

O MÉTODO DE BENDERS APLICADO AO PROBLEMA DE SEQUENCIAMENTO DE ATIVIDADES EM PROJETOS COM RESTRIÇÕES DE RECURSOS

Clarisse da Silva Vieira

Universidade Federal de Ouro Preto - CEAD – Centro de Educação Aberta e a Distância
Morro do Cruzeiro - CEP 35400-000 – Ouro Preto - MG
clarisse@cead.ufop.br

Carlos Roberto Venâncio de Carvalho

Universidade Federal de Minas Gerais - E.E. - Departamento de Engenharia de Produção
Av. Antônio Carlos, 6627, Pampulha - CEP 31270-901 – Belo Horizonte - MG
carlos@dep.ufmg.br

Ricardo Luiz Utsch de Freitas Pinto

Universidade Federal de Minas Gerais - E.E. - Departamento de Engenharia Mecânica
Av. Antônio Carlos, 6627, Pampulha - CEP 31270-901 – Belo Horizonte - MG
utsch@demec.ufmg.br

RESUMO

Este artigo propõe uma formulação de Programação Linear Inteira Mista para o Problema de Sequenciamento em Projeto com Restrição de Recurso (PSPRR). Essa formulação considera tanto o tempo de processamento de cada atividade quanto à quantidade de recurso alocada na sua execução como variáveis contínuas a serem otimizadas. Para as funções não lineares que relacionam essas duas variáveis são propostas aproximações lineares, envolvendo um número arbitrário de segmentos de reta. Além disso, o artigo apresenta algumas importantes contribuições como a incorporação de restrições de precedência práticas e o estabelecimento de uma estratégia para atribuição automática de um determinado parâmetro presente na formulação. Finalmente, o Método de Decomposição de Benders será desenvolvido e aplicado ao PSPRR. Este método será implementado computacionalmente e exemplos do problema de otimização serão resolvidos.

PALAVRAS CHAVE. Programação Linear Inteira Mista. Projeto. Método de Decomposição de Benders.

Área principal (Programação Matemática, PO na Administração & Gestão da Produção).

ABSTRACT

This paper proposes a formulation of Mixed Integer Linear Programming for Resource Constrained Project Scheduling Problem (RCPSPP). This formulation considers both the processing time of each activity and the amount of resource allocated in its implementation as continuous variables to be optimized. For nonlinear functions that relate these two variables are proposed linear approximations involving an arbitrary number of line segments. Furthermore, this paper presents some important contributions to the incorporation of precedence constraints and the establishment of a practical strategy for automatic assignment of a parameter in the formulation. Finally, the Benders Decomposition Method is developed and applied to RCPSPP. This method will be implemented computationally and optimization problem example will be resolved.

KEYWORDS. Mixed Integer Linear Programming. Project. Benders Decomposition Method.

Main area (PM, AD&GP)

1. Introdução

O Problema de Sequenciamento em Projeto com Restrição de Recurso (PSPRR), segundo Brucker *et al.* (1998), é basicamente constituído, por um conjunto de n atividades ($i = 1, 2, \dots, n$) e r recursos renováveis. Cada tipo k de recurso está disponível em uma quantidade constante em R_k unidades. A atividade i é executada em p_i unidades de tempo sem ser interrompida, ou seja, a restrição de não preempção deve ser respeitada. Cada atividade i necessita de uma quantidade constante r_{ik} de unidades do recurso k para ser executada. Os valores de R_k , p_i e r_{ik} são inteiros não negativos. As relações de precedências são definidas entre as atividades e devem ser respeitadas. O objetivo é minimizar a data de término do projeto.

Os problemas de sequenciamento classificados como PSPRR são problemas relevantes da literatura que motivam diversos autores, tais como Artigues *et al.* (2003), Balas (1967), Brucker e Knust (2000), Carlier e Néron (2003), Deblaere *et al.* (2007), Mingozzi *et al.* (1998), Christofides *et al.* (1987) e Klein e Scholl (1999). Devido, porém, à complexidade e dificuldade de resolução de tais problemas (Garey e Johnson, 1975), muitas variações da formulação original podem ser observadas. Segundo Koné *et al.* (2011), existe um grande número de formulações de Programação Linear Inteira Mista para o PSPRR.

