N, where (P_n) stands for,*

$$\text{(3)}\qquad \begin{aligned} &\textit{Maximize} && c^{(n)}x^{(n)} \\ &\quad s/t && Ax^{(n)} \leq b^{(n)} - \sum_{i=1}^{n-1} A\hat{x}^{(i)} \\ & && x^{(n)} \geq 0, \end{aligned}$$

N is the number of blocks, and $\hat{x}^{(i)}$, $b^{(i)}$, $i=1,2,...,n-1$, are respectively an optimal solution, the cost vector, and the vector of resources, to the corresponding subproblems, and $\sum_{i=1}^{0} A\hat{x}^{(i)} = 0$. Furthermore, suppose the following regularity condition is satisfied,

$$\sum_{i=1}^{n-1} \hat{x}^{(i)} \leq \sum_{i=1}^{n} x^{*(i)}$$

where $x^{*(i)}$ stands for the same partition accordingly $\hat{x}^{(i)}$. Then, $\hat{x} = (\hat{x}^{(1)},\ldots,\hat{x}^{(n)})^T$, is an optimal solution to problem (1).

Proof. The proof of this proposition will be made using mathematical induction. First it must be proved that the result is true for two blocks (Proposition 1), and then, supposing the result is true for any integer number k of blocks, we will prove it is true for $k + 1$. Thus the proposition is true for any integer number of blocks N, $2 \leq N \leq T$.

By our induction hypothesis the planing horizon T will become divided into k blocks, and let $\tilde{x}^{(n)}, n = 1,2,\ldots,k$, be a given optimal solution to the subproblems (P_n), n=1, 2, ...,k, satisfying the regularity condition

$$\sum_{i=1}^{k-1} \tilde{x}^{(i)} \leq \sum_{i=1}^{k} x^{*(i)}$$

where $x^{*(i)}$ is the corresponding optimal solution to each one of the accordingly subproblems. Then, $\tilde{x} = (\tilde{x}^{(1)}, \tilde{x}^{(2)}, \ldots, \tilde{x}^{(k)})$, is an optimal solution to problem (1). Lets suppose now that $\hat{x} = (\hat{x}^{(1)}, \hat{x}^{(2)} \ldots, \hat{x}^{(k+1)})$ is a vector of optimal solutions to the subproblems (P_n), n = 1, 2, ..., k + 1, satisfying the corresponding regularity condition,

$$\sum_{i=1}^{k} \hat{x}^{(i)} \leq \sum_{i=1}^{k+1} x^{*(i)}$$

where $x^{*(i)}$, i = 1, 2, ..., k + 1, are the corresponding partition of vector $x^{*(i)}$ into k + 1 blocks, in such a way that the first k blocks coincide with the partition of $\tilde{x}^{(i)}$, i = 1, 2, ..., k. We will prove that one such a vector $\hat{x}$ is an

optimal solution to problem (1). First, lets suppose that the vector $\tilde{x}$, which by induction hypothesis is an optimal solution
to problem (1), is partitioned into k + 1 blocks, and let also suppose that the vector $\hat{x}$ have the same partition as $\tilde{x}$, and that the regularity condition is satisfied, that is,

$$\sum_{i=1}^{k} \hat{x}^{(i)} \le \sum_{i=1}^{k+1} \tilde{x}^{(i)}$$

Then, the resulting (k+1) blocks from the partitioning of $\tilde{x}$ clearly gives feasible solutions to subproblems (P_n), n=1,2, ..., k+1. However, since $\hat{x}^{(n)}$, n = 1, 2, ..., k+1, are optimal solutions to subproblems (P_n), n = 1, 2, ..., k + 1, then,

$$\sum_{i=1}^{k+1} c^{(i)} \hat{x}^{(i)} \ge \sum_{i=1}^{k+1} c^{(i)} \tilde{x}^{(i)}$$

Thus, given that $\tilde{x}$ is an optimal solution to problem (1), then so is $\hat{x}$, and the proof is complete. It is important to observe that this proposition assure that any optimal solution to MPS linear programming problem can be obtained by the rule of proposed decomposition, and that this fact allow to solve very large scale linear programming problems by means of small ones. Is is likely that the result of this proposition remains valid if the decision variables are constrained to be integer, and in particular boolean variables, since no explicit use of the continuity of the variables were made.

4 Numerical Experiments and Conclusions

4.1 Numerical Experiments

In order to show how the approach of decomposition perform numerically, lets consider a Master Production Scheduling for a Company which manufacture 73 kind of different products, using 167 different kind of components, for a planing horizon of 12 periods. For this problem we have 876 decision variables and 2892 functional constraints. Furthermore, lets suppose we know the incomes from each product manufactured in each one of the periods in the planing horizon, the corresponding demands, the standard time for manufacture each one of the products, and the available handwork resources for each period as well. The tables below show the optimal solution for the problem without decomposition (1 Block), and then using decomposition into two, three, four, six, and twelve blocks. And we also provide the tables related to the performing time required to solve the problem corresponding to each decomposition, one for the theoretical worst case and the other for the actually real time consuming.

Table 1 – 1 Block (without decomposition)

