UM MODELO DE PROGRAMAÇÃO LINEAR INTEIRA PARA O PROBLEMA DE SEQUENCIAMENTO EM PROJETOS COM RESTRIÇÃO DE RECURSOS

Clarisse da Silva Vieira
CEAD - Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
clarisse@cead.com.br

Eduardo Henrique Reis Montalvão
DEPRO – Escola de Minas – Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
montalvao.edu@gmail.com

Filipe da Rocha Martins
DEPRO – Escola de Minas – Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
filipedarocha@gmail.com

Gabriela Goes Mattioli
DEPRO – Escola de Minas – Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
gabimattioli@hotmail.com

Lucas Henrique Hanckuk
DEPRO – Escola de Minas – Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Ouro Preto - MG
lucashanchuk@yahoo.com.br

RESUMO

O sequenciamento eficiente das atividades de um projeto é um fator determinante de seu sucesso. Nessa linha, o Problema de Sequenciamento em Projetos com Restrição de Recursos (PSPRR) tem por objetivo determinar o menor tempo possível de execução das suas atividades constituintes, dados os tempos de execução das tarefas, as respectivas quantidades de recursos necessárias e o total de recursos disponíveis. Investigações anteriores sobre o tema levaram Artigues *et al.* (2003) a propor um modelo básico ao PSPRR. O objetivo deste artigo é expandir a abrangência do modelo encontrado na literatura, por meio do acréscimo das atividades disjuntivas. Além disso, realizaram-se ainda outras contribuições em relação às restrições do PSPRR com o intuito de melhorar o desempenho computacional na resolução do modelo. Os resultados obtidos se mostraram condizentes com os objetivos iniciais, indicando a relevância deste estudo face ao tema abordado.

PALAVRAS-CHAVE. Problema de Sequenciamento, Sequenciamento em Projetos, PSPRR.

Área principal (IND - PO na Indústria, PM - Programação Matemática)

ABSTRACT

The efficient scheduling of a project's activities is an important factor in it's success. In this way, the Resource Constrained Project Scheduling Problem (RCPSP) aims to determine the shortest possible time to execute its constituent activities, given the execution times of the tasks, the respective amounts of resources required and the total resources available. Previous research on the topic led Artigues *et al.* (2003) to propose a basic model to RCPSP. The aim of this paper is to expand the scope of the model found in the literature, through the addition of the disjunctive activities. Moreover, other contributions to the constraints of the RCPSP were carried out in order to improve the computational performance in solving the model. The results obtained were consistent with the original aims, indicating the relevance of this study in view of the subject.

KEYWORDS. Scheduling Problem, Projects Scheduling, RCPSP.

Main area (IND, PM)

1. Introdução

A fim de garantir uma influente consolidação no mercado, as empresas buscam por ferramentas que tragam melhoria da eficiência operacional. Nesse cenário, o sequenciamento eficiente da produção de bens ou serviços pode assegurar a otimização do tempo de processamento, implicando no cumprimento das datas estabelecidas entre fornecedor e clientes e em uma melhor utilização dos recursos existentes. Tudo isso culmina na possibilidade do aumento da capacidade produtiva e na redução de custos de produção, proporcionando maior competitividade para a empresa.

Nesse âmbito, o Problema de Sequenciamento em Projetos com Restrição de Recursos (PSPRR) é constituído, de forma geral, como definido por Brucker *et al.* (1998), por um conjunto V de atividades (1, 2, . . . , n), e um conjunto R de recursos (1, . . . , m). Cada atividade $i \in V$ necessita de uma quantidade constante r_{ik} de unidades do recurso k para ser executada. Segundo Vieira *et al.* (2010), a atividade i é executada sem ser interrompida em p_i unidades de tempo. Cada tipo de recurso $k \in R$ está disponível em uma quantidade constante de R_k unidades. Os valores de R_k, r_{ik} e p_i são maiores ou iguais a zero. Algumas relações de precedência entre as atividades são definidas a priori. Dadas tais condições, o objetivo é minimizar o makespan $C_{\max} = \max_{i \in V} S_i$, onde S_i é a data de inicio da atividade i, ou seja, minimizar o S_{n+1}.

Como os PSPRR são em geral NP-dificeis, resolvê-los, encontrando uma solução suficientemente boa e rápida, pode ser uma tarefa árdua (GAREY; JOHNSON, 1979). No entanto, tal dificuldade motiva a pesquisa desta classe de problemas por diversos autores, tais como Brucker e Knust (2000) e Carlier e Néron (2003), além de motivar a utilização de técnicas de resolução oriundas da Programação Linear.