No entanto, apesar das características particulares apresentadas para cada formulação do PSPRR, os autores inseridos neste contexto consideram, de maneira semelhante, que tanto o valor do tempo de processamento de cada atividade do projeto, bem como da quantidade de recurso utilizada na sua execução, são constantes conhecidas a priori. Segundo Bruni *et al.* (2011), muitas vezes, o projeto constitui-se considerando os valores médios para as durações das atividades e necessidades de recursos. Como constata Konstantinidis (1998), essas condições são usuais em muitos outros trabalhos e publicações. Por outro lado, segundo Poder *et al.* (2004), em muitas aplicações concretas, as quantidades de recursos consumidas e, conseqüentemente, os tempos de execução das atividades são raramente constantes. Também segundo Bruni *et al.* (2011), em contextos reais, esta simplificação pode levar a um desempenho muito ruim da execução do sequenciamento de atividades e da alocação dos recursos em projetos.

Neste trabalho, para as funções não lineares que relacionam o tempo de execução com a quantidade de recurso são propostas aproximações lineares envolvendo um número arbitrário de segmentos de retas. Além disso, serão incorporadas na formulação importantes restrições adicionais de precedência entre as atividades. A fixação de intervalos mínimos entre o término de uma atividade e a data de início de uma atividade subsequente (tempo mínimo de espera) também será admitida. Bem como o estabelecimento de uma estratégia para atribuição automática de um determinado parâmetro presente na formulação definido de forma arbitrária nas formulações anteriores. Finalmente, destaca-se a utilização do Algoritmo de Benders como método de resolução dos PSPRR e a sua implementação computacional.

Na Seção 2 do presente trabalho, apresenta-se a formulação de Programação Linear Inteira Mista do PSPRR. A Seção 3 relaciona o Método de Decomposição de Benders ao problema estudado. A Seção 4 apresenta técnicas computacionais para a solução do Problema obtido, destacando-se a aplicação do Método de Decomposição de Benders. Exemplos de aplicação com resultados são apresentados na Seção 5. Finalmente, a Seção 6 apresenta as principais conclusões sobre o trabalho.

2. Formulação do PSPRR como um Problema de Programação Linear Inteira Mista

O PSPRR é formulado como um modelo de Programação Linear Inteira Mista:

$$\text{Minimizar } t_{n+1} \quad (1)$$

s.t.

$$t_{n+1} - t_i - p_i - \delta_i \geq 0 \quad i = 1, K, n \quad (2)$$

$$p_i \geq p_{i'w'k} - a_{i'w'k} x_{ik} \quad i = 1, K, n, k = 1, K, r \quad (3)$$

$$\begin{aligned}
 (14) \quad & t_j - t_i - p_i - \delta_i \alpha_{ij} - M y_{ij} \geq -M & i = 0, 1, \dots, n; j = 1, \dots, n+1, i \neq j & (4) \\
 & f_{ijk} - l_{ijk} y_{ij} \leq 0 & i = 0, 1, \dots, n; j = 1, \dots, n+1, i \neq j, k = 1, K, r & (5) \\
 & \sum_{j=1}^{n+1} f_{0jk} = R_k & k = 1, K, r & (6) \\
 & \sum_{j=1}^{n+1} f_{ijk} - x_{ik} = \underline{q}_{ik} & i = 1, \dots, n, k = 1, K, r & (7) \\
 & \sum_{i=0}^n f_{ijk} - x_{jk} = \underline{q}_{jk} & j = 1, \dots, n, k = 1, K, r & (8) \\
 & \sum_{i=0}^n f_{in+1k} = R_k & k = 1, K, r & (9) \\
 & y_{ij} + y_{ji} \leq 1 & \forall (i, j) \in E - \{E_0 \cup E_1\} & (10) \\
 & y_{ij} + y_{ji} = 1 & \forall (i, j) \in E_0 \cup E_1 & (11) \\
 & y_{ij} = 1 & \forall (i, j) \in E_1 & (12) \\
 & y_{ij} \in \{0, 1\} & \forall (i, j) \in E & (13) \\
 & t_i \geq 0 & i = 0, 1, K, n+1 & (14) \\
 & 0 \leq x_{ik} \leq \bar{q}_{ik} - \underline{q}_{ik} & i = 1, K, n, k = 1, K, r & (15) \\
 & f_{ijk} \geq 0 & i = 0, 1, \dots, n; j = 1, \dots, n+1, i \neq j, k = 1, K, r & (16)
 \end{aligned}$$

Onde:

