



MODELAGEM MATEMÁTICA NO SEQUENCIAMENTO DE PROJETOS COM RESTRIÇÃO DE RECURSOS: UM ESTUDO DE CASO

Clarisse da Silva Vieira

DEPRO - Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
clarisse@cead.ufop.br

Gustavo José de Aguiar Gomes Costa

DEPRO - Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
gustavcosta@yahoo.com.br

Natanael Salgado Coutinho

DEPRO - Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
natanael-salgado@hotmail.com

Vinícius Bandeira

DEPRO - Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto ó MG
viniciuss_bandeira@hotmail.com

Vinícius Rossi Oliveira

DEPRO - Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
viniciusrossitzl@outlook.com

RESUMO

Diversos projetos, sejam eles industriais ou meramente de um processo manual comum, dependem de atividades que podem ocorrer de forma sequencial ou simultânea. No cenário de competitividade atual, necessitamos de trabalhar da forma mais eficiente possível e muitas vezes é necessário reduzir o tempo desses projetos através do sequenciamento otimizado dessas atividades. Este artigo tem como objetivo estudar o modelo do Problema de Sequenciamento em Projetos com Restrição de Recursos (PSPRR) que trata do estudo do menor tempo de execução das atividades de acordo com suas precedências e a limitação de recursos. Para isso, foi realizada uma revisão em trabalhos anteriores para mostrar alguns tipos de relações de precedências entre atividades e tipos de recursos existentes em um projeto. Aplicamos então o modelo estudado em um estudo de caso proposto pelos autores desse mesmo artigo. Os resultados encontrados mostram a melhora esperada, razão da importância do tema abordado.

PALAVRAS CHAVE. Sequenciamento, Projeto, Recurso.

Tópicos (AD&GP - PO na Administração e Gestão da Produção, PM - Programação Matemática, AO - Outras aplicações em PO)

ABSTRACT

Several projects, whether industrial or merely a common manual process, depend on activities that can occur sequentially or simultaneously. In today's competitive scenario, we need to work as efficiently as possible and it is often necessary to reduce the time of these projects through optimized sequencing of these activities. This article aims to study the Resource



Constrained Project Scheduling Problem (RCPSP) that deals with the study of the shortest execution time of activities according to their precedence and resource limitation. For this, a review was done in previous works to show some types of precedence relations between activities and types of resources existing in a project. We then apply the model studied in a case study proposed by the authors of this same article. The results found show the expected improvement, reason for the importance of the topic addressed.

KEYWORDS. Scheduling. Project. Resource.

Paper topics (AD&GP - PO na Administração e Gestão da Produção, PM - Programação Matemática, AO - Outras aplicações em PO)



1. Introdução

Diante da intensa competitividade das empresas e a busca incessante por melhores resultados, as organizações investem e estudam cada vez mais para reduzir o tempo de sua produção. Com isso, se uma empresa deseja estar, de algum modo, entre as melhores do mercado, ela deve ter um planejamento ótimo. O Planejamento e Controle da Produção e a Pesquisa Operacional são áreas responsáveis por coordenar e aplicar de forma eficiente os recursos produtivos, fazendo com que as decisões deixem de ser tomadas de forma intuitiva e passem a seguir critérios de melhoria racionais e computacionais. A complexidade dos problemas reais e empresariais, em contrapartida a possibilidade de redução do tempo na realização dos projetos através da aplicação de métodos eficientes de solução, são motivos do presente estudo.

Já existem uma grande variedade de modelos matemáticos que são implementados em programas computacionais capazes de ler as informações colocadas por um programador. Portanto, cabe a esse programador entender o problema real da empresa para transformá-lo na linguagem digital do programa. Diversos problemas, dentro do gerenciamento de projetos e processos, devido a sua complexidade e dificuldade de obtenção da solução, devem ser solucionados por esses meios. Dentre estes problemas, está o Problema de Sequenciamento em Projeto com Restrição de Recurso (PSPRR)

O Problema de Sequenciamento em Projeto com Restrição de Recurso (PSPRR) é um problema clássico da Literatura, composto por um conjunto de atividades que devem seguir uma precedência exigida no processo real para serem realizadas. Porém, devemos analisar que, para viabilizar essa execução, as quantidades disponíveis de recurso no projeto devem ser respeitadas, visando diminuir o tempo total de duração do projeto.