Artigues *et al*. (2003) propõem uma formulação de Programação Linear Inteira Mista para o PSPRR como um problema de fluxo de recursos. O que também pode ser constatado em Deblaere *et al*. (2003). Segundo Goldbarg e Luna (2000), os modelos matemáticos formulados utilizando a Programação Linear Inteira Mista podem ser computacionalmente mais complexos em relação àqueles que utilizam a Programação Linear.

O presente artigo está estruturado conforme a seguir: primeiramente, na seção 2, escreve-se a formulação proposta por Artigues *et al.* (2003). Em seguida, propõem-se algumas importantes contribuições, visando um aprimoramento da formulação inicial. Com o objetivo de tornar o modelo mais próximo da realidade, acrescentaram-se as relações disjuntivas entre as atividades do projeto, ou seja, um conjunto pares de atividades que não podem ser executadas simultaneamente. A seção 3 descreve três exemplos que se referem, respectivamente, às formulações apresentadas na seção anterior. Na seção 4 os resultados são apresentados e discutidos. Por fim, conclui-se o trabalho apresentando limitações e perspectivas futuras.

2. Formulação básica do PSPRR

Sejam:

$V = \{1, 2, ..., n\}$: conjunto das n atividades do projeto;

$R = \{1, 2, ..., m\}$: 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;

M *e* N: constantes arbitrárias suficientemente grandes;

f_{ijk}: fluxo do recurso k da atividade i para a atividade j;

x_{ij}: variável binária que estabelece se a atividade i precede j;

S_i: data de início da atividade i.

Dessa forma, escreve-se o modelo como apresentado em Artigues *et al.* (2003):

$$min \quad C_{max} \tag{1}$$

$$s.a. \quad x_{ij} = 1 \qquad \forall\,(i,j) \in E \tag{2}$$

$$S_j - S_i - Mx_{ij} \geq p_i - M \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\} \tag{3}$$

$$f_{ijk} - Nx_{ij} \leq 0 \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\}, \forall\, k \in R \tag{4}$$

$$\sum_{j \in V \cup \{t\}} f_{ijk} = r_{ik} \qquad \forall\, i \in V \cup \{s\}, \forall\, k \in R \tag{5}$$

$$\sum_{i \in V \cup \{s\}} f_{ijk} = r_{jk} \qquad \forall\, j \in V \cup \{t\}, \forall\, k \in R \tag{6}$$

$$C_{max} \geq S_i + p_i \qquad \forall\, i \in V \tag{7}$$

$$f_{ijk} \in \mathbb{N} \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\}, \forall\, k \in R \tag{8}$$

$$x_{ij} \in \{0,1\} \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\} \tag{9}$$

A função objetivo em (1) deseja minimizar o tempo total de execução do projeto. As restrições (2) asseguram as relações de precedência estabelecidas *a priori*. As restrições (3) representam as relações de sequenciamento entre as atividades. O conjunto de restrições em (4) definem a possibilidade de fluxo de um tipo de recurso entre duas atividades. As restrições (5) e (6) representam a conservação da quantidade de recursos que entra e sai de cada atividade. Em (7) define-se o tempo total de execução do projeto. Por fim, (8) e (9) representam os respectivos domínios das variáveis de decisão f_{ijk} e x_{ij}.

3. O modelo proposto para o PSPRR

Para a modelagem proposta neste trabalho, considere, além das informações anteriores:

D: conjunto de atividades disjuntivas, ou seja, que não possuem uma ordem de execução definida *a priori*.

Face ao exposto, apresenta-se o seguinte o modelo:

$$\min \quad C_{max} \tag{10}$$

$$s.a. \quad x_{ij} = 1 \qquad \forall\,(i,j) \in E \tag{11}$$

$$x_{ji} = 0 \qquad \forall\,(i,j) \in E \tag{12}$$

$$x_{ij} + x_{ji} = 1 \qquad \forall\,(i,j) \in D \tag{13}$$

$$x_{ij} + x_{ji} \leq 1 \qquad \forall\,(i,j) \notin E \cup D \tag{14}$$

$$S_j - S_i - x_{ij} \sum_{i \in V} p_i \geq p_i - \sum_{i \in V} p_i \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\}, i \neq j \tag{15}$$