- t_i ($i = 1, \dots, n$): denota a data de início da i -ésima atividade;
- n : denota o número de atividades do projeto;
- 0 e $n+1$: denotam duas atividades fictícias que representam, respectivamente, o início e o término do projeto;
- t_0 : denota a data de início do projeto, correspondente a data de início de uma atividade fictícia, com índice 0 ;
- t_{n+1} : denota a data de término do projeto, correspondente a data de início de uma atividade fictícia, com índice $n+1$;
- $k=1, \dots, r$: denota os tipos de recurso do projeto;
- R_k : denota a quantidade máxima disponível do k -ésimo recurso utilizado;
- q_{ik} : denota a quantidade do k -ésimo recurso utilizada na execução da i -ésima atividade;
- \underline{q}_{ik} e \bar{q}_{ik} : denotam, respectivamente, a quantidade mínima e a quantidade máxima do k -ésimo recurso permitida que pode ser utilizada na execução da i -ésima atividade;
- p_i denota o tempo de execução da i -ésima atividade;
- $y_{ij} = \begin{cases} 1, & \text{se a } i\text{-ésima atividade precede a } j\text{-ésima atividade;} \\ 0, & \text{caso contrário;} \end{cases}$
- f_{ijk} : denota a quantidade do recurso k que será transferida da i -ésima atividade para a j -ésima atividade;
- x_{ik} : denota a quantidade do recurso k que será alocada para acelerar a execução da i -ésima atividade;
- l_{ijk} : denota a quantidade máxima do k -ésimo recurso que pode ser transferida da i -ésima atividade para a j -ésima atividade;
- $E = \{(i, j) : i = 0, 1, \dots, n, j = 1, \dots, n+1, i \neq j, \text{ tal que a atividade } i \text{ precede a atividade } j\}$: representa o conjunto de indexadores de pares de atividades;
- $E_0 = \{(i, j) : i = 0, 1, \dots, n, j = 1, \dots, n+1, i \neq j, \text{ tal que a atividade } i \text{ precede a atividade } j\}$: representa o conjunto de indexadores de pares de atividades que não podem ser executadas simultaneamente;
- $E_1 = \{(i, j) : i = 0, 1, \dots, n, j = 1, \dots, n+1, i \neq j, \text{ tal que a atividade } i \text{ precede a atividade } j\}$:

representa o conjunto de indexadores de pares de atividades cuja precedência é pré-estabelecida.

- δ_i : denota o tempo de repouso mínimo associado à atividade i , ou seja, corresponde ao tempo mínimo de espera que deve ser obedecido após o término da execução da atividade i ;
- $\alpha_{ij} = \begin{cases} 1, & \text{se } (i, j) \in E_0 \cup E_1; \\ 0 & \text{caso contrário;} \end{cases}$
- a_{iw-k}^i : denota o coeficiente de diminuição do tempo de execução da i -ésima atividade, considerando w^i números de segmentos, quando uma unidade do k -ésimo recurso é alocada;
- $M = \sum_{i=0}^n (\bar{p}_i + \delta_i)$: onde $\bar{p}_i = \max_k \bar{p}_{ik}$ representa o maior tempo de execução da i -ésima atividade entre todos os tempos máximos de execução de cada atividade, quando elas são executadas com a quantidade mínima permitida do k -ésimo recurso.

A função objetivo (1) indica a minimização da data de finalização do projeto (t_{n+1}). As restrições (2) impõem que a atividade fictícia (de índice $n+1$) preceda todas as demais atividades ($y_{in+1}=1, i = 0, 1, \dots, n$). As restrições (3) definem o tempo de execução de cada atividade do projeto como uma função linear da quantidade de recurso alocada. As restrições (4) estabelecem a precedência da atividade i em relação à atividade j . Elas serão ativadas quando $y_{ij} = 1$ e inativas quando $y_{ij} = 0$. Por sua vez as restrições (5) estabelecem limites para o fluxo de recurso entre a atividade i e a atividade j . Note que, se $y_{ij} = 0$, necessariamente deve-se ter $f_{ijk} = 0$. As restrições (6) estabelecem que o somatório dos fluxos de recurso que saem da atividade inicial é igual à quantidade total disponível desse recurso. As restrições (7) e (8) expressam que a quantidade de recurso que entra em uma atividade deve ser igual à quantidade de recurso que sai da mesma, ou seja, garante a continuidade de fluxo. As restrições (9) garantem que a quantidade total de recurso que chega à atividade final é igual à quantidade total disponível desse recurso. As restrições (10) e (11) impossibilitam que as variáveis de precedência y_{ij} e y_{ji} sejam simultaneamente iguais a um. Adicionalmente, as restrições (10) definem uma relação disjuntiva entre duas atividades e as restrições (11) definem a sequência entre duas atividades. As relações (12) garantem as precedências, estabelecidas a priori. As restrições (13) definem o domínio das variáveis y_{ij} . Finalmente, as restrições (14), (15) e (16) estabelecem, respectivamente, a positividade das datas de início das atividades, os limites mínimos e máximos para a quantidade de recurso e a positividade dos fluxos de recursos.