Com o intuito de explorar esse tema e aplicá-lo em um Estudo de Caso, este artigo apresenta, na seção 2, uma revisão bibliográfica sobre o Problema de Sequenciamento em Projetos com Restrição de Recursos. A seção 3 explicita o Estudo de Caso em uma etapa produtiva de uma fábrica de impressoras. A seção 4 mostra o modelo de Programação Linear utilizado. Os resultados e as discussões estão na seção 5. E por fim a seção 6 dispõe as conclusões obtidas seguidas das referências.

2. O Problema de Sequenciamento em Projeto com Restrição de Recurso

Segundo [Brucker *et al.* 1998], o Problema de Sequenciamento em Projetos com Restrição de Recursos (PSPRR) é definido, genericamente, por um conjunto V de atividades, ($i = 1, 2, \dots, n$), e um conjunto R de recursos renováveis, ($r = 1, \dots, m$). Cada tipo de recurso está disponível em uma quantidade constante de unidades (R_k). Cada atividade i do projeto necessita de uma quantidade constante do recurso (rik) para ser executada. A atividade i é executada sem ser interrompida em pi unidades de tempo. Os valores de R_k , rik e pi são valores positivos ou nulos. Algumas relações de precedência entre as atividades são definidas a priori. O objetivo é minimizar a duração total do projeto (*makespan*), determinando a data de início de cada atividade do projeto, respeitando as limitações de recursos e atendendo todas as relações de precedência definidas a priori.

O PSPRR é relevante tanto no meio acadêmico quanto no meio industrial. Devido à complexidade do problema e dificuldade na obtenção da solução ótima, muitas variações da formulação original podem ser observadas. Diversos autores, ao longo de vários anos, propõem modelos de Programação Matemática para o Problema, como por exemplo, em [Balas 1967], [Christofides *et al.* 1987], [Mingozzi *et al.* 1998], [Klein e Scholl 1999], [Brucker e Knust 2000], [Carlier e Néron 2003], [Artigues *et al.* 2003], [Deblaere *et al.* 2007], [Damay *et al.* 2007], [Bianco e Caramia 2013] e [Artigues *et al.* 2013].

[Balas 1967] introduz uma abordagem inovadora, na época, para o problema, considerando que as atividades que inicialmente poderiam ser executadas simultaneamente, não podem ser realizadas plenamente em paralelo, mesmo que não existam relações de precedência entre elas. Isso porque tais atividades compartilham o mesmo recurso. Essa abordagem foi



considerada em muitas outras formulações propostas posteriormente, como por exemplo, a apresentada por [Artigues *et al.* 2003].

A formulação proposta por [Artigues *et al.* 2003] considera ainda o PSPRR como um problema de fluxo e como uma extensão do modelo matemático clássico do problema de sequenciamento do *Job Shop*. O que também pode ser constatado em [Deblaere *et al.* 2007].

[Christofides *et al.* 1987], [Mingozi *et al.* 1998], [Klein e Scholl 1999], [Brucker e Knust 2000], assim como [Carlier e Néron 2003], modificam a formulação do problema original relaxando algumas restrições. [Klein e Scholl 1999], [Brucker e Knust 2000] e [Carlier e Néron 2003] também propõem o cálculo de limites inferiores para o PSPRR.

[Brucker e Knust 2000] considera que é possível dividir o horizonte de planejamento em sucessivos intervalos de tempos e que as atividades do projeto podem ser particionadas. Nele, as partes de atividades diferentes podem ser executadas simultaneamente. No entanto, as partes de uma mesma atividade devem ser executadas de uma maneira sequencial. Sendo assim, o problema consiste em determinar quais as partes das atividades que poderão ser executadas simultaneamente dentro de um mesmo intervalo, obedecendo às restrições de recursos e precedência e com o intuito de minimizar o tempo total de execução do projeto.

O modelo de Programação Matemática proposto por [Carlier e Néron 2003] também considera que o horizonte de planejamento pode ser dividido em sucessivos intervalos de tempo e que as atividades de um projeto podem ser particionadas. No entanto, as partes das atividades são executadas em intervalos consecutivos de tempo, não podendo mais ser executadas simultaneamente, sendo essa uma das principais diferenças em relação ao modelo de [Brucker e Knust 2000]. A quantidade total de recurso disponível, bem como os tempos de processamento e as quantidades de recurso para cada atividade, são valores definidos no início do projeto.