$$f_{ijk} - x_{ij} \max_{k \in R}\{r_{ik}\} \leq 0 \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\}, \forall\, k \in R, i \neq j \qquad (16)$$

$$\sum_{j \in V \cup \{t\}} f_{ijk} = r_{ik} \qquad \forall\, i \in V \cup \{s\}, \forall\, k \in R, i \neq j \qquad (17)$$

$$\sum_{i \in V \cup \{s\}} f_{ijk} = r_{jk} \qquad \forall\, j \in V \cup \{t\}, \forall\, k \in R, i \neq j \qquad (18)$$

$$C_{max} \geq S_i + p_i \qquad \forall\, i \in V \qquad (19)$$

$$f_{ijk} \in \mathbb{N} \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\}, \forall\, k \in R \qquad (20)$$

$$x_{ij} \in \{0,1\} \qquad \forall\, i \in V \cup \{s\}, \forall\, j \in V \cup \{t\} \qquad (21)$$

$$S_i \geq 0 \qquad \forall\, i \in V \cup \{s\} \cup \{t\} \qquad (22)$$

No modelo proposto, a função objetivo (10) e as restrições em (11) são as mesmas propostas na formulação anterior (ARTIGUES, 2003). As restrições (12) fixam os valores das variáveis x_{ji} quando a relação de precedência é preestabelecida. Em (13) são representadas as relações disjuntivas, determinando a precedência entre o par de atividades *i* e *j*. As restrições (14) estabelecem as relações de precedência aos demais pares de atividades. As restrições (15) e (16) são, respectivamente, similares ao conjunto de restrições (3) e (4) proposto por Artigues (2003). Entretanto, os valores de *M* e *N* não são arbitrários. Neste caso, *M* é calculado em função dos valores de p_i, e *N*, em função dos valores de r_{ik}. As restrições (17), (18), (19), (20) e (21) foram reescritas da formulação anterior. Além disso, as restrições (15), (16), (17) e (18) agora são válidas apenas para todo $i \neq j$. Por fim, incluíram-se as restrições em (22) que definem o domínio da variável S_i.

As restrições em (12) foram incluídas com o intuito de garantir que não haja ciclo entre duas atividades pertencentes ao conjunto de atividades precedentes. Fixando-se tais valores de x_{ij}, é possível reduzir o número de variáveis do problema em $|E|$.

Outro importante acréscimo ao modelo constitui-se nas restrições apresentadas em (13), as quais se referem às relações disjuntivas, estabelecendo que as duas atividades não possam ser executadas ao mesmo tempo. As relações determinadas por (13) aproximam o modelo a situações práticas, onde duas atividades que inicialmente poderiam ser executadas em paralelo, em termos de recursos, não o são, pois o início de uma implica em restrições à execução da outra.

As restrições (14) foram introduzidas a fim de garantir a viabilidade do problema em relação à execução dos pares de atividades que não possuem relações de precedência preestabelecidas e nem relações disjuntivas, além de melhorar o desempenho computacional do modelo.

Com o objetivo de estabelecer valores mais realistas e que podem ser automaticamente atribuídos à constante *M*, evitando-se assim possíveis equívocos, propõe-se uma nova metodologia nas restrições em (15). Como a constante *M* está relacionada aos tempos de processamento e deve ser uma constante suficientemente grande, *M* passa a ser a soma de todos os tempos de processamento, o que garante que seu valor atenderá ao requisito de grandeza e deixará de ser uma constante arbitrária. Este novo valor atribuído a *M* garante que, no caso de as atividades não apresentarem relação de precedência (ou seja, x_{ij}=0), o somatório de todos os tempos de processamento será sempre maior ou igual ao tempo de processamento de apenas uma atividade, validando assim as restrições em (15).

No conjunto de restrições (16) substituiu-se o valor arbitrário de N pelo valor máximo de um determinado recurso necessário a uma atividade. Como a restrição condiciona a existência de fluxo à existência de uma relação de precedência entre as atividades i e j, a escolha por um valor máximo dentre os possíveis fluxos entre i e j sempre atenderá a restrição.

Outra contribuição significativa ao modelo é dada nas restrições (15) a (18), as quais se limitam a valores de $i \neq j$. De fato, essa condição reduz em n o número de variáveis do problema para cada conjunto de restrições em (17) e (18), enquanto que em (15) e (16) são reduzidos, respectivamente, um total de n restrições do problema.