Product	Period: 1 Block												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	793.705	1,634.646	2,837.864	4,037.737	4,601.517	4,803.305	4,747.155	5,145.421	5,742.953	6,975.611	10,200.836	18,304.250	69,825.000
2	25,159.446	33,715.885	34,897.600	33,584.551	33,248.765	33,302.492	32,772.799	32,961.951	32,586.277	31,716.385	26,134.902	12,008.945	362,090.000
3	24,579.655	33,150.779	35,436.496	33,451.730	33,046.663	32,952.746	32,663.351	32,731.801	32,478.885	31,961.189	27,657.979	12,277.726	362,389.000
4	4,396.888	8,806.834	9,681.973	9,815.025	9,736.608	9,792.275	9,581.918	9,844.998	9,936.359	9,876.574	9,947.321	8,003.228	109,420.000
5	6,028.906	11,549.506	12,582.439	12,395.558	12,184.335	12,224.864	12,242.595	12,704.996	13,340.785	13,813.840	13,672.296	8,234.880	140,975.000
6	38,711.547	54,548.382	46,856.886	47,816.636	47,911.978	48,208.134	47,409.947	47,885.915	47,917.709	47,833.128	38,257.949	14,222.791	527,581.000
7	47,681.448	65,316.780	55,522.335	57,443.696	57,570.726	57,838.790	57,267.208	57,629.025	57,515.319	57,066.851	46,189.015	12,118.808	629,160.000
8	7,567.892	18,693.842	17,614.607	17,185.846	16,956.025	17,004.361	16,355.255	16,534.340	16,159.319	15,341.671	14,396.466	11,988.377	185,798.000
9	20,758.550	48,238.280	35,658.830	37,124.903	36,901.255	37,072.446	36,347.541	36,672.319	36,481.254	35,862.120	31,196.710	14,023.793	406,338.000
10	1,356.129	2,960.191	4,573.508	5,561.448	6,036.278	6,201.920	5,927.205	6,054.280	6,188.600	6,654.545	8,679.408	13,360.488	73,554.000
11	1,315.931	2,904.742	4,545.576	5,574.734	6,076.397	6,252.603	5,972.465	6,095.276	6,222.903	6,683.701	8,684.658	13,225.015	73,554.000
12	970.214	1,741.624	2,532.459	3,139.232	3,526.553	3,757.572	3,787.702	3,975.439	4,230.685	4,798.746	6,535.412	10,267.361	49,263.000
13	12,808.330	35,735.705	39,551.318	37,126.695	35,778.755	35,686.034	33,811.895	34,473.087	33,618.904	32,357.929	33,450.297	18,980.051	383,379.000
14	6,656.672	12,487.676	12,003.591	11,707.611	11,517.108	11,402.270	10,723.947	10,295.130	9,319.581	8,318.538	7,406.248	6,782.628	118,621.000
15	11,289.312	22,467.938	18,990.575	19,022.504	18,756.155	18,603.579	17,776.011	17,154.975	15,001.733	12,239.698	9,481.877	6,930.642	187,715.000
16	411.208	616.895	816.670	1,029.442	1,254.992	1,479.641	1,670.567	1,915.764	2,219.833	2,715.056	3,969.959	8,097.973	26,198.000
17	3,311.105	7,720.239	9,542.988	9,552.410	9,480.245	9,421.222	8,870.074	8,891.190	8,700.947	8,651.626	9,700.698	11,413.255	105,256.000
18	195.254	279.293	348.889	418.135	498.104	599.702	724.430	898.269	1,139.477	1,533.767	2,477.293	6,113.387	15,226.000
19	347.880	550.141	741.171	938.970	1,150.635	1,367.136	1,555.048	1,787.335	2,073.809	2,544.435	3,746.405	7,471.035	24,274.000
20	330.069	517.966	696.411	885.420	1,093.209	1,311.803	1,509.247	1,753.033	2,054.961	2,547.389	3,807.965	7,766.527	24,274.000
21	323.170	504.624	677.021	861.728	1,067.511	1,286.689	1,488.034	1,736.663	2,045.458	2,548.073	3,834.856	7,900.172	24,274.000
22	599.084	1,201.417	2,049.380	2,981.117	3,726.699	4,194.655	4,285.987	4,518.044	4,815.693	5,506.320	7,717.002	13,244.602	54,840.000
23	182.302	264.812	339.518	416.055	502.729	610.052	746.864	956.571	1,283.225	1,880.142	3,476.059	11,641.671	22,300.000
24	4,163.763	7,094.768	7,674.802	7,705.613	7,697.257	7,778.488	7,158.516	7,155.693	6,880.233	6,417.788	6,161.022	6,705.058	82,593.000
25	5,227.421	9,216.677	9,686.352	9,623.376	9,529.010	9,497.955	9,029.316	9,028.113	8,632.792	7,947.470	7,319.093	7,225.424	101,963.000
26	791.105	1,142.035	1,381.417	1,551.766	1,684.152	1,780.703	1,793.461	1,923.031	2,079.457	2,301.659	2,767.145	4,828.070	24,024.000
27	2,292.779	4,277.699	4,902.090	5,041.149	5,062.311	5,036.808	4,776.315	4,669.116	4,484.740	4,468.276	5,223.212	6,713.506	56,948.000
28	937.004	1,427.475	1,741.119	1,925.692	2,039.532	2,097.802	2,051.054	2,148.590	2,270.190	2,460.507	2,912.867	4,941.168	26,953.000
29	715.460	1,030.084	1,238.271	1,381.964	1,491.434	1,571.805	1,587.533	1,699.288	1,837.559	2,034.647	2,454.128	4,312.826	21,355.000
30	1,118.569	1,717.362	2,072.540	2,274.268	2,397.315	2,458.139	2,416.591	2,519.292	2,642.754	2,828.092	3,229.134	4,979.944	30,654.000
31	3,106.527	7,559.371	8,634.579	8,363.458	8,200.419	8,070.805	7,531.390	7,364.080	6,932.494	6,501.541	6,881.562	7,451.772	86,598.000
32	915.869	1,503.231	1,891.109	2,136.325	2,297.312	2,406.794	2,430.194	2,557.396	2,715.919	3,003.860	3,868.241	6,282.750	32,009.000
33	2,305.819	5,154.323	6,439.729	6,583.883	6,509.373	6,390.174	5,961.626	5,891.452	5,737.593	5,725.988	6,668.538	8,205.503	71,574.000
34	1,313.105	2,466.157	3,214.687	3,581.697	3,738.473	3,780.525	3,639.113	3,690.009	3,749.618	3,946.011	4,795.482	6,815.123	44,730.000
35	1,974.473	3,970.092	4,850.407	5,089.824	5,123.014	5,082.163	4,805.051	4,716.257	4,540.672	4,529.801	5,324.850	6,941.397	56,948.000
36	2,127.862	4,159.558	4,915.719	5,089.029	5,103.163	5,065.692	4,801.566	4,704.547	4,517.766	4,490.445	5,239.738	6,732.915	56,948.000
37	1,408.715	3,110.428	4,319.727	4,820.224	5,025.091	5,115.957	4,941.577	4,969.319	4,916.591	5,023.656	5,842.802	7,453.912	56,948.000
38	1,338.477	2,947.412	4,169.133	4,746.036	4,998.946	5,112.995	4,947.289	4,983.222	4,941.621	5,071.858	5,962.377	7,728.635	56,948.000
39	18,980.827	36,807.021	31,669.656	31,018.963	27,115.006	21,971.233	41,422.391	32,145.997	30,667.671	29,414.341	25,160.373	12,805.520	339,179.000
40	16,066.822	36,100.295	30,585.490	30,423.332	30,048.565	30,789.047	28,341.021	29,385.716	29,258.943	28,069.934	24,218.374	13,245.461	326,533.000
41	2,360.163	4,430.973	5,885.000	6,422.956	6,583.162	6,426.437	5,652.263	5,700.651	5,858.899	6,280.933	7,256.646	9,709.917	72,568.000
42	1,255.216	2,316.481	3,336.278	3,649.567	3,586.658	3,576.622	3,511.396	3,658.743	3,898.956	4,460.169	5,898.933	10,124.981	49,274.000
43	615.055	1,046.443	1,507.592	1,965.105	2,408.199	2,824.002	3,100.610	3,543.104	4,102.414	5,110.842	8,003.716	15,046.917	49,274.000
44	6,307.589	12,528.348	13,353.164	13,055.263	12,710.184	12,558.726	11,926.736	12,018.429	11,474.105	10,889.048	10,745.065	10,233.344	137,800.000
45	6,601.385	14,288.525	15,329.114	14,985.560	14,769.796	14,424.625	13,304.957	13,504.856	13,419.382	13,371.384	12,897.926	11,351.491	158,249.000
46	1,874.899	3,179.873	4,051.340	4,417.764	4,544.865	4,460.106	4,023.595	4,084.060	4,247.275	4,632.918	5,557.640	8,254.664	53,329.000
47	398.857	646.818	883.663	1,110.471	1,328.069	1,541.745	1,732.692	2,009.400	2,394.342	3,079.284	4,902.041	10,461.617	30,489.000
48	672.392	1,253.362	1,888.122	2,478.184	2,969.487	3,355.812	3,554.146	3,919.372	4,411.916	5,350.784	8,069.758	13,929.664	51,853.000
49	340.156	518.950	684.088	845.081	1,006.845	1,175.053	1,340.086	1,577.584	1,913.196	2,501.833	4,003.029	8,863.101	24,769.000
50	1,039.457	2,240.671	3,577.527	4,415.228	4,727.194	4,809.984	4,616.873	4,717.009	4,919.545	5,514.664	7,109.382	10,957.464	58,645.000
51	214.766	276.828	327.852	375.955	425.017	476.064	527.433	600.632	705.440	893.872	1,368.995	3,420.144	9,613.000
52	372.134	504.109	605.862	695.176	780.713	862.405	931.412	1,039.235	1,191.078	1,467.809	2,177.570	4,701.497	15,329.000
53	303.231	422.809	516.861	602.178	687.502	774.657	857.344	980.173	1,153.905	1,466.048	2,265.470	5,298.823	15,329.000
54	283.635	394.584	483.021	565.017	648.883	736.614	822.777	949.511	1,129.812	1,454.010	2,286.761	5,574.376	15,329.000
55	630.292	1,134.759	1,625.919	2,138.348	2,443.252	2,575.991	2,540.125	2,590.266	2,650.244	2,833.867	3,488.207	5,129.730	29,781.000
56	551.224	975.883	1,414.787	1,876.043	2,172.750	2,330.204	2,366.739	2,443.053	2,533.571	2,755.433	3,472.655	5,329.659	28,222.000
57	2,036.894	5,570.015	6,295.871	1,307.029	1,205.615	1,221.069	1,234.961	1,274.709	1,356.202	1,530.411	1,912.315	3,276.909	28,222.000
58	1,993.347	5,348.367	6,351.636	1,389.496	1,228.097	1,230.509	1,244.886	1,284.496	1,366.339	1,542.896	1,932.152	3,309.780	28,222.000
59	8,891.891	14,585.293	14,182.320	13,969.365	13,822.720	13,741.573	13,130.170	13,204.202	13,773.610	11,874.587	9,158.993	7,627.276	147,962.000
60	9,004.355	14,926.167	14,471.370	14,175.597	13,932.551	14,003.760	13,459.625	13,986.124	13,963.727	12,077.590	9,446.745	7,704.389	151,152.000
61	1,376.070	1,916.263	2,210.238	2,341.470	2,382.404	2,360.908	2,203.517	2,272.584	2,411.983	2,575.523	2,794.880	4,444.160	29,290.000
62	2,042.448	3,377.371	3,844.885	3,959.025	3,937.414	3,842.663	3,524.168	3,400.591	3,321.027	3,316.861	3,676.957	4,926.591	43,170.000
63	1,213.861	1,689.423	1,963.080	2,098.605	2,158.032	2,164.627	2,053.407	2,141.900	2,296.399	2,486.001	2,765.028	4,620.639	27,651.000
64	1,985.620	3,317.717	3,807.245	3,932.527	3,912.493	3,816.179	3,497.835	3,381.650	3,322.456	3,345.632	3,756.685	5,093.962	43,170.000
65	746.355	1,270.688	1,755.467	2,243.993	2,705.169	3,114.936	3,317.672	3,669.490	4,273.366	5,168.059	6,803.751	10,377.053	45,446.000
66	472.050	846.741	1,304.755	1,839.942	2,378.965	2,844.710	3,124.464	3,487.819	3,921.671	4,714.718	6,970.484	13,539.683	45,446.000
67	2,582.731	4,071.594	4,642.635	4,763.918	4,750.299	4,684.008	4,261.171	4,296.612	4,343.964	4,317.599	4,258.795	5,742.674	52,716.000
68	3,587.737	5,861.361	6,485.299	6,522.029	6,452.294	6,375.371	5,821.275	5,773.721	5,639.733	5,368.383	4,901.525	5,885.271	68,674.000
69	3,266.583	5,613.576	6,473.382	6,569.299	6,494.421	6,385.874	5,730.386	5,716.324	5,630.671	5,419.042	5,071.539	6,302.902	68,674.000
70	1,912.549	3,299.003	3,837.069	3,988.290	3,999.248	3,952.271	3,687.369	3,631.842	3,625.019	3,682.037	4,161.852	5,669.450	45,446.000
71	2,012.308	3,397.302	3,883.021	4,010.973	4,015.723	3,968.774	3,697.574	3,624.793	3,595.621	3,630.279	4,088.097	5,521.535	45,446.000
72	1,826.538	3,208.285	3,787.536	3,960.405	3,976.526	3,929.567	3,667.920	3,623.815	3,641.405	3,726.761	4,249.613	5,847.629	45,446.000
73	1,796.025	3,171.224	3,760.807	3,943.596	3,965.232	3,920.110	3,655.396	3,608.431	3,630.709	3,729.072	4,296.418	5,968.980	45,446.000