Outros autores, como [Damary *et al.* 2007], [Bianco e Caramia 2013] e [Artigues *et al.* 2013], propõem modelos matemáticos de Programação Linear Inteira Mista para o Problema, alterando algumas considerações e relaxando algumas restrições da definição original, como por exemplo, a duração das atividades, quantidade de recurso, tipo de recurso utilizado e relações de precedências.

Devido a grande proposição de formulações para os problemas classificados como PSPRR, em contrapartida, um número muito grande de métodos de resolução também são apresentados.

Segundo [Morillo *et al.* 2015], definido o PSPRR, a solução pode ser encontrada com a utilização da otimização clássica ou métodos heurísticos.

Os métodos exatos são agrupados em algoritmos que usam técnicas analíticas ou matemáticas, direcionando a resolução para uma solução ótima, caso ela exista. Estes métodos se baseiam em características específicas, como por exemplo, continuidade, diferenciabilidade, espaço de busca, entre outras. Os métodos exatos podem ser aplicados a uma grande variedade de problemas. No entanto, segundo [Fahmy *et al.* 2014], o tempo de análise se torna impraticável como o aumento do tamanho do problema.

Alguns desses métodos mais representativos são: Busca exaustiva (*Exhaustive Search*), Programação Linear Inteira Mista (*Mixed Integer Programming*), Dividir e Conquistar (*Divide and Conquer*), Programação Dinâmica (*Dynamic Programming*) e, Ramificação e Dimensionamento (*Branch and Bound*). [Morillo *et al.* 2015], ressaltam que o melhor dos métodos exatos é o *Branch and Bound*, que usa soluções parciais, construídas heurísticamente, e que resolve o problema de forma a garantir a otimização. Porém, essa metodologia não pode ser aplicável a problemas muito complexos.

Ainda conforme [Morillo *et al.* 2015], os métodos heurísticos, se baseiam em um processo de obtenção de uma boa solução para o problema. A maior parte das abordagens das heurísticas consiste em dois elementos: uma regra de prioridade (*Priority Rule*) para determinar a prioridade de cada atividade baseada em critérios pré-definidos e um esquema de geração de cronograma (*Schedule Generation Scheme*) para criar um cronograma usando a lista de atividades priorizadas. Alguns dos métodos heurísticos mais utilizados são: Busca Local (*Local Search*), Algoritmos Gulosos (*Greedy Algorithms*), Métodos Truncados de Derivação e



Limitação, métodos baseados em arcos disjuntivos e algoritmos baseados em sequenciamento por blocos.

Em [Herroelen e Leus 2004], um sistema matemático de programação é desenvolvido visando à diminuição das discrepâncias entre os modelos e a realidade. [Valls *et al.* 2005] adicionaram outro elemento, a técnica de *Justification* para melhorar a qualidade dos cronogramas gerados. [Fahmy *et al.* 2014] também apresentam uma nova técnica, a *Stacking Justification* (empilhamento), e sua implantação no proposto PSO (*Particle Swarm Optimization*) de Múltiplo *Justification* (ou MJPSO) com o objetivo de melhorar a qualidade das soluções dos PSPRR. Os resultados experimentais da pesquisa dos autores ilustraram que a combinação de *Stacking Justification* com o original *Double Justification* alcançou melhora considerável na qualidade das soluções. Além disso, mostram que a combinação das duas técnicas foi testada usando *Particle Swarm Optimization* e o algoritmo desenvolvido, MJPSO, que superou muitos dos algoritmos de melhores desempenhos na literatura, e alcançou os melhores resultados em respeito às aplicações da técnica de PSO para os PSPRR.

Para solucionar o impacto da complexidade do problema nas soluções dos algoritmos, outras metodologias foram desenvolvidas inspiradas na natureza e em unidades evolutivas como o Algoritmo Imune (AI). Os fundamentos do AI são baseados na resposta do sistema imunológico humano quando um antígeno desconhecido é reconhecido por anticorpos [Castro e Timmis 2002]. Um exemplo desse tipo de algoritmo é o CBIIA (*Chaos-based improved immune algorithm*) que, segundo [Wu *et al.* 2011], tem como objetivo reduzir custo computacional e melhorar a eficiência da busca pela solução sem ser restringido pela dificuldade do problema.