4. Exemplo de aplicação

Os exemplos apresentados neste artigo foram manipulados através da metodologia apresentada e resolvidos computacionalmente utilizando-se a linguagem MPL (*Mathematical Programming Language*) e o otimizador GLPK (GNU *Linear Programming Kit*).

Neste trabalho serão apresentados três exemplos. No primeiro, os dados e a modelagem seguem na forma original conforme a proposição de Artigues *et al.* (2003). Já no segundo, apresenta-se a formulação proposta, mantendo-se inalterados dados de entrada do exemplo anterior. Por fim, no terceiro exemplo, consideram-se relações disjuntivas entre alguns pares de atividades do projeto.

4.1 Exemplo 1

Considere um projeto constituído por um conjunto de 10 atividades $V = \{1, 2, ..., 10\}$, com seus respectivos tempos de processamento p_i e quantidades de recursos r_{ik} necessárias para sua execução, mostrados na Tabela 1. Neste exemplo, assumiu-se apenas um tipo de recurso renovável ($R = \{1\}$) com disponibilidade total de 4 unidades ($R_1 = 4$).

Atividade	p_i	r_{ik}	**Precedência**
1	3	2	0
2	5	3	0
3	1	3	0
4	3	1	1
5	2	1	2,3
6	4	2	4,2
7	5	3	5
8	6	1	5
9	4	1	5
10	4	1	6,7

Fonte: Artigues *et al.* (2003)

Tabela 1 - Dados de entrada do modelo

O grafo da figura 1 representa a sequência tecnológica de execução das atividades do projeto. Os nós 0 e 11 representam, respectivamente, as atividades fictícias inicial ($s = 0$) e final ($t = 11$). O tempo de processamento e a quantidade de recurso são mostrados, nesta ordem, abaixo de cada atividade. Além disso, todas as relações de precedência entre as atividades podem ser facilmente visualizadas a partir dos arcos direcionados.

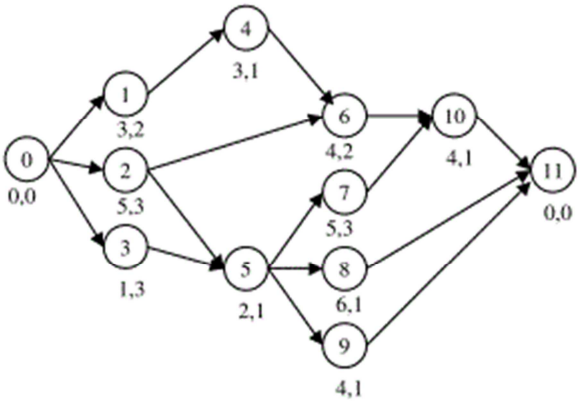

Figura 1- Grafo representativo do projeto
Fonte: Artigues *et al.* (2003)

Inicialmente, os valores de *M* e *N* serão considerados conforme Artigues *et al.* (2003), ou seja, constantes arbitrariamente grandes.

4.2 Exemplo 2

Para esse exemplo foi considerado a mesma modelagem sugerida por Artigues *et al.* (2003), porém com algumas modificações com o objetivo de reduzir o tempo computacional. A primeira modificação se dá no conjunto de restrições (15) a (18), agora válidas apenas para $i \neq j$, pois sabe-se que em índices iguais $(i = j)$ o valor das variáveis é zero, evitando-se assim que o programa calcule esses valores. A segunda contribuição são os cálculos das constantes *M* e *N*. Em Artigues *et al.* (2003), seus valores são definidos como arbitrariamente grandes. Para *M*, a ideia é defini-lo como a soma dos tempos de processamento de todas as atividades do projeto. Isto se deve ao fato de que o *M* está relacionado à restrição das sequências de execução. O valor de *N* por sua vez, é dado como o valor máximo dentre as quantidades necessárias de recursos para cada atividade, já que esta constante é um limitante superior do fluxo de recursos no projeto. Por fim, os dados de entrada para este exemplo permanecem os mesmos.