Note que esta formulação proposta estende-se para mais de um tipo de recurso e que o tempo de execução de cada atividade do projeto é definido como uma função linear da quantidade de recurso alocada, dado pelas restrições (3). Dessa forma, são propostas aproximações lineares envolvendo um número arbitrário de segmentos de retas para as funções que relacionam o tempo de execução com a quantidade de recurso. Note também que existirá uma restrição deste tipo para cada segmento de reta considerado na aproximação. Assim, p_i deve atender a todas as restrições (3). Dessa forma, para um dado q_{i_s} , a restrição (3) ativa será aquela correspondente ao segmento de reta que fornece o maior valor de p_i . Isso garante que o tempo de processamento da atividade i será calculado de acordo com o segmento de reta apropriado. Toda essa argumentação e o desenvolvimento para uma formulação linear aproximada do PSPRR estão demonstrados em Vieira *et al.* (2010).

Note ainda que δ_i é um parâmetro de entrada do modelo que será diferente de zero somente para aquelas atividades que necessitem de um tempo mínimo de espera após a sua execução, ou seja, entre duas atividades sequenciadas a priori, a última só poderá ser iniciada após o término completo da primeira. Por outro lado, se existir fluxo de recurso entre duas atividades, que não são sequenciadas a priori, o δ_i não deverá ser considerado. Dessa forma, o valor da constante α_{ij} será ativo para o primeiro caso, onde as atividades são sequenciais, e

inativo para o segundo caso, onde há fluxo de recurso entre elas, embora as atividades não sejam necessariamente sequenciais.

3. O Método de Decomposição de Benders e o PSPRR

A introdução no modelo da relação linear por partes entre o tempo de processamento e a quantidade de recurso alocada através de novas restrições e, conseqüentemente, de outras variáveis aumentam a dimensão do problema formulado. Além disso, problemas que apresentam uma estrutura especial, como os PSPRR, podem ser explorados com a utilização de procedimentos computacionais de resolução (Luenberger, 1989). Tais procedimentos englobam os métodos matemáticos de composição. Como já conhecido da literatura, de acordo com Minoux (1986), o princípio da decomposição fornece uma maneira eficaz de explorar essa estrutura para resolver problemas envolvendo um número de variáveis suficientemente grande e que são mais difíceis de serem resolvidos pelos métodos convencionais.

Dentre os métodos de decomposição, destaca-se aqui o Método de Decomposição de Benders. O interesse neste método deve-se, principalmente, ao fato deste ser aplicado para resolver problemas de Programação Linear Inteira Mista, como é o caso. Pois, com a utilização deste método, tais problemas são decompostos em dois, essencialmente distintos, e resolvidos separadamente. Frequentemente, torna-se interessante, ao aplicar o Método de Benders na resolução de problemas, decompô-lo em um problema com somente variáveis contínuas (subproblema) e o outro com variáveis inteiras e, eventualmente, outras variáveis contínuas (problema mestre). Isso possibilita, muitas vezes, um bom desempenho computacional na obtenção da solução de alguns problemas, como pode ser verificado no trabalho pioneiro de Geoffrion (1972).

Adicionalmente, a busca por um método alternativo que decompõe o PSPRR em dois níveis de decisão, um relacionado à alocação de recursos e outro ao sequenciamento das atividades, abre uma nova frente de pesquisa onde técnicas já desenvolvidas separadamente possam ser utilizadas.

A associação entre a modelagem e o método de resolução pode resultar em uma valiosa ferramenta para o processo de tomada de decisão. O aprimoramento e a aplicabilidade dessa ferramenta poderão ser de grande importância no tratamento de significativos PSPRR, inseridos no contexto da administração de projetos.

Dessa forma, a próxima Seção propõe a aplicação do Método de Decomposição de Benders ao PSPRR.

4. Aplicação do Método de Decomposição de Benders ao PSPRR

Considere o poliedro $X = \{(x, p, f, t)\}$, composto pelas variáveis x_{ik} , p_i , f_{ijk} e t_i , que satisfaz as restrições (2), (3), (6), (7), (8), (9), (14), (15) e (16). Observe que na definição deste poliedro não inclui as variáveis binárias y_{ij} .