Table 2 – 2 Blocks

Product	Period: 2 Blocks												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	4,472.312	4,986.761	9,937.775	7,065.641	6,682.604	6,683.004	620.433	1,072.724	1,690.076	2,926.738	6,436.818	17,250.115	69,825.000
2	34,832.867	34,935.106	38,443.669	32,748.270	35,595.942	35,595.961	34,537.758	29,975.255	32,131.398	32,211.609	14,271.992	6,810.174	362,090.000
3	33,511.562	32,267.086	38,896.920	31,678.789	34,432.807	32,754.129	34,146.044	37,747.161	32,640.234	29,193.377	14,048.430	11,072.460	362,389.000
4	8,899.175	8,422.173	9,506.347	9,745.279	12,369.143	17,786.548	3,722.480	7,762.640	8,269.054	9,160.931	7,766.569	6,009.662	109,420.000
5	11,467.498	10,947.272	12,326.029	11,946.210	12,786.570	14,167.060	7,482.608	15,524.769	14,067.272	13,581.356	8,410.231	8,268.125	140,975.000
6	47,740.929	42,206.295	15,471.139	0.037	0.018	0.012	189,986.821	87,747.640	54,771.219	44,132.342	22,920.934	22,603.614	527,581.000
7	59,969.935	57,716.612	66,872.200	55,853.815	60,710.483	60,710.656	36,839.643	67,157.368	62,201.832	58,391.049	31,864.983	10,871.425	629,160.000
8	16,897.147	16,463.878	20,694.512	16,358.943	17,781.451	17,781.506	4,764.176	13,431.772	15,121.313	16,549.970	17,421.355	12,531.976	185,798.000
9	38,285.386	35,091.324	6,331.666	0.045	0.022	0.014	130,692.290	73,043.521	44,300.163	37,031.849	29,024.289	12,537.432	406,338.000
10	5,212.398	5,337.868	10,848.838	6,476.655	7,039.347	7,039.406	912.289	2,169.793	3,222.757	4,513.086	7,483.497	13,298.067	73,554.000
11	5,253.717	5,369.796	10,772.965	6,479.269	7,039.352	7,039.413	922.719	2,194.041	3,259.394	4,542.818	7,490.109	13,190.407	73,554.000
12	3,520.590	3,983.249	6,828.524	4,337.472	4,714.626	4,714.692	775.083	1,378.727	2,040.144	2,990.606	4,931.524	9,047.762	49,263.000
13	34,529.145	33,506.816	43,503.268	33,755.350	36,690.573	36,690.635	8,059.707	23,337.198	29,814.751	39,459.171	44,001.023	20,031.363	383,379.000
14	12,084.356	11,909.392	14,455.374	11,616.466	7,972.362	0.339	5,289.500	11,156.649	11,622.784	12,314.092	11,843.089	8,356.598	118,621.000
15	19,578.648	19,242.177	22,698.144	18,618.280	20,236.787	20,237.198	6,064.132	12,197.622	13,197.397	13,885.880	11,834.481	9,924.254	187,715.000
16	1,460.342	1,743.693	4,417.867	2,306.669	2,507.236	2,507.287	365.985	566.070	813.055	1,254.629	2,286.643	5,968.524	26,198.000
17	9,112.800	8,636.518	12,873.528	9,267.478	10,073.329	10,073.384	1,968.578	5,210.138	6,814.699	8,217.726	10,932.270	12,075.554	105,256.000
18	0.000	0.000	0.000	0.000	0.000	0.000	541.122	888.945	1,228.943	1,746.798	2,979.750	7,840.442	15,226.000
19	1,766.004	1,622.905	3,438.700	2,371.525	2,323.155	2,323.372	347.269	529.352	740.904	1,114.645	2,036.773	5,659.393	24,274.000
20	0.000	0.000	0.000	0.000	0.000	0.000	811.578	1,873.869	2,415.437	3,122.785	5,076.265	10,974.066	24,274.000
21	1,658.146	1,611.040	3,420.834	2,509.098	2,323.150	1,273.021	378.662	573.793	805.876	1,215.861	2,225.147	6,279.373	24,274.000
22	3,773.956	3,033.548	5,547.784	7,701.813	5,971.718	4,595.970	484.379	987.139	1,659.397	2,671.544	5,049.539	13,363.214	54,840.000
23	584.948	970.370	4,290.618	2,603.118	2,136.306	1,885.399	222.805	348.831	499.963	764.993	1,492.926	6,499.723	22,300.000
24	7,500.711	7,053.030	8,221.362	7,473.555	8,674.724	9,218.814	2,611.481	4,747.764	6,200.022	6,796.777	6,236.662	7,858.098	82,593.000
25	9,334.095	8,812.036	10,100.378	8,985.211	10,737.405	11,500.646	3,417.594	6,397.359	8,051.859	8,611.727	7,532.550	8,482.139	101,963.000
26	1,666.283	1,596.941	1,946.636	2,188.670	2,951.123	3,809.817	600.287	870.028	1,157.545	1,407.059	1,792.157	4,037.454	24,024.000
27	4,835.489	4,583.208	5,308.879	5,379.081	6,038.590	6,421.533	1,406.248	3,052.429	3,615.918	4,098.107	5,113.904	7,094.615	56,948.000
28	2,011.532	1,942.009	2,345.594	2,475.791	3,123.408	3,677.082	692.255	1,035.090	1,388.619	1,688.616	2,089.678	4,483.325	26,953.000
29	1,520.959	1,479.564	1,830.840	1,928.572	2,573.021	3,410.566	537.826	766.591	1,002.601	1,197.380	1,558.784	3,548.295	21,355.000
30	2,425.196	2,341.528	2,752.585	2,787.314	3,618.609	4,482.078	796.810	1,194.668	1,591.704	1,911.561	2,261.180	4,490.767	30,654.000
31	8,075.585	7,339.539	8,611.702	8,296.394	9,133.021	9,730.474	1,691.407	4,472.690	5,684.940	6,502.595	8,060.308	8,999.345	86,598.000
32	2,169.189	2,016.637	2,494.048	2,883.901	3,435.668	3,053.541	765.140	1,354.875	1,810.846	2,329.749	3,344.235	6,351.171	32,009.000
33	6,165.961	5,789.016	6,694.617	6,771.434	7,494.048	8,034.297	1,300.328	3,218.235	4,211.836	5,170.176	7,077.888	9,646.166	71,574.000
34	3,622.442	3,471.624	4,150.906	4,546.612	5,049.265	4,961.111	875.214	1,696.816	2,296.538	2,936.183	4,121.379	7,001.909	44,730.000
35	4,803.948	4,544.584	5,197.960	5,279.486	6,061.149	6,679.653	1,340.977	2,940.170	3,568.795	4,091.823	5,142.262	7,297.194	56,948.000
36	4,824.574	4,543.928	5,197.690	5,268.447	6,046.599	6,685.542	1,370.034	3,010.853	3,632.687	4,135.554	5,132.905	7,099.187	56,948.000
37	4,644.808	4,533.785	4,961.155	4,935.043	5,482.542	3,181.661	1,824.149	3,831.271	4,686.309	5,478.538	6,279.280	7,109.460	56,948.000
38	4,510.961	4,411.621	4,821.065	4,946.640	6,172.760	7,619.558	1,143.450	2,504.562	3,575.479	4,638.924	5,637.498	6,965.481	56,948.000
39	31,575.712	30,461.484	34,936.070	30,668.637	33,178.622	32,663.031	10,986.071	34,694.842	28,102.808	31,846.858	27,234.944	12,829.921	339,179.000
40	30,021.100	29,542.661	33,426.614	29,528.781	32,440.477	31,909.102	9,287.864	27,636.412	29,301.264	32,372.042	27,808.222	13,258.462	326,533.000
41	6,180.172	5,782.169	6,786.165	7,502.200	8,360.543	7,558.244	1,538.800	2,658.074	3,825.530	5,168.169	6,579.453	10,628.481	72,568.000
42	4,163.110	4,291.789	5,883.973	4,339.866	4,716.933	4,717.325	845.269	1,588.446	2,093.471	2,788.474	4,481.445	9,363.900	49,274.000
43	3,001.656	3,045.402	6,856.859	5,769.615	4,716.247	4,716.219	560.331	954.699	1,435.794	2,298.429	4,392.867	11,525.881	49,274.000
44	12,568.966	11,978.834	14,060.355	12,868.021	14,147.124	13,754.569	3,212.376	7,198.914	10,202.255	12,149.707	13,047.116	12,611.762	137,800.000
45	14,419.014	13,753.327	16,121.012	14,726.083	16,171.503	15,762.869	3,733.086	8,307.133	11,858.087	14,480.062	15,596.730	13,320.094	158,249.000
46	4,423.936	4,214.013	5,124.916	5,561.411	5,966.776	5,302.502	1,207.993	1,940.428	2,704.017	3,630.776	4,751.760	8,500.472	53,329.000
47	1,901.798	2,312.338	4,348.387	3,510.024	3,063.997	3,054.218	341.874	608.150	884.451	1,317.963	2,368.644	6,777.157	30,489.000
48	2,992.779	2,636.574	4,549.750	6,465.228	7,015.935	6,106.434	546.853	1,175.467	1,761.175	2,577.300	4,576.746	11,448.758	51,853.000
49	805.043	928.849	2,103.088	3,328.186	4,004.058	3,131.755	325.887	545.123	780.173	1,150.984	2,023.934	5,641.921	24,769.000
50	4,243.804	4,215.856	5,744.328	6,532.795	6,727.929	4,822.517	726.372	1,691.294	2,526.673	3,575.487	6,040.252	11,797.695	58,645.000
51	171.795	164.729	346.188	439.827	229.922	18.028	465.461	615.234	758.342	982.617	1,489.367	3,931.489	9,613.000
52	775.056	823.462	1,393.542	2,013.713	2,323.216	1,766.392	314.058	439.876	559.993	736.462	1,121.044	3,062.187	15,329.000
53	716.351	842.223	1,394.779	1,816.902	2,330.901	1,994.224	289.258	412.026	531.662	711.020	1,109.708	3,179.946	15,329.000
54	582.897	733.752	1,309.289	1,737.140	2,373.106	2,359.197	272.488	394.456	514.309	695.184	1,100.888	3,256.294	15,329.000
55	3,006.261	2,584.173	3,629.451	4,029.285	4,076.581	3,685.444	468.043	782.735	1,093.602	1,397.191	1,895.362	3,132.873	29,781.000
56	2,432.849	2,549.975	3,325.346	3,774.387	3,996.850	4,042.548	413.553	675.676	956.553	1,257.263	1,754.910	3,042.090	28,222.000
57	0.000	0.000	0.000	0.000	0.000	0.000	1,271.216	2,918.997	3,794.945	4,684.755	6,809.591	8,742.496	28,222.000
58	0.000	0.000	0.000	0.000	0.000	0.000	1,254.873	2,853.903	3,746.530	4,657.671	6,837.926	8,871.097	28,222.000
59	13,709.844	13,312.938	15,011.997	13,014.214	15,115.103	15,829.585	5,796.008	10,826.439	12,788.215	14,050.231	9,193.247	9,314.179	147,962.000
60	14,024.805	13,608.904	15,272.548	13,163.110	15,554.431	16,417.351	5,925.218	11,013.317	13,154.080	14,332.853	9,309.185	9,376.199	151,152.000
61	2,465.317	2,333.315	2,529.346	2,625.501	3,743.649	4,730.089	769.386	1,068.546	1,404.427	1,663.220	1,852.888	4,104.315	29,290.000
62	3,880.247	3,683.845	4,104.925	4,199.229	5,062.111	5,686.736	1,113.309	1,964.073	2,439.513	2,812.213	3,126.550	5,097.250	43,170.000
63	2,175.982	2,077.840	2,295.697	2,374.670	3,319.731	4,135.830	750.280	1,059.756	1,419.905	1,716.671	1,932.663	4,391.977	27,651.000
64	3,869.004	3,678.166	4,088.105	4,153.835	5,080.002	5,747.980	1,065.924	1,928.125	2,392.533	2,778.790	3,153.178	5,234.358	43,170.000
65	2,657.786	2,644.662	4,997.371	6,357.301	4,905.625	4,359.227	635.312	1,146.018	1,815.544	2,888.433	4,391.501	8,647.220	45,446.000
66	2,685.901	2,401.131	5,504.510	6,628.971	4,347.293	4,354.167	388.641	698.035	1,132.438	1,921.735	3,843.547	11,539.631	45,446.000
67	4,604.811	4,346.763	4,832.066	4,451.210	5,791.110	6,686.070	1,512.124	2,426.655	3,429.510	4,148.040	3,800.217	6,687.425	52,716.000
68	6,305.457	5,949.646	6,669.303	5,977.062	7,568.716	8,479.188	2,078.291	3,431.649	4,797.319	5,596.789	4,644.902	7,175.679	68,674.000
69	6,277.205	5,803.395	5,781.776	3,231.578	2,162.601	1,217.305	4,230.417	6,561.826	8,461.831	8,866.333	6,778.273	9,301.461	68,674.000
70	3,862.176	3,662.530	4,049.066	4,135.789	5,069.088	5,838.442	1,147.142	2,120.357	2,705.222	3,213.379	3,642.825	5,999.985	45,446.000
71	3,887.170	3,679.913	4,071.601	4,139.578	5,051.746	5,787.085	1,160.651	2,174.470	2,738.881	3,217.737	3,635.593	5,901.577	45,446.000
72	3,839.556	3,649.737	4,018.801	4,074.427	5,067.933	5,723.606	1,124.663	2,106.206	2,692.399	3,225.013	3,709.445	6,214.214	45,446.000
73	3,809.128	3,612.549	3,961.767	4,054.507	5,101.514	6,077.627	1,032.816	1,998.901	2,582.674	3,139.863	3,703.975	6,370.680	45,446.000