4.3 Exemplo 3

Nesta terceira situação, considera-se um conjunto formado por pares de relações disjuntivas, adicionadas ao projeto original que, conforme mencionado anteriormente, é representado por *D*. As relações disjuntivas não possuem uma ordem de precedência predeterminada, e também não podem ser realizadas ao mesmo tempo. Dessa forma, uma vez que uma atividade é escolhida para ser executada primeiramente, o seu par disjuntivo só será iniciada após sua conclusão, configurando então uma relação de precedência. A Tabela 2 define as relações disjuntivas para o exemplo:

Par	***(i,j)***
1	(6,8)
2	(7,8)

Fonte: Pesquisa direta (2011)
Tabela 2 - Pares de relações disjuntivas

É válido ressaltar também que, para este exemplo, os valores de *M* e *N* são dados, respectivamente, em função dos tempos de processamento e da quantidade máxima de determinado recurso requerida por uma atividade. Por fim, todos os demais dados relativos ao projeto permanecem inalterados e iguais aos do Exemplo 1.

5. Resultados e discussões

Nesta seção os resultados obtidos pela execução do modelo são apresentados e discutidos.

A Tabela 3 apresenta o tempo computacional e o valor do *makespan* obtidos pela execução do modelo aos três exemplos apresentados.

	Exemplo 1	**Exemplo 2**	**Exemplo 3**
Tempo computacional (s)	15,0	5,0	2,7
Makespan (u.t.)	22	22	24

Fonte: Pesquisa direta (2011)

Tabela 3 – Tempo de execução e valores do makespan para os exemplos 1, 2 e 3

A Tabela 4 mostra os fluxos dos recursos *de/para* das atividades nos exemplos 1, 2, e 3, respectivamente.

Exemplo 1		**Exemplo 2**		**Exemplo 3**	
(i, j)	**Quantidade enviada**	***(i, j)***	**Quantidade enviada**	***(i, j)***	**Quantidade enviada**
(1,2)	2	(1,3)	3	(1,2)	2
(1,3)	2	(1,4)	1	(1,3)	2
(2,3)	1	(2,5)	1	(2,3)	1
(2,5)	1	(2,8)	1	(2,5)	1
(3,4)	3	(3,4)	2	(3,4)	3
(4,6)	1	(3,8)	1	(4,6)	1
(4,7)	1	(4,2)	2	(4,7)	1
(4,8)	1	(4,6)	1	(4,8)	1
(5,7)	1	(5,10)	1	(5,7)	1
(6,9)	1	(6,8)	1	(6,10)	1
(7,8)	2	(7,12)	2	(7,8)	2
(8,11)	1	(8,7)	2	(8,9)	1
(8,12)	2	(8,11)	1	(8,11)	1
(9,10)	1	(9,12)	1	(8,12)	1
(10,12)	1	(10,9)	1	(9,12)	1
(11,12)	1	(11,12)	1	(10,12)	1
-	-	-	-	(11,12)	1

Fonte: Pesquisa direta (2011)

Tabela 4 – Quantidades de recursos enviados de *i* para *j*

É importante notar nas Tabelas 3 e 4 que as contribuições apresentadas na formulação proposta neste trabalho, além de melhorarem o desempenho computacional frente à modelagem descrita em Artigues *et al.* (2003), fazem com que o novo modelo determine o *makespan* por meio de um novo arranjo sequencial das atividades. A título de ilustração, no exemplo 1, a

quantidade de recursos que executam a atividade 3 e que "fluem" para a atividade 4 (par 3,4) é de 3 unidades, enquanto que no exemplo 2, esta quantidade é de apenas 2 unidades. Como a demanda de recursos da atividade 4 é de 3 unidades, a condição de conservação de fluxo é respeitada pelo modelo ao enviar 1 unidade de recurso da atividade 1 (início) diretamente para a atividade 4.

Uma particularidade deve ser ressaltada em termos do tempo de execução apresentado na Tabela 3. Com relação aos exemplos 1 e 2, esperava-se uma redução do tempo computacional do modelo, uma vez que o número de variáveis e restrições fora reduzido pelas contribuições propostas. Entretanto, para o exemplo 3, a expectativa era o aumento do tempo computacional devido ao acréscimo das relações disjuntivas, o que não foi verificado conforme exposto na Tabela 3. Uma vez que o modelo descrito em Artigues *et al.* (2003) não contempla as disjunções, não se pode dizer até quanto o modelo proposto melhora o tempo de execução em projetos que possuam relações disjuntivas.