Considere agora um conjunto de vetores definido como:

$$Y = \left\{ \begin{array}{ll} y_{ij} + y_{ji} \leq 1 & \forall (i, j) \in E - \{E_0 \cup E_1\} \end{array} \right. \quad (17)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (18)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (19)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (20)$$

}

Dessa forma, o PSPRR pode ser reescrito como:

$$\underset{(x, p, f, t) \in X}{\text{Minimizar}} \quad t_{n+1} \quad (21)$$

s. t.

$$(22) \quad t_j - t_i - p_i - \delta_i \alpha_{ij} - M y_{ij} \geq -M$$

$$i = 0, 1, \dots, n; j = 1, \dots, n+1, i \neq j$$

$$f_{ijk} - l_{ijk} y_{ij} \leq 0$$

$$i = 0, 1, \dots, n; j = 1, \dots, n+1, i \neq j, k = 1, K, r \quad (23)$$

$$y_{ij} \in Y \quad (24)$$

Fixando um vetor qualquer $\bar{y} \in Y$, define-se o problema:

$$Q(\bar{y}) = \underset{(x,p,f,t) \in X}{\text{Minimizar}} \quad t_{n+1} \quad (25)$$

s.t.

$$t_j - t_i - p_i - \delta_i \alpha_{ij} - M \bar{y}_{ij} \geq -M \quad i = 0, 1, \dots, n; j = 1, \dots, n+1, i \neq j \quad (26)$$

$$f_{ijk} - l_{ijk} \bar{y}_{ij} \leq 0 \quad i = 0, 1, \dots, n; j = 1, \dots, n+1, i \neq j, k = 1, K, r \quad (27)$$

Este problema de minimização, com \bar{y} fixado, é denominado Subproblema de Benders.

O Subproblema dual pode ser representado como:

$$Q(\bar{y}) = \underset{u \geq 0}{\text{Max}} \left\{ \underset{(x,p,f,t) \in X}{\text{Min}} \left\{ t_{n+1} + \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i \alpha_{ij} - M(1 - \bar{y}_{ij})] u_{1ij} + \sum_{(i,j) \in E} \sum_{k \in R} [f_{ijk} - l_{ijk} \bar{y}_{ij}] u_{2ijk} \right\} \right\} \quad (28)$$

com $u_{1ij} \geq 0$ e $u_{2ijk} \geq 0$.

Para garantir a viabilidade do PSPRR, segundo Geoffrion (1972), todo o valor de y deve satisfazer a condição:

$$\left\{ \underset{(x,p,f,t) \in X}{\text{Min}} \left\{ \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i \alpha_{ij} - M(1 - \bar{y}_{ij})] v_{1ij} + \sum_{(i,j) \in E} \sum_{k \in R} [f_{ijk} - l_{ijk} \bar{y}_{ij}] v_{2ijk} \right\} \right\} \quad (29)$$

com $v_{1ij} \geq 0$ e $v_{2ijk} \geq 0$.

Assumindo então que os valores de y devem satisfazer a condição de viabilidade e adicionando uma variável auxiliar y_0 , o problema inicial geral passa a ser equivalente ao Problema Linear Inteiro Misto denominado de Problema Mestre:

$$\underset{y \in Y}{\text{Minimizar}} \quad y_0 \quad (30)$$

s.t.

$$\left\{ \underset{(x,p,f,t) \in X}{\text{Min}} \left\{ t_{n+1} + \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i \alpha_{ij} - M(1 - \bar{y}_{ij})] u_{1ij}^c + \sum_{(i,j) \in E} \sum_{k \in R} [f_{ijk} - l_{ijk} \bar{y}_{ij}] u_{2ijk}^c \right\} \right\} \leq y_0 \quad (31)$$

$$\left\{ \underset{(x,p,f,t) \in X}{\text{Min}} \left\{ \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i \alpha_{ij} - M(1 - \bar{y}_{ij})] v_{1ij}^s + \sum_{(i,j) \in E} \sum_{k \in R} [f_{ijk} - l_{ijk} \bar{y}_{ij}] v_{2ijk}^s \right\} \right\} \leq 0 \quad (32)$$

$$y_0 \geq 0 \quad (33)$$

Para $(c = 1, \dots, C)$ e $(s = 1, \dots, S)$. Sendo C o número máximo de restrições do tipo (31) e S o número máximo de restrições do tipo (32). C e S são valores grandes e não conhecidos a priori.

As restrições (31) representam o corte I de Benders, que será gerado quando o Subproblema dual tiver solução ótima. As restrições (32) representam o corte II de Benders, que será gerado quando o Subproblema dual tiver solução ilimitada.

5. Testes Comparativos com a Implementação do Método de Decomposição de Benders

Para analisar o desempenho do algoritmo proposto no presente artigo, alguns experimentos computacionais foram realizados utilizando o processador Quad-Core Intel Xeon

E5335, com 2.0 GHz de velocidade e 8 GB de memória RAM, executando o sistema operacional Linux. Primeiramente, utilizou-se o CPLEX para resolver o problema na sua formulação original. Posteriormente, o algoritmo de Decomposição de Benders aplicado ao problema estudado foi implementado na linguagem C++, usando como interface o DevC++ e para resolver o problema o *solver* CPLEX (versão 9.1).