Outras proposições de metodologia podem ser encontradas na literatura. Como por exemplo, [Bhaskar *et al.* 2011] que aconselham um método *fuzzy* com tempos aproximados das atividades para resolver o problema de sequenciamento e [Damay *et al.* 2007] que resolvem o problema de PSPRR usando um modelo de programação linear, tomando como base as atividades serem ou não preemptivas.

3. Estudo de Caso

[Almeida 2009] elaborou uma problemática envolvendo o sequenciamento da produção em uma fábrica de impressoras. O autor buscou analisar o processo de montagem final de impressoras, o qual representa uma importante etapa da produção, pois apresentam recursos limitados, o que diferencia das demais etapas do processo produtivo. O autor ainda afirma que a etapa de montagem é o gargalo e que esta diretamente ligada à de manda final. Daí torna-se relevante uma ferramenta que auxilie as tomadas de decisões as quais permitem elaborar um planejamento novo com novas estratégias para a adequação das exigências do cenário externo.

Para um maior entendimento do problema proposto por [Almeida 2009], a Figura 1 destaca as etapas as quais foram modeladas e adaptadas ao modelo proposto por este estudo.

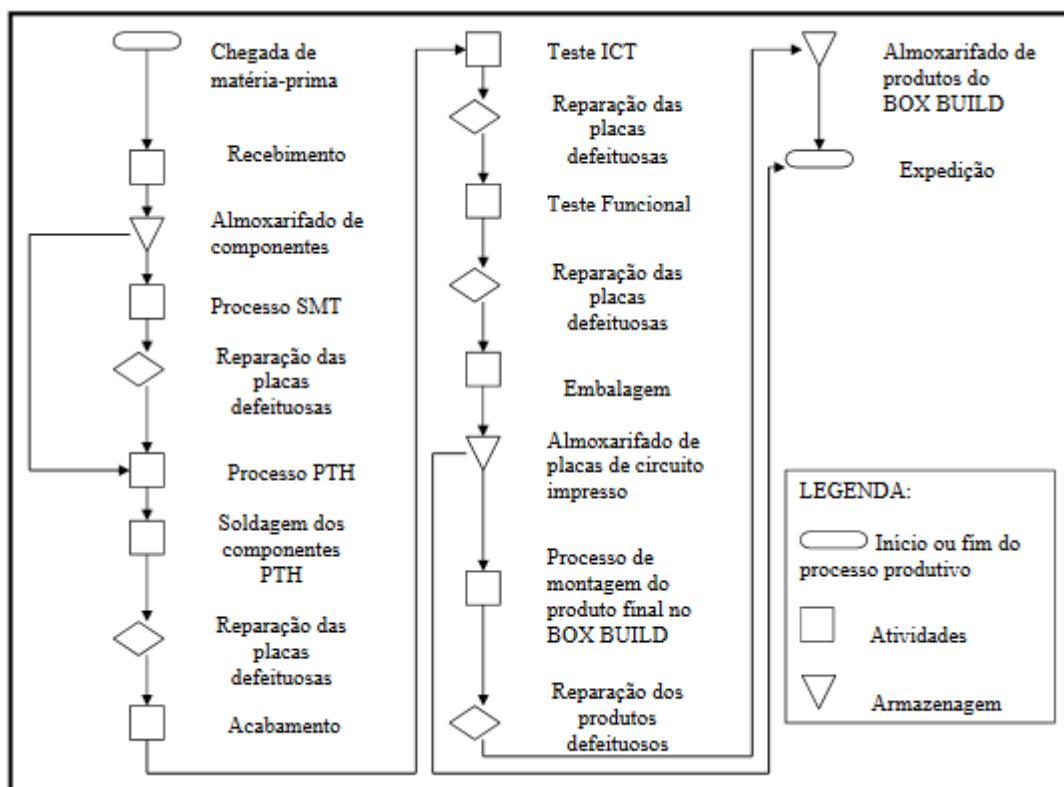


Figura 1 - Etapas do processo de montagem das impressoras
Fonte: [Almeida 2009]