Na Tabela 5 estão descritas as datas de início das atividades para os três exemplos apresentados neste trabalho:

Datas de início das atividades			
Atividade	**Exemplo 1**	**Exemplo 2**	**Exemplo 3**
1	-	-	-
2	0	6	0
3	3	0	3
4	8	5	8
5	6	9	3
6	10	6	9
7	9	14	9
8	13	9	13
9	12	16	18
10	18	12	18
11	18	18	18
12	-	-	-

Fonte: Pesquisa direta (2011)

Tabela 5 - Datas de início das atividades do exemplo

Como as atividades 1 e 12 são fictícias e possuem tempo de processamento nulo, suas respectivas datas de início são a data inicial e final de todo o projeto. É importante notar na Tabela 5 a diferença entre as soluções encontradas pelo modelo de Artigues *et al.* (2003) e o modelo proposto. As duas soluções são ótimas (*makespan = 22*), mas com um arranjo diferente em relação à ordem de execução. As figuras 2, 3, 4 mostram as mesmas informações da Tabela 5 por meio do gráfico de Gantt para os exemplos 1, 2 e 3, respectivamente.

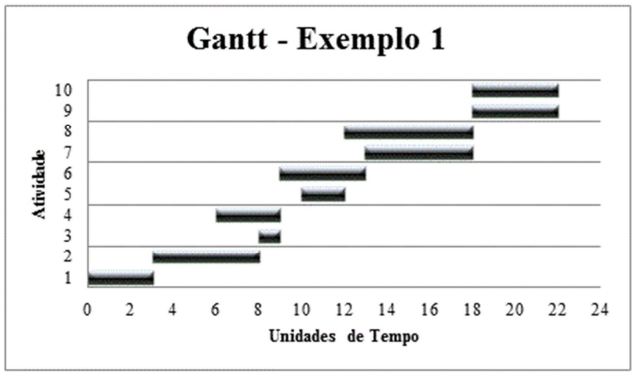

Figura 2 – Gráfico de evolução das atividades do projeto: Exemplo 1
Fonte: Pesquisa direta (2011)

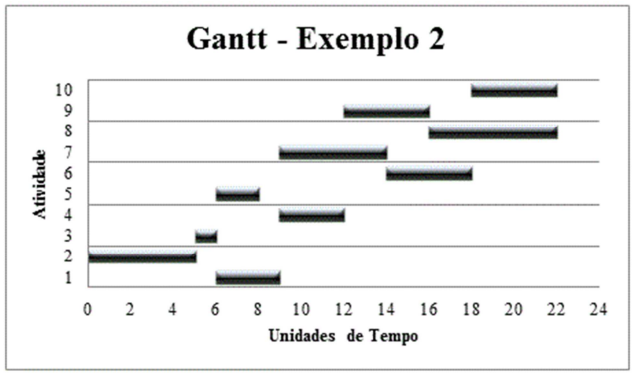

Figura 3 - Gráfico de evolução das atividades do projeto: Exemplo 2
Fonte: Pesquisa direta (2011)

As Figuras 2 e 3 explicitam a diferença entre as soluções ótimas encontradas para os exemplos 1 e 2, respectivamente.

A análise do sequenciamento das atividades do projeto por meio dos gráficos de Gantt facilita a visualização e permite um gerenciamento mais eficiente dos recursos, controlando-os ao longo de todo o processo. Além disso, é possível ainda determinar quais serão as folgas e as ociosidades dos mesmos, através do acompanhamento dos fluxos entre as atividades. Isto é particularmente interessante, pois permite, no caso de recursos renováveis como mão-de-obra, a execução simultânea de mais de um projeto ao mesmo tempo, aproveitando ao máximo a disponibilidade da força de trabalho.

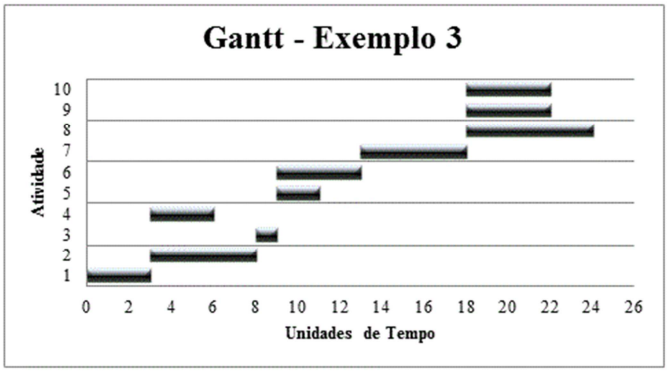

Figura 4 - Gráfico de evolução das atividades do projeto: Exemplo 3
Fonte: Pesquisa direta (2011)

Na Figura 4, a observação mais importante é a sequência de execução dos pares disjuntivos (6,8) e (7,8). Pode-se notar que a sequência seria 6, 7 e 8. Em relação aos recursos, as quantidades que cada atividade necessita são 2, 3 e 1, respectivamente.