Os problemas testes da seção 5.1 foram propostos em Vieira (2010). Já os problemas testes da seção 5.2 foram gerados pelo Progen (kolisch e Sprecher, 1996) que é um gerador de problemas testes utilizado na literatura para o PSPRR, dentre outros.

5.1. Testes Comparativos I

Foram criados dois grupos de problemas: P01 e P02. Cada grupo é formado por um conjunto de cinco exemplos: a, b, c, d e e. Para cada exemplo existe somente um tipo de recurso; o número de atividades é igual à 12 e 17 para os exemplos dos grupos 01 e 02, respectivamente; os demais dados, como o tempo de execução e a quantidade de recurso alocada, variam de um exemplo para outro. Esses dados podem ser obtidos em Vieira (2010).

Os exemplos de cada um dos grupos, P01 e P02, são representados pelos grafos correspondentes das figuras 1 e 2:

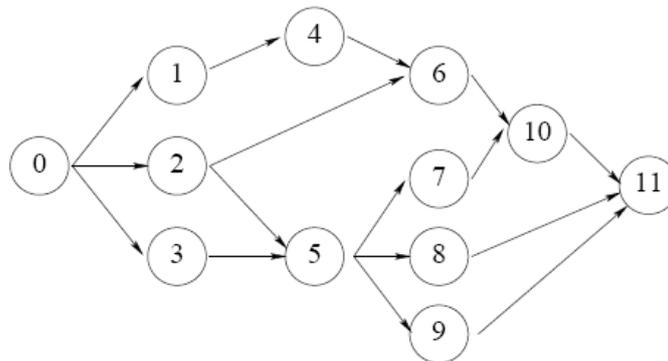


Figura 1: Grafo dos exemplos do grupo P01

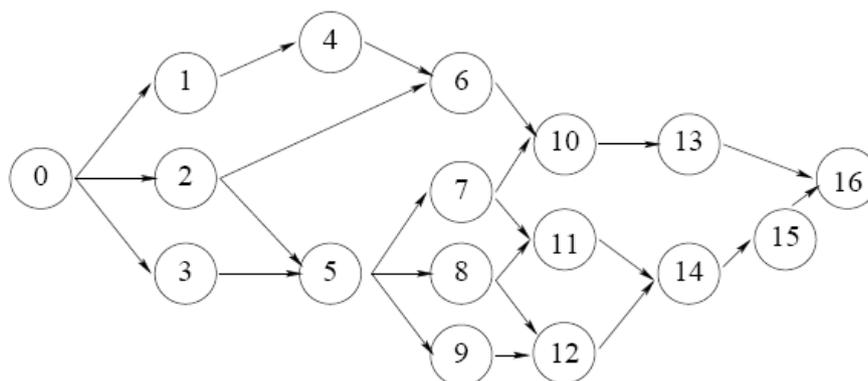


Figura 2: Grafo dos exemplos do grupo P02

O número de restrições é igual a 352 e o número de variáveis é igual a 278 para o grupo de problemas P01. Já para o grupo de problemas P02, esses números são, respectivamente, iguais a 772 e 563. Os resultados computacionais de cada grupo estão resumidos, respectivamente, nas tabelas 1 e 2, a seguir. As variáveis y_{ij} , nas tabelas apresentadas, representam a possibilidade de existir arcos que conectam as atividades que não possuem relações de precedência definidas a priori, mas utilizam de um mesmo tipo de recurso.

Problemas	Solução Ótima	Tempo CPU [s]	Num. Var. y_{ij}	Num. Iter.
P01 ^a	16	0,01	46	90
P01 ^b	21	0,02	46	114
P01 ^c	16	0,01	46	129
P01 ^d	17	0,01	46	91
P01 ^e	20	0,01	46	114

Tabela 1: Resultados computacionais do P01 - CPLEX

Problemas	Solução Ótima	Tempo CPU [s]	Num. Var. y_{ij}	Num. Iter.
P02 ^a	26	0,04	96	189
P02 ^b	26	0,03	96	118
P02 ^c	26	0,01	96	65
P02 ^d	30	0,03	96	116
P02 ^e	30	0,01	96	58

Tabela 2: Resultados computacionais do P02 - CPLEX

A seguir, as tabelas 3 e 4 apresentam outras variações dos problemas exemplos de acordo com o valor inicial das variáveis y_{ij} . Dessa forma, para y_0 , os valores das variáveis y são todos iguais a zero e para y_1 , todos iguais a um. Nessas tabelas, a segunda coluna representa o tempo total, em segundos, gastos para resolver o problema; a terceira e quarta colunas representam, respectivamente, o tempo total gasto para o subproblema e problema mestre; a quinta coluna corresponde ao número de iterações para obter a solução ótima e a última coluna corresponde ao número de iterações para comprovar que a solução obtida era a ótima.