De acordo com a figura é possível observar que existem diferentes caminhos a serem seguidos para realizar o processo de montagem da impressora. Como no trabalho do mesmo autor é ressaltada a existência de quatro tipos de impressoras que devem fluir dentro do processo, o *leadtime* pode ser prejudicado, uma vez que cada impressora dependerá dos recursos das mesmas atividades. Ao final do estudo, [Almeida 2009] apresenta o tempo total de produção unitário das impressoras que podem passar por duas linhas de montagem. A Tabela 1, demonstra os tempos obtidos.

Tempo unit. (horas)	Linha 1	Linha 2
Impressora 1	55	49
Impressora 2	53	46
Impressora 3	35	39
Impressora 4	45	47

Tabela 1- Tempo unitário de produção da impressora x linha.
Fonte: [Almeida 2009]

Para desenvolver o estudo, consideramos o processo de fabricação da impressora 1, fluindo nas duas linhas propostas por [Almeida 2009], já que ela apresenta o maior tempo produtivo em ambas as linhas. Logo, os dados de relevância são:

- para impressora 1 na linha 1: 55 horas;
- para impressora 1 na linha 2: 49 horas.

A Tabela 2 apresenta as relações de precedência e as durações de cada atividade do processo produtivo.



Atividade	Predecessores	Duração (horas)
1	-	1
2	1	3
3	2	4
4	3	1,5
5	4	0,5
6	5, 3	2
7	6	3
8	7	5
9	8	4
10	9	1
11	10	2
12	11	5
13	12	3
14	13	5
15	14	6
16	15	0,5
17	16	2,5
18	17	1
19	18, 15	5

Tabela 2 - Dados do problema

Portanto, conhecendo os dados que serão manipulados, o problema pode ser ilustrado através de um Diagrama. Cada atividade foi enumerada de um a dezenove. A Figura 2 representa o Diagrama mencionado.

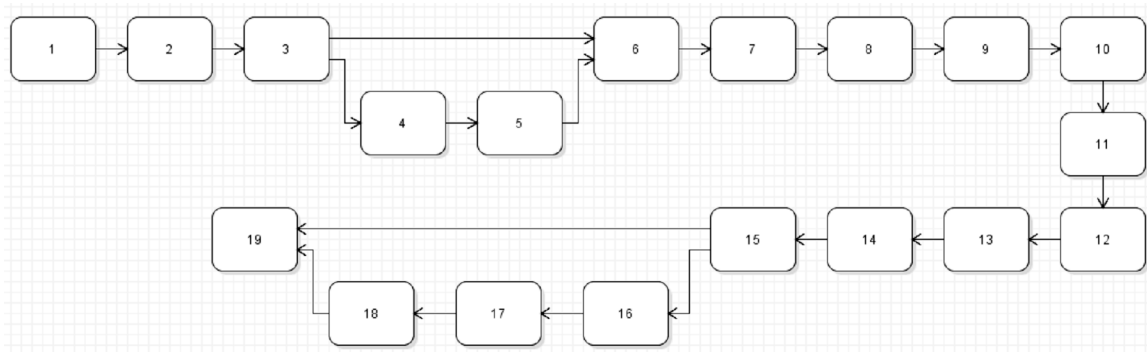


Figura 2 - Diagrama de Atividades

De acordo com a Figura 2, é possível identificar as linhas de produção da impressora. A linha 1 apresenta o maior número de atividades possível. Já a linha 2 é caracterizada pelo menor número de atividades. A Tabela 3 identifica as atividades descritas no Diagrama de Atividades.

Atividade	Nomenclatura
1	Chegada de matéria-prima
2	Recebimento
3	Almoxarifado de componentes
4	Processo SMT
5	Reparação das placas defeituosas
6	Processo PTH
7	Soldagem dos componentes PHT



8	Reparação das placas defeituosas
9	Acabamento
10	Teste ICT
11	Reparação das placas defeituosas
12	Teste funcional
13	Reparação das placas defeituosas
14	Embalagem
15	Almoxarifado de placas de circuito impresso
16	Processo de montagem do produto final no BOX BUILD
17	Reparação dos produtos defeituosos
18	Almoxarifado de produtos do BOX BUILD
19	Expedição

Tabela 3 - Nomenclatura das atividades
Fonte: [Almeida 2009]

A Tabela 4 destaca as atividades pertencentes a cada linha de produção.