Table 3 – 3 Blocks

Product	Period: 3 Blocks												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	5,406.917	5,627.500	5,804.764	8,618.215	2,156.116	5,884.362	9,109.092	8,307.826	753.946	1,532.401	3,108.646	13,515.215	69,825.000
2	32,693.650	32,828.543	35,953.629	34,127.766	29,107.676	38,125.453	34,401.496	33,968.963	30,633.704	27,209.184	24,837.580	8,202.354	362,090.000
3	32,525.144	32,596.704	32,697.655	32,280.398	32,718.381	33,164.619	33,610.826	30,606.074	31,748.521	33,227.190	27,008.308	10,205.181	362,389.000
4	9,100.338	9,056.986	9,372.847	15,121.580	4,024.236	8,695.555	14,054.245	15,877.714	3,769.833	7,012.336	7,002.485	6,331.844	109,420.000
5	11,520.107	11,530.390	11,487.394	12,531.868	10,495.642	11,648.491	12,310.998	12,614.629	9,669.743	16,518.516	13,075.198	7,572.022	140,975.000
6	47,618.962	47,827.364	36,668.925	1,002.435	103,535.408	31,034.790	802.790	70.680	135,002.158	64,995.489	37,099.728	21,922.272	527,581.000
7	57,212.722	57,520.484	57,922.132	58,622.195	55,235.208	58,723.255	59,170.498	58,148.571	46,570.136	59,533.698	42,928.342	17,572.757	629,160.000
8	16,436.509	16,665.154	17,011.102	17,626.090	14,781.596	17,868.732	17,951.529	17,136.996	6,238.993	15,237.062	14,055.740	14,788.496	185,798.000
9	36,756.659	36,616.625	29,721.490	1,119.244	78,548.205	22,704.060	533.619	102.151	102,301.280	47,610.578	30,858.473	19,465.614	406,338.000
10	5,862.443	5,804.468	6,674.652	8,475.000	2,793.364	7,120.915	9,752.179	7,150.105	1,194.951	2,997.056	4,814.219	10,914.649	73,554.000
11	5,913.847	5,868.087	6,731.064	8,303.564	2,940.875	7,287.497	9,580.685	7,007.506	1,219.279	3,014.850	4,818.502	10,868.243	73,554.000
12	3,932.026	4,006.469	4,592.245	5,429.729	2,336.643	5,716.764	5,163.502	4,743.559	924.531	1,806.174	3,136.006	7,475.351	49,263.000
13	33,863.358	34,378.947	35,439.641	36,091.647	29,166.803	39,847.787	35,560.792	35,198.211	12,763.951	31,814.299	31,232.464	28,021.098	383,379.000
14	11,547.650	11,625.736	11,572.866	2,350.768	18,966.721	12,947.264	4,481.638	701.396	8,154.623	14,259.686	11,753.773	10,258.878	118,621.000
15	18,826.749	18,770.462	19,279.369	20,215.931	16,370.860	20,058.885	20,548.809	20,113.957	5,634.676	9,609.385	10,027.837	8,258.080	187,715.000
16	1,718.644	1,831.105	2,423.428	3,578.177	846.310	2,081.642	3,872.077	2,751.325	440.052	787.346	1,451.548	4,416.346	26,198.000
17	9,071.757	8,953.712	9,638.857	10,710.257	6,572.127	10,516.119	11,402.154	9,884.183	2,426.706	6,249.884	7,780.062	12,050.181	105,256.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1,171.016	2,329.827	3,382.571	8,342.587	15,226.000
19	1,878.394	1,953.679	2,196.813	2,821.009	1,126.145	2,060.004	2,823.779	2,839.967	440.363	765.817	1,345.336	4,022.692	24,274.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2,163.224	4,508.759	5,483.896	12,118.121	24,274.000
21	1,840.459	1,916.819	2,147.555	2,273.684	1,417.781	2,236.238	2,832.925	1,691.572	544.636	935.221	1,624.289	4,812.821	24,274.000
22	4,216.109	4,293.136	4,593.383	6,472.181	1,638.375	4,249.958	7,542.681	6,143.795	655.477	1,454.255	2,924.951	10,655.700	54,840.000
23	935.618	940.507	1,619.409	4,475.549	387.997	933.923	2,202.184	4,446.979	304.265	534.798	1,035.545	4,483.228	22,300.000
24	7,385.469	7,337.887	7,533.297	8,514.965	5,821.287	7,824.004	8,439.810	8,686.516	3,247.030	5,844.377	5,154.384	6,803.973	82,593.000
25	9,147.347	9,136.076	9,288.049	10,440.525	7,456.516	9,604.007	10,254.130	10,697.345	4,403.575	7,604.986	6,156.996	7,773.447	101,963.000
26	1,908.427	1,894.392	2,121.001	3,126.656	1,025.139	1,830.559	2,733.179	3,461.598	722.125	1,079.801	1,369.878	2,751.244	24,024.000
27	4,916.946	4,811.165	4,977.962	6,110.022	3,166.521	5,159.632	6,032.356	6,457.586	1,791.087	3,433.614	3,738.775	6,352.336	56,948.000
28	2,198.938	2,175.405	2,333.858	3,247.322	1,241.354	2,165.446	3,017.073	3,531.650	858.841	1,326.382	1,657.721	3,199.010	26,953.000
29	1,723.655	1,706.945	1,882.020	2,832.808	919.467	1,616.509	2,402.753	3,206.699	635.292	924.911	1,153.468	2,350.473	21,355.000
30	2,624.328	2,592.864	2,755.838	3,792.588	1,539.835	2,595.055	3,485.014	4,145.714	939.901	1,404.594	1,706.214	3,072.054	30,654.000
31	7,902.220	7,649.099	7,972.438	9,193.862	5,764.338	8,303.303	9,146.601	9,503.377	2,082.702	5,027.253	5,489.668	8,563.140	86,598.000
32	2,520.076	2,512.580	2,590.988	2,637.132	1,767.088	2,809.542	3,435.902	2,248.245	1,090.076	2,051.428	2,760.600	5,585.344	32,009.000
33	6,258.926	5,979.993	6,284.802	7,650.374	4,021.000	6,576.936	7,626.255	7,949.904	1,720.085	4,128.621	4,782.731	8,594.374	71,574.000
34	3,839.480	3,826.342	3,949.640	4,876.684	2,208.290	4,048.224	5,344.178	4,891.454	1,113.982	2,147.084	2,882.068	5,602.577	44,730.000
35	4,944.966	4,803.664	4,963.079	6,104.386	3,214.761	5,132.273	5,990.119	6,478.941	1,734.888	3,434.783	3,796.926	6,349.214	56,948.000
36	4,929.517	4,805.195	4,974.555	6,106.828	3,202.293	5,141.115	5,983.233	6,489.453	1,770.628	3,514.835	3,823.549	6,206.800	56,948.000
37	4,927.448	4,765.596	4,730.347	3,306.870	5,451.543	5,551.570	5,016.959	1,710.189	2,629.807	5,391.385	5,957.653	7,508.633	56,948.000
38	4,828.069	4,748.843	4,813.723	6,371.657	2,701.617	5,037.936	6,223.421	6,799.317	1,317.938	3,123.851	4,516.212	6,465.415	56,948.000
39	30,620.460	30,527.099	30,810.694	31,712.924	28,565.921	31,462.171	31,819.465	31,823.619	15,819.979	33,781.252	21,801.371	20,434.045	339,179.000
40	29,457.598	29,462.189	29,602.233	30,921.083	26,859.607	30,397.008	30,956.078	31,230.409	12,241.497	31,662.365	22,322.263	21,420.670	326,533.000
41	6,403.378	6,163.169	6,569.158	7,818.269	3,594.510	6,974.143	8,301.202	8,084.119	1,727.951	3,154.802	4,347.928	9,429.370	72,568.000
42	4,211.856	4,102.831	4,304.801	5,349.829	2,347.888	4,628.925	5,668.155	5,324.348	1,076.990	1,950.669	3,015.932	7,291.777	49,274.000
43	3,516.816	3,564.516	4,238.525	6,644.987	1,240.928	3,304.156	6,780.828	6,638.932	682.480	1,292.173	2,451.992	8,917.667	49,274.000
44	12,245.009	12,359.198	12,472.051	13,660.635	9,610.625	13,353.385	13,772.948	13,999.935	3,523.367	9,524.640	9,568.158	13,710.049	137,800.000
45	14,062.005	14,191.129	14,312.779	15,570.109	11,639.687	14,990.224	15,637.913	15,868.198	4,907.679	10,677.227	11,057.253	15,334.795	158,249.000
46	4,603.370	4,494.316	4,712.118	5,745.040	2,625.579	4,915.528	6,007.181	6,006.556	1,358.921	2,345.640	3,302.700	7,212.051	53,329.000
47	1,990.129	2,030.831	2,516.189	5,090.055	598.182	1,457.028	3,329.905	6,242.090	438.231	804.695	1,431.202	4,560.463	30,489.000
48	3,843.488	3,797.539	4,282.644	7,102.664	1,152.089	3,191.290	6,470.777	8,212.179	735.786	1,560.538	2,746.106	8,757.900	51,853.000
49	1,492.027	1,546.516	2,011.181	4,091.202	537.734	1,156.783	2,483.887	4,962.522	436.000	755.423	1,318.442	3,977.285	24,769.000
50	4,920.257	5,041.449	4,964.892	5,710.812	2,301.297	5,288.923	7,102.817	5,944.372	983.184	2,160.988	3,613.342	10,612.666	58,645.000
51	313.430	231.159	213.586	117.816	231.907	239.271	231.721	173.094	808.655	1,186.317	1,728.693	4,137.351	9,613.000
52	1,090.605	1,105.839	1,347.564	2,269.594	499.425	927.393	1,616.445	2,770.339	382.745	544.826	800.420	1,973.804	15,329.000
53	1,034.356	1,064.052	1,342.378	2,372.816	460.361	882.769	1,604.831	2,865.642	357.138	523.041	790.663	2,030.953	15,329.000
54	878.939	958.664	1,306.553	2,669.447	392.090	753.806	1,438.713	3,228.994	342.526	509.920	782.746	2,066.604	15,329.000
55	3,068.410	2,990.022	3,198.335	4,173.206	1,763.899	3,268.313	4,182.519	4,215.243	315.196	449.464	693.412	1,462.980	29,781.000
56	2,762.427	2,862.869	3,005.549	4,230.744	1,501.359	2,923.095	4,033.579	4,403.556	264.999	369.966	579.792	1,284.067	28,222.000
57	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2,017.517	5,560.895	7,645.735	12,997.853	28,222.000
58	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	2,018.087	5,414.838	7,584.618	13,204.457	28,222.000
59	13,393.820	13,479.698	13,655.330	14,436.751	11,989.758	13,954.229	14,348.925	14,672.687	8,794.312	13,478.864	6,980.237	8,777.388	147,962.000
60	13,712.059	13,802.174	13,973.006	14,787.065	12,232.563	14,257.289	14,664.194	15,120.259	9,196.562	13,244.073	7,350.991	8,811.766	151,152.000
61	2,633.434	2,613.149	2,827.863	3,703.897	1,587.743	2,685.168	3,482.428	4,023.004	793.247	1,121.289	1,334.958	2,483.821	29,290.000
62	3,975.553	3,921.586	4,108.801	5,007.222	2,571.222	4,181.180	4,857.817	5,402.943	1,200.018	1,932.363	2,300.412	3,710.883	43,170.000
63	2,334.280	2,315.652	2,498.164	3,321.542	1,392.755	2,364.636	3,088.539	3,623.707	876.754	1,314.343	1,601.453	2,919.174	27,651.000
64	3,967.320	3,908.655	4,103.976	5,033.210	2,510.083	4,172.609	4,900.071	5,430.399	1,170.417	1,907.419	2,286.831	3,779.010	43,170.000
65	3,199.919	3,262.945	3,807.848	6,298.143	1,460.766	2,818.152	5,777.762	6,512.174	867.421	1,906.123	3,858.611	5,676.136	45,446.000
66	3,110.164	3,173.472	3,786.036	6,499.182	1,068.245	2,930.115	6,052.130	6,518.363	488.425	1,006.613	2,089.463	8,723.790	45,446.000
67	4,629.073	4,572.068	4,776.371	5,653.059	3,326.842	4,885.161	5,487.890	5,930.678	1,937.743	3,182.863	3,443.585	4,890.667	52,716.000
68	6,241.098	6,208.467	6,418.867	7,305.662	4,846.005	6,592.831	7,137.034	7,598.225	2,594.011	4,174.466	4,208.533	5,348.799	68,674.000
69	6,224.496	5,987.666	2,480.461	950.623	10,928.228	2,823.068	1,238.584	653.366	7,244.615	12,736.098	8,265.368	9,141.427	68,674.000
70	3,981.386	3,927.743	4,105.425	4,998.608	2,601.460	4,172.528	4,831.870	5,407.304	1,416.263	2,476.648	2,912.206	4,614.560	45,446.000
71	3,983.007	3,933.967	4,111.458	4,984.730	2,627.762	4,182.127	4,818.990	5,384.283	1,442.316	2,512.708	2,896.549	4,568.104	45,446.000
72	3,971.208	3,911.635	4,092.550	4,882.427	2,641.389	4,199.176	4,858.440	5,158.815	1,427.554	2,521.314	2,981.627	4,799.865	45,446.000
73	3,955.030	3,889.091	4,082.079	5,086.962	2,384.848	4,125.372	4,937.163	5,565.779	1,323.127	2,388.792	2,873.323	4,834.435	45,446.000