Problemas	Tempo Total CPU [s]	Tempo Total Sub [s]	Tempo Total Mestre [s]	Num. Iter. Sol. Ótima	Num. Total Iter.
P01 ^a _{y_0}	0,89	0,13	0,76	146	148
P01 ^a _{y_1}	0,94	0,13	0,81	144	144
P01 ^b _{y_0}	0,31	0,11	0,20	92	92
P01 ^b _{y_1}	0,64	0,13	0,51	114	116
P01 ^c _{y_0}	0,03	0,01	0,02	9	9
P01 ^c _{y_1}	0,03	0,01	0,02	10	12
P01 ^d _{y_0}	0,02	0,01	0,01	12	12
P01 ^d _{y_1}	0,02	0,01	0,01	16	16
P01 ^e _{y_0}	0,03	0,01	0,02	17	17
P01 ^e _{y_1}	0,07	0,02	0,05	20	28

Tabela 3: Resultados computacionais do P01 - Benders

Problemas	Tempo Total CPU [s]	Tempo Total Sub [s]	Tempo Total Mestre [s]	Num. Iter. Sol. Ótima	Num. Total Iter.
P02 ^a _{yo}	3,58	0,47	3,11	268	268
P02 ^a _{yl}	5,76	0,58	5,18	353	535
P02 ^b _{yo}	0,32	0,11	0,21	87	124
P02 ^b _{yl}	2,39	0,15	2,24	24	145
P02 ^c _{yo}	0,28	0,08	0,20	68	88
P02 ^c _{yl}	0,42	0,12	0,32	31	69
P02 ^d _{yo}	2,52	0,10	2,42	249	258
P02 ^d _{yl}	3,48	0,08	3,40	127	214
P02 ^e _{yo}	0,32	0,08	0,24	74	105
P02 ^e _{yl}	0,28	0,08	0,20	103	202

Tabela 4: Resultados computacionais do P02 - Benders

Os resultados obtidos mostraram que em ambos os grupos o tempo computacional total gasto para resolver o problema mestre é sempre maior do que o do subproblema. Dessa forma, pode-se concluir que o tempo total para resolver o problema é determinado pelo tempo gasto para o problema mestre convergir para a mesma solução ótima já determinada pelo subproblema. Isso também pode ser comprovado, de forma geral, pelo número de iterações em que a solução ótima é obtida e pelo número de iterações que comprovam a convergência do subproblema e problema mestre para solução ótima.

5.2. Testes Comparativos II

Os problemas testes desta seção foram gerados pelo programa Progen. Dois importantes parâmetros do Progen são a complexidade da rede (*network complexity* - NC) e o fator de recursos (*Resource factor* - RF). A complexidade da rede reflete o número médio de sucessores imediatos de uma atividade. O fator de recurso reflete a densidade do recurso que uma atividade necessita para ser executada.

Os valores utilizados desses parâmetros para gerar os problemas testes foram:

- RF: 0,25, 0,5, 0,75 e 1,0
- NC: 1,5, e 1,8

Foram criados dois grupos de problemas testes: EXPL01 e EXPL02. Para cada combinação de RF e NC foi gerado um problema teste, totalizando $4 \times 2 = 8$ problemas testes para cada grupo. Para cada problema teste existe somente um tipo de recurso; o número de atividades é igual 12 e 17 para os exemplos dos grupos EXPL01 e EXPL02, respectivamente.

As tabelas 5 e 6 comparam os tempos computacionais (em segundos) do CPLEX e do Método de Decomposição de Benders para cada grupo.

Problemas	Solução Ótima	CPLEX Tempo CPU [s]	Benders Tempo CPU [s]
EXPL01(1)	16,20	0,01	0,14
EXPL01(2)	15,24	0,01	0,16
EXPL01(3)	17,43	0,01	0,16
EXPL01(4)	17,53	0,01	0,17

EXPL01(5)	18,33	0,01	0,18
EXPL01(6)	19,71	0,01	0,17
EXPL01(7)	22,19	0,01	0,19
EXPL01(8)	23,04	0,01	0,22

Tabela 5: Resultados computacionais do EXPL01

Problemas	Solução Ótima	CPLEX	Benders
		Tempo CPU [s]	Tempo CPU [s]
EXPL02(1)	34,97	0,90	1.429,7
EXPL02(2)	34,58	0,80	1.481,5
EXPL02(3)	35,62	1,08	1.637,6
EXPL02(4)	36,22	1,09	1.724,9
EXPL02(5)	37,61	1,95	1.831,4
EXPL02(6)	38,31	2,12	1.852,7
EXPL02(7)	41,97	3,15	1.912,6
EXPL01(8)	42,03	3,56	1.941,1

Tabela 6: Resultados computacionais do EXPL02

Além dos problemas apresentados, outros testes também foram realizados aplicando-se o Método de Decomposição de Benders. Dentro de um conjunto de pequeno e médio porte e sem grandes dificuldades na alocação de recursos e relações de precedência, obteve-se soluções ótimas. A partir de problemas com um número maior de atividades, o método mostrou limitações em comparação ao *solver* CPLEX (versão 9.1).