Linha 1	Linha 2
Atividade 1	Atividade 1
Atividade 2	Atividade 2
Atividade 3	Atividade 3
Atividade 4	-
Atividade 5	-
Atividade 6	Atividade 6
Atividade 7	Atividade 7
Atividade 8	Atividade 8
Atividade 9	Atividade 9
Atividade 10	Atividade 10
Atividade 11	Atividade 11
Atividade 12	Atividade 12
Atividade 13	Atividade 13
Atividade 14	Atividade 14
Atividade 15	Atividade 15
Atividade 16	-
Atividade 17	-
Atividade 18	-
Atividade 19	Atividade 19

Tabela 4 - Definição das linhas de produção

4. Proposição do Problema e Metodologia

O problema consiste em definir a duração total do projeto, de tal forma que as atividades das duas linhas de produção sejam executadas ao mesmo tempo compartilhando os recursos disponíveis. Dessa forma, as atividades para a produção da impressora 1 devem ser realizadas tanto na linha 1 e 2 para que o projeto seja totalmente concluído. Para isso, é preciso definir as datas de início para cada uma das atividades considerando a execução das duas linhas de produção e as limitações referentes à disponibilidade dos recursos. Considera-se também que cada uma das 19 atividades necessitam de um tipo diferente de recurso disponível exatamente na quantidade necessária para a sua execução e que há compartilhamento de recurso entre as mesmas atividades de linhas diferentes, ou seja, todas as atividades compartilham recursos, exceto as atividades 4, 5, 16, 17 e 18 que pertencem somente a linha de produção 1.



O modelo matemático implementado foi apresentado inicialmente em [Artigues *et al.* 2003] e está reescrito a seguir:

$$\min t_{n+1} \quad (1)$$

s.a.

$$y_{ij} = 1 \quad \forall (i,j) \in E \quad (2)$$

$$y_{ji} = 0 \quad \forall (i,j) \in E \quad (3)$$

$$t_j - t_i - y_{ij} \hat{U}_i \vee p_i \times p_i - \hat{U}_i \vee p_i \quad \forall i \in V \cup \{s\}, \forall j \in V \cup \{t\}, i \tilde{N}j \quad (4)$$

$$f_{ijk} \leq y_{ij} \max(r_{ik}) \quad \forall i \in V \cup \{s\}, \forall j \in V \cup \{t\}, \forall k \in R, i \tilde{N}j \quad (5)$$

$$\sum_{j \in V \cup \{t\}} f_{ijk} = r_{ik} \quad \forall i \in V \cup \{s\}, \forall k \in R, i \tilde{N}j \quad (6)$$

$$\sum_{i \in V \cup \{s\}} f_{ijk} = r_{jk} \quad \forall j \in V \cup \{t\}, \forall k \in R, i \tilde{N}j \quad (7)$$

$$t_i \geq 0 \quad \forall i \in V \cup \{s\} \cup \{t\} \quad (8)$$

$$f_{ijk} \geq 0 \quad \forall i \in V \cup \{s\}, \forall j \in V \cup \{t\}, \forall k \in R \quad (9)$$

$$y_{ij} \in \{0,1\} \quad \forall i \in V \cup \{s\}, \forall j \in V \cup \{t\} \quad (10)$$

Onde:

- $V = \{1, 2, \dots, n\}$: conjunto das n atividades do projeto;
- $R = \{1, 2, \dots, m\}$: conjunto dos m recursos renováveis;
- E : conjunto das atividades cuja relação de precedência é conhecida *a priori*, ou seja, a atividade j só será iniciada após a conclusão da atividade i ;
- s e t : atividades fictícias que representam, respectivamente, o início e o fim do projeto;
- r_{ik} : quantidade do recurso k que a atividade i necessita para ser executada;
- p_i : tempo de processamento da atividade i ;
- f_{ijk} : fluxo do recurso k da atividade i para a atividade j ;
- y_{ij} : variável binária que estabelece se a atividade i precede j ;
- t_i : data de início da atividade i ;
- t_{n+1} : data de início da atividade fictícia $n+1$.