Table 4 – 4 Blocks

Product	Period: 4 Blocks												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	5,178.380	5,393.984	8,520.683	2,531.009	7,790.927	8,771.111	2,487.611	9,092.429	7,513.006	896.469	1,982.509	9,666.882	69,825.000
2	32,767.802	32,836.149	36,098.741	29,455.708	38,089.962	34,157.021	33,900.879	33,900.266	33,901.547	25,996.158	19,092.844	11,892.923	362,090.000
3	32,581.111	32,555.542	32,438.272	32,678.016	33,659.470	31,237.439	33,949.523	33,351.614	30,273.788	29,711.388	29,180.178	10,772.660	362,389.000
4	9,054.920	9,202.063	13,731.830	5,274.415	10,915.684	15,798.714	2,770.412	12,879.694	16,338.708	3,138.423	4,756.771	5,558.366	109,420.000
5	11,552.200	11,495.714	12,254.407	10,756.188	12,059.754	12,486.378	8,925.077	13,420.138	12,957.105	11,011.198	15,372.843	8,683.999	140,975.000
6	47,736.707	47,659.709	6,163.172	89,017.280	8,315.941	342.263	99,705.981	4.017	0.425	158,916.106	48,396.207	21,323.195	527,581.000
7	57,361.944	57,445.531	58,650.676	55,910.456	59,039.044	58,508.650	56,188.320	59,233.811	58,036.019	48,954.434	40,761.340	19,069.777	629,160.000
8	16,622.128	16,524.298	17,657.714	15,222.402	18,169.397	17,412.341	16,459.937	17,408.807	16,935.396	7,242.876	11,330.678	14,812.024	185,798.000
9	36,676.774	36,587.514	3,174.903	69,851.692	10,078.557	393.046	73,235.271	0.020	0.002	115,923.075	37,381.948	23,035.198	406,338.000
10	5,753.777	5,818.228	8,540.417	3,334.012	8,870.709	7,907.701	4,327.404	9,005.692	6,779.326	1,377.038	2,937.295	8,902.401	73,554.000
11	5,800.982	5,852.513	8,458.927	3,458.155	8,963.697	7,690.569	4,671.924	8,733.972	6,706.525	1,397.586	2,942.817	8,876.331	73,554.000
12	3,934.264	3,940.218	5,595.869	2,514.121	5,811.784	5,144.447	4,294.868	3,900.876	5,274.608	1,021.188	1,992.587	5,838.171	49,263.000
13	34,274.757	34,135.367	36,420.071	30,688.618	38,448.168	35,693.410	34,753.832	35,069.339	35,007.024	13,507.282	25,304.563	30,076.570	383,379.000
14	11,606.785	11,447.942	4,768.038	16,902.849	10,079.693	840.223	21,527.933	5,929.441	365.391	10,528.478	12,441.584	12,182.643	118,621.000
15	18,715.990	18,941.786	20,161.607	17,046.591	20,458.013	20,314.780	17,102.499	21,002.500	19,714.385	3,372.557	4,802.840	6,081.453	187,715.000
16	1,720.661	1,900.525	3,542.329	997.564	2,855.134	3,310.818	1,640.339	3,129.980	2,393.196	482.162	947.772	3,277.519	26,198.000
17	9,080.485	9,026.918	10,673.535	7,198.648	11,211.077	10,371.213	8,670.021	10,512.743	9,598.174	2,879.642	5,386.325	10,647.220	105,256.000
18	0.000	0.000	0.000	0.000	0.000	0.000	2,149.851	1,689.784	2,007.995	1,124.446	1,957.428	6,296.497	15,226.000
19	1,934.524	1,947.404	2,755.494	1,225.723	2,509.429	2,902.270	1,091.411	2,623.758	2,922.252	477.896	901.692	2,982.146	24,274.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	4,225.515	5,412.730	14,635.755	24,274.000
21	1,855.931	1,915.033	2,362.924	1,473.479	2,571.742	2,088.667	1,473.390	2,912.460	1,748.037	652.377	1,188.575	4,031.387	24,274.000
22	4,350.547	4,081.300	6,249.260	1,926.007	6,009.476	6,745.624	1,678.743	6,706.035	6,296.329	742.735	1,629.288	8,424.657	54,840.000
23	920.158	1,034.535	4,023.619	409.139	1,359.225	4,209.948	529.110	1,789.645	4,018.000	310.785	625.663	3,070.173	22,300.000
24	7,413.902	7,375.643	8,289.169	6,291.777	8,229.084	8,557.852	5,108.179	9,131.150	8,839.384	3,100.744	3,997.297	6,258.819	82,593.000
25	9,197.755	9,165.131	10,146.113	7,957.117	10,011.566	10,540.316	6,520.639	11,079.819	10,908.541	4,017.800	5,089.866	7,328.337	101,963.000
26	1,910.258	1,980.696	2,896.903	1,210.364	2,301.098	3,276.395	981.604	2,274.231	3,532.022	586.133	877.348	2,196.948	24,024.000
27	4,876.112	4,879.766	5,856.193	3,690.015	5,663.503	6,258.553	2,975.956	6,071.659	6,564.456	1,821.394	2,716.958	5,573.436	56,948.000
28	2,189.681	2,214.970	3,061.991	1,445.237	2,632.676	3,388.730	1,147.866	2,681.503	3,637.274	749.636	1,119.417	2,684.019	26,953.000
29	1,723.839	1,765.258	2,619.974	1,081.004	2,031.679	2,996.388	887.374	1,985.671	3,236.026	490.959	722.676	1,814.152	21,355.000
30	2,617.472	2,638.730	3,568.012	1,776.313	3,073.882	3,974.020	1,419.279	3,118.805	4,286.130	724.011	1,052.521	2,404.826	30,654.000
31	7,875.146	7,769.496	8,893.573	6,365.110	8,825.326	9,347.778	5,162.021	9,757.414	9,618.779	2,193.942	3,615.781	7,173.635	86,598.000
32	2,517.561	2,484.896	2,693.125	2,004.820	3,219.300	2,471.462	1,805.904	3,548.171	2,341.507	1,263.811	2,149.414	5,509.028	32,009.000
33	6,124.036	6,169.479	7,337.057	4,575.703	7,237.302	7,817.567	3,508.526	8,051.521	8,070.525	1,948.077	3,295.029	7,439.180	71,574.000
34	3,808.498	3,821.419	4,739.191	2,589.844	4,826.745	4,952.520	2,135.504	5,271.116	4,962.479	1,103.862	1,878.032	4,640.788	44,730.000
35	4,884.439	4,889.922	5,837.709	3,715.570	5,611.362	6,285.138	2,953.614	6,011.851	6,646.606	1,805.035	2,727.931	5,578.822	56,948.000
36	4,885.776	4,887.943	5,838.352	3,713.635	5,606.661	6,291.775	2,948.981	6,011.713	6,651.377	1,831.972	2,759.463	5,520.354	56,948.000
37	4,739.164	4,767.721	3,790.811	5,475.498	5,529.138	2,293.060	5,824.543	5,626.153	1,846.999	3,542.871	5,067.115	8,444.927	56,948.000
38	4,707.263	4,734.827	6,129.630	3,187.468	5,781.942	6,602.309	2,517.964	6,031.385	7,022.370	1,411.572	2,841.249	5,980.023	56,948.000
39	30,568.945	30,621.687	31,562.751	29,287.406	31,657.801	31,808.175	27,230.337	33,564.548	31,958.498	19,054.994	19,095.107	22,768.752	339,179.000
40	29,488.120	29,454.568	30,639.639	27,869.459	30,555.000	31,157.868	24,731.633	33,485.867	31,364.826	14,707.377	20,557.679	22,520.965	326,533.000
41	6,240.147	6,380.827	7,594.507	4,565.474	7,664.268	7,985.739	3,067.748	8,930.167	8,217.565	1,730.833	2,938.210	7,252.515	72,568.000
42	4,099.030	4,170.938	5,207.019	2,794.360	5,249.793	5,432.835	2,785.269	5,872.605	4,819.113	1,162.668	1,975.104	5,705.267	49,274.000
43	3,605.619	3,457.899	6,410.114	1,511.480	5,203.071	6,759.082	1,467.881	5,889.649	6,116.102	743.577	1,495.547	6,613.977	49,274.000
44	12,405.640	12,252.092	13,394.937	10,783.574	13,405.431	13,863.665	8,154.000	15,774.011	14,124.660	4,581.361	7,168.052	11,892.578	137,800.000
45	14,254.855	14,059.882	15,287.281	12,581.485	15,228.736	15,791.797	10,205.617	17,420.711	15,975.689	5,251.748	8,594.389	13,596.810	158,249.000
46	4,525.112	4,579.224	5,561.798	3,264.435	5,505.418	5,896.281	2,334.048	6,217.133	6,114.952	1,348.019	2,232.663	5,749.919	53,329.000
47	1,986.112	2,172.355	4,561.937	776.638	2,418.287	5,525.478	707.995	2,311.391	5,701.018	429.463	818.584	3,079.743	30,489.000
48	3,884.948	3,831.526	6,553.278	1,616.671	5,164.105	7,488.975	1,347.594	5,077.360	7,844.797	841.834	1,698.925	6,502.988	51,853.000
49	1,471.365	1,713.497	3,670.832	686.072	1,854.007	4,315.615	624.334	1,799.771	4,431.592	436.302	809.672	2,955.940	24,769.000
50	4,875.250	4,876.746	5,726.062	3,000.521	6,651.862	5,825.674	2,375.811	7,281.624	5,820.634	1,219.470	2,505.763	8,485.584	58,645.000
51	236.975	229.583	190.437	231.296	227.214	198.484	201.574	207.166	248.254	1,045.884	1,680.433	4,915.700	9,613.000
52	1,089.646	1,197.095	2,073.461	626.092	1,313.341	2,420.769	520.157	1,257.825	2,582.219	313.151	496.910	1,438.335	15,329.000
53	1,021.499	1,181.945	2,156.757	566.786	1,287.872	2,505.544	505.506	1,236.653	2,618.043	300.237	486.149	1,462.009	15,329.000
54	874.791	1,118.127	2,367.284	484.363	1,128.562	2,747.277	442.906	1,096.596	2,820.701	286.868	475.836	1,485.691	15,329.000
55	2,948.439	3,052.917	4,071.124	2,007.789	3,831.090	4,233.602	1,797.911	3,943.921	3,894.208	0.000	0.000	0.000	29,781.000
56	2,794.058	2,794.708	4,057.425	1,732.561	3,524.213	4,389.418	1,491.876	3,551.942	3,885.800	0.000	0.000	0.000	28,222.000
57	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	4,276.870	7,584.055	16,361.075	28,222.000
58	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	4,278.290	7,506.974	16,436.735	28,222.000
59	13,448.857	13,489.768	14,285.575	12,469.377	14,230.643	14,524.181	11,292.981	15,103.234	14,827.985	7,233.150	8,280.537	8,775.714	147,962.000
60	13,761.513	13,819.213	14,625.002	12,728.706	14,529.774	14,947.249	11,452.220	15,441.573	15,311.935	7,250.161	8,406.392	8,878.263	151,152.000
61	2,627.596	2,696.042	3,510.119	1,846.699	3,141.126	3,845.932	1,445.850	3,129.328	4,258.579	487.136	664.724	1,636.869	29,290.000
62	3,955.837	3,987.023	4,817.011	2,994.580	4,583.774	5,181.518	2,447.796	4,733.070	5,579.006	886.428	1,299.636	2,704.322	43,170.000
63	2,328.888	2,382.913	3,140.428	1,613.581	2,769.369	3,469.278	1,290.478	2,778.334	3,783.416	719.347	1,019.230	2,355.737	27,651.000
64	3,943.601	3,978.743	4,837.527	2,932.567	4,623.701	5,203.603	2,366.126	4,774.349	5,619.396	870.949	1,289.822	2,729.615	43,170.000
65	3,264.940	3,352.585	5,809.115	1,645.699	4,248.124	6,532.818	1,378.489	4,720.148	6,328.003	866.663	1,618.874	5,680.541	45,446.000
66	3,199.293	3,157.374	6,069.973	1,271.402	4,462.878	6,692.361	1,152.006	4,964.124	6,310.510	547.062	1,197.106	6,421.910	45,446.000
67	4,624.718	4,642.117	5,456.094	3,696.485	5,238.643	5,787.800	2,963.937	5,612.140	6,146.852	1,770.563	2,428.309	4,348.342	52,716.000
68	6,259.008	6,263.236	7,108.327	5,269.671	6,925.210	7,435.690	4,349.510	7,477.754	7,803.307	2,179.570	2,893.107	4,709.609	68,674.000
69	6,219.689	4,621.358	891.388	9,028.485	1,961.440	742.510	7,512.330	3,237.179	982.925	10,851.738	10,202.009	12,422.949	68,674.000
70	3,962.065	3,993.647	4,804.159	3,012.375	4,561.262	5,186.235	2,460.155	4,702.904	5,596.812	1,319.722	1,981.974	3,864.690	45,446.000
71	3,966.914	3,995.089	4,797.869	3,028.626	4,565.664	5,165.582	2,457.360	4,723.958	5,578.554	1,338.222	1,983.944	3,844.219	45,446.000
72	3,947.265	3,983.584	4,712.516	3,052.583	4,585.826	5,004.956	2,546.132	4,769.517	5,327.716	1,374.311	2,069.736	4,071.857	45,446.000
73	3,927.821	3,963.806	4,868.244	2,831.438	4,614.637	5,313.796	2,220.422	4,724.546	5,814.903	1,244.541	1,930.537	3,991.307	45,446.000