As atividades 5 e 6 iniciam em $t = 9$ e necessitam de duas unidades de recurso cada. Entretanto, em $t = 11$, a atividade 5 é finalizada, e duas unidades de recursos ficarão ociosas até $t = 13$, momento em que a atividade 7 é iniciada com a necessidade de três unidades de recurso e assim por diante. Esta análise corrobora o que foi dito anteriormente com relação à disponibilidade dos recursos e da sua utilização, mas que não compõem o escopo deste trabalho.

Ainda na Figura 4, nota-se a piora do tempo total de execução em função do acréscimo das disjunções, bem como a execução das atividades 8, 9 e 10, todas iniciadas ao mesmo tempo e demandantes de uma unidade de recurso cada.

6. Considerações finais

Neste trabalho foi apresentada uma formulação ao Problema de Sequenciamento de Projetos com Restrição de Recursos (PSPRR) que leva em consideração alguns aspectos não abordados em estudos anteriores. Com base na formulação e nos resultados obtidos, tais considerações proporcionaram melhor desempenho computacional na resolução do PSPRR em relação ao tempo de execução do modelo e à quantidade de memória utilizada.

No que concerne aos resultados, observou-se que a formulação proposta no modelo, além de apresentar melhor desempenho em termos de tempo computacional, apresenta o valor ótimo com uma combinação diferente entre a execução das atividades, com novo fluxo dos recursos ao longo do projeto.

Uma das limitações deste trabalho consiste na aplicação do modelo a uma única instância, o que torna qualquer conclusão categórica bastante delicada. Como desenvolvimentos futuros, pretende-se estender a aplicação da formulação proposta a instâncias gradativamente maiores, com mais de um recurso e maior complexidade nas relações disjuntivas, a fim de se observar o comportamento do tempo de execução do modelo.

Outra possibilidade de desenvolvimento futuro seria uma análise mais aprofundada em termos da utilização dos recursos renováveis em projetos paralelos, visando maximizar a utilização daqueles em função de um sequenciamento prévio das atividades.

Em suma, o desenvolvimento de um método de sequenciamento de atividades que forneça uma solução ótima em tempo viável encontra grande respaldo no ambiente competitivo e dinâmico das empresas. Os ganhos advindos da aplicação de tal ferramenta no gerenciamento das atividades corporativas contribuem para a redução de desperdícios e auxiliam a tomada de decisões em âmbito estratégico.

Referências

Artigues, C., Michelon, P. e Reusser, S. *Insertion techniques for static and dynamic resource-constrained project scheduling.* European Journal of Operational Research Vol. 149, p. 249-267, 2003.

Brucker, P. e Knust, S. A Linear Programming and Constraint Propagation-Based Lower Bound for the RCPSP. European Journal of Operational Research. Vol. 127, p.355-362, 2000.

Brucker, P., Knust, S., Schoo, A. e Thiele, O. A Branch and Bound Algorithm for the Resource-Constrained Project Scheduling Problem. European Journal of Operational Research. Vol. 107, p.272-288, 1998.

Carlier, J. e Neron, E. On Linear Lower Bounds for the Resource Constrained Project Scheduling Problem. European Journal of Operational Research. Vol. 149, p.314-324, 2003.

Deblaere, F., Demeulemeester, E. L., Herroelen, W. S. e Vonder, S. V. *Robust Resource Allocation Decisions in Resource Constrained Projects.* Decision Science 38, p. 5-37, 2007.

Garey, M. R. e Johnson, D. S. Computers and Intractability: A guide to the Theory of NP Completeness, Ed. W. H. Freeman and Company, New York, 1979.

GNU Project. GLPK: GNU *linear programming kit*. http://www.gnu.org/software/glpk.

Goldbarg, M.C., Luna, H.P.L. Otimização combinatória e programação linear: modelos e algoritmos. 5. ed. Rio de Janeiro: Campus, 2000.

Vieira, C. S., Carvalho, C. R. V. e Pinto, R. L. U. F. *O Problema de Sequenciamento em Projetos com Restrições de Recursos modelado como um problema de programação linear inteira mista.* In: XLII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. Anais. Bento Gonçalves, 2010.