Neste modelo, a função objetivo (1) é minimizar t_{n+1} , pois é a data de início da última atividade (fictícia) e, minimizando essa data, conseqüentemente a duração do projeto é minimizada. As restrições (2) e (3) estão relacionadas às relações de precedência. Para as atividades cuja relação de precedência é conhecida *a priori*, o valor da variável $y_{ij} = 1$. Isso implica que j só será iniciada após a conclusão da atividade i , logo $y_{ji} = 0$. As restrições (4) definem as relações de sequenciamento entre as atividades que não possuem a relação de precedência definida *a priori*. As restrições (5) estabelecem a existência de fluxo entre as atividades através das variáveis f_{ijk} e x_{ij} . As restrições (6) e (7) mostram que o fluxo do recurso k , em uma atividade i , deve ser igual à quantidade requerida deste recurso pela atividade ser executada, isto é, deve chegar e sair da atividade a mesma quantidade de recurso que esta atividade necessita (lei de conservação do fluxo). As restrições (8), (9) e (10) definem o domínio das variáveis t_i , f_{ijk} e y_{ij} .

O problema apresentado foi modelado através da metodologia acima e para a implementação computacional do problema, utilizou-se a linguagem MPL (*Mathematical Programming Language*) e o otimizador GLPK 4.4 (*GNU Language Program Kit*) que é um pacote livre de otimização para problemas de Programação Linear. O computador utilizado na implementação e execução do modelo possui a seguinte configuração: sistema operacional *Windows 7 Home Premium*, processador *Intel Core i5*, 2.8 GHz e memória RAM instalada de 4 Gb. Para a solução deste problema, relativamente simples, foram gastos 67,2 segundos para se encontrar a solução, utilizando 32,3 MB de memória.

Por meio desta ação, pudemos verificar que o tempo total gasto nos procedimentos é, de 59 horas, conforme mostra a Tabela 5:

Atividade Linha 1	Data de Início	Atividade Linha 2	Data de Início
1	1	1	0
2	4	2	1
3	8	3	4
4	12	-	-
5	13,5	-	-
6	14	6	8
7	16	7	10
8	19	8	13
9	24	9	18
10	28	10	22
11	29	11	23
12	31	12	25
13	36	13	30
14	39	14	33
15	44	15	38
16	50	-	-
17	50,5	-	-
18	53	-	-
19	54	19	45

Tabela 5 - Datas de início das atividades



5. Resultados e Conclusões

Observa-se que, no citado modelo implementado para o problema apresentado, existem dezenove tipos de recursos disponíveis em uma unidade. Cada tipo de recurso está alocado para cada tipo de atividade do projeto. Dessa forma, as atividades 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 e 19 são comuns às duas Linhas (essas atividades ocorrem tanto na Linha 1 e na Linha 2) e compartilham os mesmos recursos (recursos tipo 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 e 19). Apenas as atividades 4, 5, 16, 17 e 18 não compartilham recursos (recursos tipo 4, 5, 16, 17 e 18). Pois, essas atividades fazem parte apenas da Linha 1.

A proposta de melhoria seria a inclusão de recursos para a execução das atividades que pertencem as duas Linhas. Assim, as atividades 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 e 19 que inicialmente seriam executadas sequencialmente, porque utilizam do mesmo tipo de recurso, agora poderiam ser realizadas simultaneamente, em paralelo. Após a inclusão de quantidades adicionais de recursos, o tempo total de conclusão do projeto seria agora definido pelo maior tempo da linha de produção, ou seja, seria igual à 55, tempo da Linha 1.

Conclui-se, portanto, que com a inclusão dos recursos tipo 4, 5, 16, 17 e 18, o tempo estimado pode ser melhorado. A redução no tempo gasto do início ao final do processo foi de 6,78%, o que representa uma redução significativa quando somamos a produção feita em um ano.

Entretanto, deve-se observar que a inclusão de novas quantidades de recursos traz gastos extras à produção. Assim sendo, atenta-se para a necessidade de que um cálculo dos ganhos com a redução do tempo e dos gastos com recursos a mais seja feito, de forma que se verifique se vale a pena a aquisição dos recursos, do ponto de vista econômico.