Table 5 – 6 Blocks

Product	Period: 6 Blocks												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	4,945.873	7,782.825	3,768.150	8,960.548	4,018.908	8,709.790	4,713.471	8,015.226	4,946.898	7,781.800	1,046.039	5,135.471	69,825.000
2	32,823.630	34,978.165	33,605.831	34,195.964	33,685.046	34,116.749	33,853.939	33,947.855	33,870.471	33,931.323	11,477.652	11,603.375	362,090.000
3	32,626.919	32,423.031	33,003.995	32,045.955	33,969.222	31,080.728	34,516.654	30,533.295	35,121.209	29,928.741	19,495.597	17,643.653	362,389.000
4	9,143.285	12,182.591	6,535.138	14,790.737	5,665.992	15,659.883	5,292.030	16,033.846	2,702.556	16,115.181	1,812.421	3,486.340	109,420.000
5	11,572.445	11,962.435	11,139.397	12,395.483	10,997.441	12,537.439	10,753.306	12,781.574	10,235.073	15,807.946	5,588.024	15,204.437	140,975.000
6	47,712.445	24,840.786	60,512.818	570.732	66,375.334	480.246	70,876.167	83.188	53,065.745	56.777	146,420.581	56,586.180	527,581.000
7	57,373.879	58,264.888	56,757.711	58,881.056	57,047.792	58,590.975	57,318.493	58,320.273	57,308.193	58,330.573	24,882.612	26,083.556	629,160.000
8	16,563.025	17,306.402	16,214.961	17,654.466	16,484.520	17,384.907	16,715.558	17,153.869	16,821.233	17,048.194	5,100.670	11,350.195	185,798.000
9	36,729.328	9,383.293	56,909.628	672.674	51,277.528	532.744	47,560.448	146.049	35,462.592	75.138	102,625.820	64,962.758	406,338.000
10	5,804.519	7,603.762	4,864.829	8,543.452	5,615.387	7,792.894	6,206.802	7,201.480	6,469.951	6,938.330	1,376.252	5,136.341	73,554.000
11	5,846.893	7,561.389	5,022.706	8,385.575	5,835.209	7,573.072	6,380.622	7,027.659	6,568.916	6,839.366	1,408.923	5,103.670	73,554.000
12	3,874.172	5,106.062	3,481.221	5,499.013	3,870.705	5,109.530	4,217.841	4,762.393	4,351.710	4,628.524	1,013.524	3,348.304	49,263.000
13	34,240.250	35,646.547	33,719.530	36,167.267	34,161.420	35,725.376	34,612.780	35,274.017	34,750.480	35,136.317	9,818.009	24,127.007	383,379.000
14	11,475.323	7,073.187	15,567.013	2,981.497	17,261.550	1,286.961	17,760.152	788.358	22,241.731	1,323.056	7,833.773	13,028.400	118,621.000
15	18,789.432	19,756.823	18,099.912	20,446.344	18,107.176	20,439.079	18,344.610	20,201.645	18,518.789	15,011.189	0.000	0.000	187,715.000
16	1,739.923	3,035.754	1,270.530	3,505.147	1,607.598	3,168.079	2,024.154	2,751.523	2,204.795	2,570.882	523.363	1,796.251	26,198.000
17	9,056.182	10,131.110	8,340.234	10,847.058	8,875.420	10,311.871	9,271.458	9,915.833	9,452.039	9,735.253	2,385.158	6,934.384	105,256.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	10,548.027	3,329.837	313.658	1,034.477	15,226.000
19	1,823.029	2,601.919	1,520.291	2,904.657	1,519.453	2,905.494	1,652.075	2,772.872	1,400.472	3,024.476	472.620	1,676.640	24,274.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	18,308.246	3,816.494	467.936	1,681.325	24,274.000
21	1,774.432	2,314.826	1,693.516	2,395.742	1,858.227	2,231.031	2,336.873	1,752.385	2,398.407	3,158.296	495.104	1,865.160	24,274.000
22	4,177.986	5,609.418	2,840.317	6,947.087	2,953.244	6,834.160	3,505.940	6,281.464	3,664.642	7,170.115	813.886	4,041.739	54,840.000
23	904.189	3,081.353	582.177	3,403.364	650.713	3,334.828	844.594	3,140.947	895.209	3,488.602	365.011	1,609.012	22,300.000
24	7,384.849	8,000.960	6,978.320	8,407.489	6,673.545	8,712.263	6,294.615	9,091.194	6,140.042	9,245.767	1,791.442	3,872.513	82,593.000
25	9,179.797	9,826.202	8,723.348	10,282.651	8,296.942	10,709.057	7,820.160	11,185.839	7,550.352	11,455.647	2,234.011	4,698.993	101,963.000
26	1,887.248	2,637.990	1,530.236	2,995.002	1,407.543	3,117.695	1,260.493	3,264.745	1,191.746	3,333.492	386.056	1,011.755	24,024.000
27	4,910.435	5,497.612	4,338.988	6,069.059	4,128.316	6,279.731	3,919.209	6,488.838	3,849.165	6,558.883	1,365.857	3,541.907	56,948.000
28	2,175.880	2,801.882	1,794.404	3,183.357	1,647.486	3,330.275	1,509.044	3,468.717	1,445.956	3,531.806	561.621	1,502.571	26,953.000
29	1,709.241	2,363.473	1,360.052	2,712.662	1,223.182	2,849.532	1,116.637	2,956.077	1,048.132	3,024.582	280.791	710.638	21,355.000
30	2,608.453	3,274.357	2,182.991	3,699.818	1,982.366	3,900.443	1,786.289	4,096.520	1,654.280	4,228.529	358.770	881.184	30,654.000
31	7,835.600	8,523.210	7,177.841	9,180.969	6,877.957	9,480.852	6,739.694	9,619.115	6,589.358	9,769.452	1,311.896	3,492.057	86,598.000
32	2,527.758	2,602.630	2,359.957	2,770.431	2,505.658	2,624.730	2,603.718	2,526.670	2,654.842	2,475.546	1,492.526	4,864.532	32,009.000
33	6,160.595	6,926.452	5,439.538	7,647.510	5,171.518	7,915.530	5,075.957	8,011.091	4,957.157	8,129.890	1,539.261	4,599.502	71,574.000
34	3,834.886	4,411.187	3,268.944	4,977.128	3,208.658	5,037.415	3,254.465	4,991.608	3,186.387	4,952.456	928.973	2,677.895	44,730.000
35	4,921.043	5,487.004	4,359.517	6,048.530	4,083.951	6,324.097	3,830.506	6,577.541	3,505.113	6,902.934	1,322.494	3,585.270	56,948.000
36	4,918.043	5,490.004	4,354.033	6,054.014	4,081.712	6,326.335	3,823.783	6,584.264	3,619.103	6,788.944	1,345.086	3,562.678	56,948.000
37	4,720.502	4,144.628	5,559.318	3,305.812	6,352.007	2,513.123	6,650.954	2,214.176	5,661.873	2,845.943	3,487.725	9,491.937	56,948.000
38	4,673.402	5,707.744	4,041.892	6,339.254	3,756.108	6,625.038	3,425.490	6,955.656	3,088.903	6,411.776	1,405.468	4,517.269	56,948.000
39	30,607.878	31,227.710	29,982.682	31,852.907	29,725.607	32,109.981	29,082.607	32,752.982	29,713.857	32,121.731	10,428.743	19,572.316	339,179.000
40	29,480.805	30,240.746	28,627.296	31,094.255	28,180.940	31,540.611	27,389.058	32,332.494	28,245.456	31,476.095	9,013.045	18,912.199	326,533.000
41	6,233.003	7,243.985	5,507.705	7,969.282	5,283.567	8,193.420	4,932.959	8,544.028	5,231.602	8,245.385	1,339.621	3,843.443	72,568.000
42	4,168.092	4,816.566	3,479.786	5,504.872	3,540.890	5,443.769	3,737.760	5,246.898	3,814.087	5,170.571	1,171.733	3,178.976	49,274.000
43	3,494.999	5,487.423	2,187.358	6,795.064	2,346.713	6,635.709	2,735.086	6,247.336	3,002.863	4,122.888	1,219.163	4,999.399	49,274.000
44	12,348.548	13,019.899	11,502.471	13,865.976	11,092.384	14,276.062	10,631.242	14,737.204	11,102.224	14,266.222	3,091.730	7,866.037	137,800.000
45	14,159.129	14,908.883	13,342.563	15,725.448	12,981.010	16,087.001	12,429.357	16,638.654	13,007.601	16,060.411	3,642.410	9,266.532	158,249.000
46	4,527.563	5,249.859	3,928.749	5,848.673	3,749.536	6,027.886	3,512.004	6,265.418	3,719.294	6,058.128	1,169.219	3,272.670	53,329.000
47	2,105.878	3,707.724	1,076.936	4,736.667	1,006.611	4,806.992	1,078.191	4,735.411	1,066.953	4,746.650	337.598	1,083.390	30,489.000
48	3,897.743	5,615.425	2,183.278	7,329.889	2,272.591	7,240.577	2,461.548	7,051.620	2,530.206	6,982.961	916.233	3,370.930	51,853.000
49	1,615.202	2,955.261	883.144	3,687.319	880.188	3,690.275	964.813	3,605.650	1,041.387	3,564.462	440.975	1,440.325	24,769.000
50	4,896.769	5,421.936	3,917.478	6,401.227	4,079.274	6,239.431	4,518.565	5,800.140	4,248.627	6,195.538	1,368.198	5,557.818	58,645.000
51	305.003	132.993	225.945	212.051	242.117	195.879	218.712	219.284	1,309.698	984.969	1,343.422	4,222.927	9,613.000
52	1,154.263	1,752.538	763.088	2,143.714	742.246	2,164.555	705.108	2,201.694	710.607	2,196.194	226.202	568.792	15,329.000
53	1,107.302	1,799.499	739.315	2,167.486	717.647	2,189.154	677.790	2,229.011	659.372	2,247.429	225.323	569.671	15,329.000
54	1,002.062	1,904.739	637.590	2,269.211	620.494	2,286.307	584.798	2,322.003	601.572	2,305.229	222.854	572.140	15,329.000
55	3,020.869	3,694.118	2,513.372	4,201.615	2,743.807	3,971.180	2,513.233	4,201.754	1,574.488	1,346.564	0.000	0.000	29,781.000
56	2,788.564	3,642.230	2,219.724	4,211.070	2,073.729	4,357.065	2,089.969	4,340.825	1,126.707	1,372.116	0.000	0.000	28,222.000
57	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	8,653.857	19,568.143	28,222.000
58	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	8,539.021	19,682.979	28,222.000
59	13,444.585	14,038.214	13,067.010	14,415.790	12,789.801	14,692.999	12,427.964	15,054.836	12,239.581	15,243.219	3,903.434	6,644.567	147,962.000
60	13,752.571	14,384.581	13,359.981	14,777.171	13,008.418	15,128.734	12,627.409	15,509.744	12,350.588	15,786.564	3,846.431	6,619.807	151,152.000
61	2,611.027	3,278.145	2,224.618	3,664.553	2,046.665	3,842.507	1,837.112	4,052.060	1,631.481	4,101.834	0.000	0.000	29,290.000
62	4,002.943	4,503.638	3,463.779	5,042.802	3,303.958	5,202.623	3,102.962	5,403.619	2,966.796	5,539.785	175.680	461.415	43,170.000
63	2,314.545	2,920.274	1,938.268	3,296.552	1,787.509	3,447.310	1,626.981	3,607.838	1,524.090	3,710.729	422.904	1,054.001	27,651.000
64	3,992.335	4,514.246	3,441.498	5,065.083	3,270.126	5,236.455	3,081.702	5,424.879	2,920.909	5,585.671	177.339	459.756	43,170.000
65	3,249.304	5,035.123	2,154.752	6,129.675	2,097.632	6,186.795	2,281.026	6,003.401	2,385.475	5,898.952	900.976	3,122.889	45,446.000
66	3,102.735	5,181.692	1,918.059	6,366.368	1,943.083	6,341.344	2,260.922	6,023.505	2,326.212	5,958.215	693.580	3,330.285	45,446.000
67	4,600.902	5,214.384	4,173.631	5,641.655	3,918.913	5,896.373	3,623.465	6,191.821	3,429.514	6,385.772	1,114.093	2,525.479	52,716.000
68	6,237.386	6,849.662	5,787.902	7,299.146	5,518.677	7,568.371	5,173.136	7,913.911	4,905.121	8,181.927	1,013.308	2,225.455	68,674.000
69	6,116.810	1,704.813	6,770.380	1,051.243	6,470.425	1,351.198	6,285.065	1,536.558	6,335.632	1,641.848	11,037.502	18,372.525	68,674.000
70	4,002.719	4,503.862	3,474.134	5,032.447	3,284.466	5,222.115	3,076.353	5,430.228	2,894.624	5,611.957	819.572	2,093.524	45,446.000
71	4,008.076	4,498.505	3,485.193	5,021.388	3,302.533	5,204.048	3,105.550	5,401.031	2,944.873	5,561.708	828.080	2,085.016	45,446.000
72	3,989.894	4,439.016	3,506.184	4,922.726	3,370.618	5,058.292	3,215.179	5,213.731	3,049.489	5,379.422	930.062	2,371.387	45,446.000
73	3,975.890	4,530.691	3,357.129	5,149.452	3,133.273	5,373.308	2,903.674	5,602.907	2,699.503	5,807.078	803.224	2,109.871	45,446.000