6. Conclusão

O artigo apresentou uma formulação do PSPRR que considera tanto a quantidade de recurso alocada quanto o tempo de processamento das atividades como variáveis contínuas a serem otimizadas.

Através de aproximações lineares com número arbitrário de segmentos de reta, aplicadas às relações que representam o tempo de processamento das atividades como função da quantidade de recurso, foi possível resgatar a natureza linear do PSPRR.

Contribuições importantes também foram apresentadas, como a inclusão de artifícios matemáticos que tornaram possível incorporar ao PSPRR o tempo mínimo de espera de atividades, sem que a natureza linear do problema seja quebrada.

Outra contribuição foi o estabelecimento de uma forma fechada de atribuições de valores válidos (suficientes) e não exagerados para o parâmetro M .

Além dessas, destaca-se a aplicação do Método de Decomposição de Benders ao PSPRR.

Testes computacionais foram realizados, comprovando a eficácia da técnica de aproximação desenvolvida e a importância de realizar estudos para melhorar o desempenho da aplicação do Método de Decomposição de Benders na solução do PSPRR, principalmente no que se refere às soluções de problemas de grandes dimensões. Principalmente, através da implementação de técnicas matemáticas e computacionais aprimoradas, em particular, versões melhoradas do Método de Decomposição de Benders.

Os autores do presente artigo esperam, em trabalhos futuros, apresentar estudos de técnicas que viabilizem a introdução de cortes que agilizem o procedimento de otimização,

importantes para uma eficiente implementação computacional.

Referências

- 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.
- Balas, E.** (1967), Project Scheduling with Resource Constraints, *Operational Research*, 15, 915-957.
- 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.
- 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.
- Bruni, M. E., Beraldi, P., Guerriero, F. e Pinto, E.,** (2011), A Heuristic Approach for Resource Constrained Project Scheduling with Uncertain Activity Durations, *Computers & Operations Research*, 38, 1305-1318.
- 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.
- 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.
- 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.
- Garey, M. R. e Johnson, D. S.** (1975), Complexity results for multiprocessor scheduling resource constraints, *SIAM J. Comput*, 4, 397-411.
- Geoffrion, A. M.** (1972), Generalized Benders Decomposition, *Journal of Optimization Theory and Applications*, 10, 237-260.
- Klein, R. e Scholl, A.** (1999), Computing Lower Bounds by Destructive Improvement: An Application to Resource-Constrained Project Scheduling, *European Journal of Operational Research*, 112, 322-346.
- Kolisch, R. e Sprecher, A.** (1996), PspLib – A Project Scheduling Problem Library, *European Journal of Operational Research*, 96, 205-216.
- Koné, O., Artigues, C., Lopez, P. e Mongeau, M.** (2011), Event-based MILP Models for Resource-Constrained Project Scheduling Problems, *Computers & Operations Research*, 38, 3-13.
- Konstantinidis, P. D.** (1998), A Model to Optimized Project Resource Allocation by Construction of a Balanced Histogram, *European Journal of Operational Research*, 104, 559-571.
- Luenberger, G.,** *Linear and Nonlinear Programming.* Addison-Wesley Publishing Company, 1989.
- Mingozzi, A., Maniezzo, V., Ricciardelli, S. e Bianco, L.** (1998), An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation, *Management Science*, 44, 714-729.
- Minoux, M.,** *Mathematical Programming Theory and Algorithms.* Wiley-Interscience, 1986.
- Poder, E., Beldiceanu, N., e Sanlaville, E.** (2004), Computing a Lower Approximation of the Compulsory Part of a Task with Varying Duration and Varying Resource Consumption, *European Journal of Operational Research*, 153, 239-254.
- Vieira, C. S., Pinto, R. L. U. F. e Carvalho, C. R. V.** (2010), O Problema de Sequenciamento em Projetos com Restrições de Recursos Modelado como um Problema de Programação Linear Inteira Mista. *42º Simpósio Brasileiro de Pesquisa Operacional.*
- Vieira, C. S.,** *Modelagem e Solução de Problemas de Sequenciamento de Atividades em Projetos com Restrição de Recursos.* Tese de Doutorado, UFMG, 2010.