Atente-se para o fato de que outros testes foram feitos com o modelo melhorado, consistindo na inclusão de mais recursos. Porém, notou-se que nenhuma melhoria no tempo foi alcançada com a inclusão de tais recursos, já que o menor tempo total de produção é definido pelo maior tempo entre as linhas de montagem já otimizado. Dessa forma, a melhoria pôde ser verificada apenas com a inclusão de uma unidade aos recursos comuns entre as atividades que os compartilhavam. Trata-se, portanto, do melhor rearranjo possível segundo o modelo proposto.

Referências

- Almeida, J.F.F. (2009). Planejamento da produção de impressoras por dimensionamento de lotes e sequenciamento: um estudo de caso. Dissertação de Mestrado em Engenharia de Produção - Escola de Engenharia, Universidade Federal de Minas Gerais.
- Artigues, C., Michelon, P. e Reusser, S. (2003). Insertion techniques for static and dynamic resource constrained project scheduling. *European Journal of Operational Research*, 149: 249-267.
- Artigues, C. Leus, R. e Nobibon, F. (2013). Robust optimization for resource-constrained project scheduling with uncertain activity durations. *Flexible Services and Manufacturing Journal*, 25: 175-205.
- Balas, E. (1967). Project scheduling with resource constraints. *Operational Research*, 15: 915-957.
- Bhaskar, T., Pal, M. e Pal, A. (2011). A heuristic method for RCPSP with fuzzy activity times. *European Journal of Operational Research*, 208: 57-66.
- Bianco, L. e Caramia, M. (2013). A new formulation for the project scheduling problem under limited resources. *Flexible Services and Manufacturing Journal*, 25: 6-24.



- Brucker, P., Knust, S., Schoo, A. e Thiele, O. (1998). A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 107: 272-288.
- Brucker, P. e Knust, S. (2000). A linear programming and constraint propagation-based lower bound for the RCPSP. *European Journal of Operational Research*, 127: 355-362.
- Carlier, J. e Néron, E. (2003). On linear lower bounds for the resource constrained project scheduling problem. *European Journal of Operational Research*, 149: 314-324.
- Castro, L. and Timmis, J. (2002). An artificial immune network for multimodal function optimization. *Proceedings of the IEEE Congress on Evolutionary Computation*, 1: 699-674.
- Christofides, N., Alvarez-Valdes, R. e Tamarit, J. M. (1987). Project scheduling with resource constraints: a branch and bound approach. *European Journal of Operational Research*, 29: 262-273.
- Damay, J., Quilliot, A. e Sanlaville, E. (2007). Linear programming based algorithms for preemptive and non-preemptive RCPSP. *European Journal of Operational Research*, 182: 1012-1022.
- Deblaere, F., Demeulemeester, E. L., Herroelen, W. S. e Vonder, S. V. (2007). Robust resource allocation decisions in resource constrained projects. *Decision Science*, 38: 5-37.
- Fahmy, A., Hassan, T. e Bassion, H. (2014). Improving RCPSP solutions quality with stacking justification ó application with particle swarm optimization. *Expert Systems with Applications*, 41: 5870-5881.
- Herroelen, W. e Leus, R. (2004). Robust and reactive project scheduling: a review and classification of procedures. *International Journal of Production Research*, 42: 1599-1620.
- Klein, R. & Scholl, A. (1999). Computing lower bounds by destructive improvement: an application to resource-constrained project scheduling. *European Journal of Operational Research*, 112: 322-346.
- Mingozzi, A.; Maniezzo, V.; Ricciardelli, S. & Bianco, L. (1998). An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Management Science*, 44: 714-729.
- Morillo, D., Moreno, L. e Serna, F. (2015). A branch and bound hybrid algorithm with four deterministic heuristics for the resource constrained project scheduling problem (RCPSP). *DYNA*, 82.
- Valls, V., Ballestin, F. e Quintanilla, S. (2005). Justification and RCPSP: a technique that pays. *European Journal of Operational Research*, 165: 375-386.
- Wu, S., Wan, H. Shukla, S. e Li, B. (2011). Chaos-based improved immune algorithm (CBIIA) for resource-constrained project scheduling problems. *Expert Systems with Applications*, 38: 3387-3395.