Table 6 – 12 Blocks

Product	Period: 12 Blocks												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
1	6,364.349	6,364.349	6,364.349	6,364.349	6,364.349	6,364.349	6,364.349	6,364.349	6,364.349	6,364.349	6,181.510	0.000	69,825.000
2	33,900.894	33,900.900	33,900.897	33,900.897	33,900.897	33,900.897	33,900.897	33,900.897	33,900.897	33,900.897	23,081.028	0.000	362,090.000
3	32,524.975	32,524.975	32,524.975	32,524.975	32,524.975	32,524.975	32,524.975	32,524.975	32,524.975	32,524.975	37,139.250	0.000	362,389.000
4	10,662.940	10,662.936	10,662.938	10,662.938	10,662.938	10,662.938	10,662.938	10,662.938	10,662.938	8,154.799	5,298.761	0.000	109,420.000
5	11,767.440	11,767.440	11,767.440	11,767.440	11,767.440	11,767.440	11,767.440	11,767.440	11,767.440	14,275.579	20,792.461	0.000	140,975.000
6	35,465.403	37,450.894	33,272.607	36,006.868	33,315.957	30,887.977	27,913.770	37,093.708	28,637.684	28,532.341	62,119.934	136,883.858	527,581.000
7	57,819.381	57,819.385	57,819.383	57,819.383	57,819.383	57,819.383	57,819.383	57,819.383	57,819.383	57,819.383	50,966.167	0.000	629,160.000
8	16,934.713	16,934.714	16,934.714	16,934.714	16,934.714	16,934.714	16,934.714	16,934.714	16,934.714	16,934.714	16,450.865	0.000	185,798.000
9	23,867.534	21,882.018	26,060.319	23,326.058	26,016.969	28,444.949	31,419.156	22,239.218	25,642.082	5,848.147	26,641.500	144,950.048	406,338.000
10	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,512.594	0.000	73,554.000
11	6,704.140	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,704.141	6,512.594	0.000	73,554.000
12	4,490.116	4,490.118	4,490.117	4,490.117	4,490.117	4,490.117	4,490.117	4,490.117	4,490.117	4,490.117	4,361.828	0.000	49,263.000
13	34,943.398	34,943.398	34,943.398	34,943.398	34,943.398	34,943.398	34,943.398	34,943.398	34,943.398	34,943.398	33,945.016	0.000	383,379.000
14	9,274.249	9,274.261	9,274.255	9,274.255	9,274.255	9,274.255	9,274.255	9,274.255	9,274.255	14,290.533	20,862.173	0.000	118,621.000
15	19,273.126	19,273.129	19,273.128	19,273.128	19,273.128	19,273.128	19,273.128	19,273.128	19,273.128	14,256.850	0.000	0.000	187,715.000
16	2,387.838	2,387.839	2,387.839	2,387.839	2,387.839	2,387.839	2,387.839	2,387.839	2,387.839	2,387.839	2,319.615	0.000	26,198.000
17	9,593.645	9,593.646	9,593.646	9,593.646	9,593.646	9,593.646	9,593.646	9,593.646	9,593.646	9,593.646	9,319.542	0.000	105,256.000
18	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	5,847.732	8,030.131	1,348.135	0.000	15,226.000
19	2,212.474	2,212.474	2,212.474	2,212.474	2,212.474	2,212.474	2,212.474	2,212.474	2,212.474	2,212.474	2,149.260	0.000	24,274.000
20	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	22,124.739	2,149.260	0.000	24,274.000
21	2,044.629	2,044.629	2,044.629	2,044.629	2,044.629	2,044.629	2,044.629	2,044.629	2,044.629	3,512.074	2,360.264	0.000	24,274.000
22	4,893.701	4,893.703	4,893.702	4,893.702	4,893.702	4,893.702	4,893.702	4,893.702	4,893.702	5,941.055	4,855.625	0.000	54,840.000
23	1,992.774	1,992.767	1,992.771	1,992.771	1,992.771	1,992.771	1,992.771	1,992.771	2,351.214	2,032.598	1,974.023	0.000	22,300.000
24	7,692.904	7,692.905	7,692.904	7,692.904	7,692.904	7,692.904	7,692.904	7,692.904	7,692.904	7,692.904	5,663.956	0.000	82,593.000
25	9,502.999	9,503.000	9,503.000	9,503.000	9,503.000	9,503.000	9,503.000	9,503.000	9,503.000	9,503.000	6,933.004	0.000	101,963.000
26	2,262.619	2,262.619	2,262.619	2,262.619	2,262.619	2,262.619	2,262.619	2,262.619	2,262.619	2,262.619	1,397.811	0.000	24,024.000
27	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	4,907.764	0.000	56,948.000
28	2,488.881	2,488.881	2,488.881	2,488.881	2,488.881	2,488.881	2,488.881	2,488.881	2,488.881	2,488.881	2,064.192	0.000	26,953.000
29	2,036.357	2,036.357	2,036.357	2,036.357	2,036.357	2,036.357	2,036.357	2,036.357	2,036.357	2,036.357	991.429	0.000	21,355.000
30	2,941.405	2,941.405	2,941.405	2,941.405	2,941.405	2,941.405	2,941.405	2,941.405	2,941.405	2,941.405	1,239.954	0.000	30,654.000
31	8,179.392	8,179.418	8,179.405	8,179.405	8,179.405	8,179.405	8,179.405	8,179.405	8,179.405	8,179.405	4,803.953	0.000	86,598.000
32	2,565.195	2,565.193	2,565.194	2,565.194	2,565.194	2,565.194	2,565.194	2,565.194	2,565.194	2,565.194	6,357.059	0.000	32,009.000
33	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,138.762	0.000	71,574.000
34	4,123.048	4,123.024	4,123.036	4,123.036	4,123.036	4,123.036	4,123.036	4,123.036	4,123.036	4,015.806	3,606.868	0.000	44,730.000
35	5,204.023	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	4,907.764	0.000	56,948.000
36	5,204.023	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	5,204.024	4,907.764	0.000	56,948.000
37	4,432.567	4,432.564	4,432.565	4,432.565	4,432.565	4,432.565	4,432.565	4,432.565	4,432.565	4,075.251	12,979.662	0.000	56,948.000
38	5,190.573	5,190.573	5,190.573	5,190.573	5,190.573	5,190.573	5,190.573	5,190.573	5,190.573	4,310.106	5,922.738	0.000	56,948.000
39	30,917.794	30,917.794	30,917.794	30,917.794	30,917.794	30,917.794	30,917.794	30,917.794	30,917.794	30,917.794	30,001.058	0.000	339,179.000
40	29,860.775	29,860.776	29,860.776	29,860.776	29,860.776	29,860.776	29,860.776	29,860.776	29,860.776	29,860.776	27,925.245	0.000	326,533.000
41	6,738.494	6,738.494	6,738.494	6,738.494	6,738.494	6,738.494	6,738.494	6,738.494	6,738.494	6,738.494	5,183.064	0.000	72,568.000
42	4,492.329	4,492.329	4,492.329	4,492.329	4,492.329	4,492.329	4,492.329	4,492.329	4,492.329	4,492.329	4,350.709	0.000	49,274.000
43	4,491.211	4,491.211	4,491.211	4,491.211	4,491.211	4,491.211	4,491.211	4,491.211	4,491.211	2,634.540	6,218.562	0.000	49,274.000
44	12,684.223	12,684.223	12,684.223	12,684.223	12,684.223	12,684.223	12,684.223	12,684.223	12,684.223	12,684.223	10,957.768	0.000	137,800.000
45	14,534.006	14,534.006	14,534.006	14,534.006	14,534.006	14,534.006	14,534.006	14,534.006	14,534.006	14,534.006	12,908.942	0.000	158,249.000
46	4,888.711	4,888.711	4,888.711	4,888.711	4,888.711	4,888.711	4,888.711	4,888.711	4,888.711	4,888.711	4,441.890	0.000	53,329.000
47	2,906.801	2,906.801	2,906.801	2,906.801	2,906.801	2,906.801	2,906.801	2,906.801	2,906.801	2,906.801	1,420.988	0.000	30,489.000
48	4,756.584	4,756.584	4,756.584	4,756.584	4,756.584	4,756.584	4,756.584	4,756.584	4,756.584	4,756.584	4,287.163	0.000	51,853.000
49	2,285.227	2,285.235	2,285.231	2,285.231	2,285.231	2,285.231	2,285.231	2,285.231	2,285.231	2,320.617	1,881.300	0.000	24,769.000
50	5,159.353	5,159.352	5,159.353	5,159.353	5,159.352	5,159.353	5,159.352	5,159.353	5,159.353	5,284.812	6,926.016	0.000	58,645.000
51	218.999	218.997	218.998	218.998	218.998	218.998	218.998	218.998	218.998	2,075.669	5,566.349	0.000	9,613.000
52	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	794.994	0.000	15,329.000
53	1,453.400	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	794.994	0.000	15,329.000
54	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	1,453.401	794.994	0.000	15,329.000
55	3,357.493	3,357.494	3,357.493	3,357.493	3,357.493	3,357.493	3,357.493	3,357.493	2,921.052	0.000	0.000	0.000	29,781.000
56	3,215.397	3,215.397	3,215.397	3,215.397	3,215.397	3,215.397	3,215.397	3,215.397	2,498.823	0.000	0.000	0.000	28,222.000
57	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	28,222.000	28,222.000
58	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	28,222.000	28,222.000
59	13,741.400	13,741.400	13,741.400	13,741.400	13,741.400	13,741.400	13,741.400	13,741.400	13,741.400	13,741.400	10,548.001	0.000	147,962.000
60	14,068.576	14,068.576	14,068.576	14,068.576	14,068.576	14,068.576	14,068.576	14,068.576	14,068.576	14,068.576	10,466.239	0.000	151,152.000
61	2,944.585	2,944.586	2,944.586	2,944.586	2,944.586	2,944.586	2,944.586	2,944.586	2,944.586	2,788.729	0.000	0.000	29,290.000
62	4,253.290	4,253.291	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	637.095	0.000	43,170.000
63	2,617.408	2,617.411	2,617.410	2,617.410	2,617.410	2,617.410	2,617.410	2,617.410	2,617.410	2,617.410	1,476.905	0.000	27,651.000
64	4,253.290	4,253.291	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	637.095	0.000	43,170.000
65	4,142.213	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,023.865	0.000	45,446.000
66	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,142.214	4,023.865	0.000	45,446.000
67	4,907.643	4,907.643	4,907.643	4,907.643	4,907.643	4,907.643	4,907.643	4,907.643	4,907.643	4,907.643	3,639.572	0.000	52,716.000
68	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	6,543.524	3,238.762	0.000	68,674.000
69	3,910.815	3,910.808	3,910.812	3,910.812	3,910.812	3,910.812	3,910.812	3,910.812	3,910.812	4,066.669	29,410.028	0.000	68,674.000
70	4,253.289	4,253.292	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	2,913.095	0.000	45,446.000
71	4,253.290	4,253.291	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	2,913.095	0.000	45,446.000
72	4,214.457	4,214.454	4,214.455	4,214.455	4,214.455	4,214.455	4,214.455	4,214.455	4,214.455	4,214.455	3,301.450	0.000	45,446.000
73	4,253.290	4,253.291	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	4,253.290	2,913.095	0.000	45,446.000

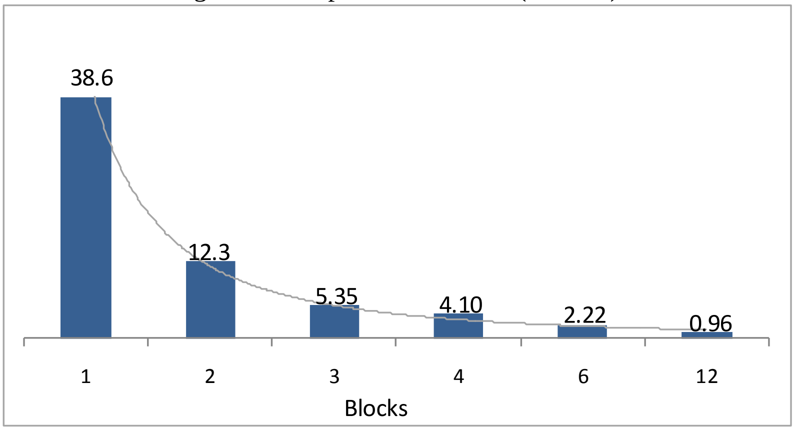
Table 7 – Results for Theoretical Time

Blocks	1	2	3	4	6	12
Time	5.04E+263	1.42E+132	2.38715E+88	3.37E+66	5.35218E+44	1.133E+23

Table 8 – Results for Real Time (seconds)

Blocks	1	2	3	4	6	12
Time	38.69	12.32	5.35	4.1	2.22	0.96

Figure 1 – Graph for Real Time (seconds)



4.2 Conclusions

The approach of Linear Programming is still one of the most important framework in industrial engineering, see [1], [2], just to name a few recent papers. And this is not only because of its simplicity but also because it provide most of the useful informations its is needed to run the Master Production Scheduling. Furthermore, the approach of linear programming provide also useful informations for understand what happens when one or more parameters varies. In general the Master Production Scheduling model is a large scale structured and sparse integer programming problem, which is hardly complex to be solved without integrality relaxation to linear programming problem. Even in modest cases, solve this integer problem is quite challenging, and we fail in most of the cases [6], when we have limited computational capacity. However in many cases we my consider this problem as being a large scale linear programming problem, and in these cases the problem can be solved using polynomial time algorithms. This is the case in general when it is important to discuss the issues related to prices. As soon we consider to solve the problem as being a large scale linear programming problem, decomposition is in order. First, because we reduce the needs of computational storage, and second, because we reduce computational time, which is obvious in general for decomposition approach. So, for resume, the approach of decomposition presented here allow to solve large scale problems of MPS, which perform a key role on the field of industrial engineering, enlarging our capacity to solve practical problems. The table of real time performance of the decomposition using MATLAB may suggest the choice of the size of subproblems appropriated for each situation.

References

[1] Ali Kafeli, Reha Uzsoy, Yahya Fathi, Michael Kay, *Using a mathematical programming model to examine the marginal price of capacity resources*, To appear in Int. J. Production Economics.

[2] D. Fatih Irdem, N. Baris Kacar, and Reha Uzsoy, Senior Member IEEE, *An Exploratory Analysis of Two Iterative Linear Programming–Simulation Approaches for Production Planning*, IEEE Transactions on Semiconductor Manufacturing, Vol. 23, NO. 3, August 2010.

[3] Andréa Toniolo Staggemeier, Alistair R. Clark, *23rd Annual Symposium of the Brazilian Operational Research Society* (SOBRAPO), Campos do Jord˜ao, Brazil, November 2001.

[4] Raimundo J. B. de Sampaio, Guilherme E. Vieira, Fabio Favaretto, *An Approach of Mathematical Programming to the Master Production Scheduling Problem*, Technical Report 2009, PUCPR-PPGEPS, Brazil.

[5] Haoxun Chen, Peter B. Luh, *An alternative framework to Lagrangian relaxation approach for job shop scheduling*, European Journal of Operational Research 149(2003) 499512.

[6] Sydney C. K. Chu, *A mathematical programming approach towards optimized master production scheduling*, Int. J. Production Economics 38(1995) 269-279.

[7] Sydney C. K. Chu, *Optimal master production scheduling in a flexible manufacturing system: the case of total aggregation*, Proc. of the First Conf. on the Operational Research Society of Hong Kong, pp 103-108, 1991.

[8] Hany Osman, Kudret Demirli, *A bilinear goal programming model and a modified Benders decomposition algorithm for supply chain reconfiguration and supplier selection*, Int. J. Production Economics 124 (2010) 97105.

[9] Ozgur Sastim, Serkan Koroglu, Mustafa Yuzukirmizi, Suleyman Ersoz, *Using Artificial Intelligence in Material Requirement Planning*, Proceeding of 5th International Symposium on Intelligent Manufacturing System, May 29-31, 2006 : 339-345, Sakarya University, Department of Industrial Engineering.

[10] Moacir Godinho Filho, Flavio Cesar Faria Fernandes, *Redução da instabilidade e melhoria de desempenho do sistema MRP*, Prod. v.16 n.1 So Paulo jan./abr. 2006.

[11] Flávia M. Takey, Marco A. Mesquita, *Aggregate Planning for a Large Food Manufacturer with High Seasonal Demand*, Brazilian Journal of Operations and Production Management Volume 3, Number 1, 2006, pp. 05-20.

[12] L. S. Lasdon, *Duality and Decomposition in Mathematical Programming*, IEEE Transactions on Systems Science and Cybernetics, 4(2),86-100, 1968.

3 **CONCLUSÃO**

Este trabalho apresenta uma solução para um problema de Programação Linear Inteira de Grande Porte, no caso, o Problema do Plano Mestre de Produção (MPS), utilizando um modelo de Decomposição de Programação Linear. Especificamente, a utilização destas técnicas visou obter ganho de tempo no processamento. De fato, o problema de MPS realizado a partir do modelo mostrou que as previsões foram comprovadas de maneira satisfatória, tanto pelo aspecto prático quanto pelo ganho considerável no tempo de processamento a partir da decomposição em blocos.

A partir deste trabalho fica a proposta de aperfeiçoar este modelo, que pode ser realizada utilizando programação inteira mista (PIM), e também a inclusão de variáveis decisórias, permitindo, por exemplo, considerar restrições referentes à utilização de recursos.

Este trabalho é uma contribuição importante para os estudos relacionados à Engenharia Industrial, especificamente na solução do problema de MPS de grande porte, pois a ilustração numérica apresentada se refere a um problema real, o que significa que a modelagem teórica do problema se ajusta adequadamente à realidade do cotidiano, comprovando que a abordagem é prática e eficiente.

REFERÊNCIAS

CHEN, H.; LUH, P. *An alternative framework to Lagrangian relaxation approach for job shop scheduling*, European Journal of Operational Research 149 (2003) 499512.

CHU, S.C.K. *A mathematical programming approach towards optimized master production scheduling.* Int. J. Production Economics 38 (1995) 269-279.

________. *Optimal master production scheduling in a flexible manufacturing system: the case of totral aggregatio*n, Proc. of the First Conf. on the Operational Research Society of Hong Kong, pp 103-108, 1991.

GODINHO FILHO, M.; FERNANDES, F.C.F. *Redução da instabilidade e melhoria de desempenho do sistema MRP.* Prod. v.16 n.1 São Paulo, jan./abr. 2006.

IRDEM, D.F.; KACAR, N.B.; UZSOY, R. Senior Member IEEE, *An Exploratory Analysis of Two Iterative Linear Programming–Simulation Approaches for Production Planning*, IEEE Transactions on Semiconductor Manufacturing, Vol. 23,N. 3, August 2010.

KAFELI, A.; UZSOY, R.; FATHI, Y.; KAY, M. *Using a mathematical programming model to examine the marginal price of capacity resources,* To appear in Int. J. Production Economics.

LASDON, L.S. *Duality and Decomposition in Mathematical Programming.* IEEE Transactions on Systems Science and Cybemetics, 4(2),86-100, 1968.

NAYLOR, A.W.; SELL, G.R. *Linear Operator Theory in Engineering and Science,* Spring-Verlag, New York, 1982.

OSMAN, H.; DEMIRLI, K. *A bilinear goal programming model and a modified Benders decomposition algorithm for supply chain reconfiguration and supplier selection*, Int. J.ProductionEconomics124(2010)97105.

SAMPAIO, R.J.B.; VIEIRA, G.E.; FAVARETTO, F. *An Approach of Mathematical Programming to the Master Production Scheduling Problem,* Technical Report 2009 , PUCPR-PPGEPS, Brazil.

SASTIM, O.; KOROGLU, S.; YUZUKIRMIZI, M.; ERSOZ, S. *Using Artificial Intelligence in Material Requirement Planning*, Proceeding of 5th International Symposium on Intelligent Manufacturing System, May 29-31.2006 : 339-345, Sakarya University, Department of Industrial Engineering.

STAGGEMEIER, A.T.; CLARK, A.R. *23rd Annual Symposium of the Brazilian Operational Research Society (SOBRAPO)*, Campos do Jordão, Brazil, november 2001.

TAKEY, F.M.; MESQUITA, M.A. *Aggregate Planning for a Large Food Manufacturer with High Seasonal Demand*, Brazilian Joumal of Operations and Production Management Volume 3. Number 1. 2006, p. 05-20.