

Clarisse da Silva Vieira

Modelagem e Solução de Problemas
de Sequenciamento de Atividades em Projetos
com Restrição de Recursos

Tese apresentada à Escola de Engenharia da
Universidade Federal de Minas Gerais como
requisito parcial à obtenção do título
de Doutor em Engenharia Mecânica

Belo Horizonte - MG

Janeiro / 2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Clarisse da Silva Vieira

Modelagem e Solução de Problemas
de Sequenciamento de Atividades em Projetos
com Restrição de Recursos

Tese apresentada à Escola de Engenharia da
Universidade Federal de Minas Gerais como
requisito parcial à obtenção do título
de Doutor em Engenharia Mecânica

Orientador:

Prof. Dr. Ricardo Luiz Utsch de Freitas Pinto

DOUTORADO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA
ESCOLA DE ENGENHARIA
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Belo Horizonte - MG

Fevereiro / 2010

*Dedico esta tese sempre à minha irmã, Olívia,
pelo apoio e incentivo,
e à minha mãe, Gláucia,
pelo amor e carinho.*

Agradecimentos

Dedico meus sinceros agradecimentos:

à Deus;

à minha família, especialmente, à minha irmã, Olívia, e a minha mãe, Gláucia, pelo amor e por tudo o que sou;

ao meu orientador Ricardo Luiz Utsch de Freitas Pinto e ao professor Carlos Roberto Venâncio de Carvalho por todos os momentos que trabalhamos juntos, pelo incentivo e por acreditarem em mim;

aos professores participantes da banca examinadora, Henrique Pacca Loureiro Luna, Virgílio José Martins Ferreira Filho, Alexandre Salles da Cunha e Ricardo Poley Martins Ferreira, pelas sugestões e observações;

a todas as pessoas com quem convivi e compartilhei momentos importantes durante o período desta pesquisa;

à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela bolsa de estudos concedida.

*“O que for a profundidade do teu ser, assim será teu desejo.
O que for o teu desejo, assim será tua vontade.
O que for a tua vontade, assim serão teus atos.
O que forem teus atos, assim será teu destino.”*

Brihadaranyaka Upanishad IV, 4.5

Resumo

Esta tese propõe uma formulação para o Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR) que considera os tempos de processamento como funções não lineares das quantidades de recursos aplicadas e que incorpora dois tipos adicionais de restrições de precedência: a previsão de atividades que não podem ser executadas simultaneamente e a consideração da existência de tempo de espera ao final de algumas atividades. Aproximações lineares usando múltiplos segmentos de retas são propostas para reformular o problema como um Problema de Programação Linear Inteira Mista. Embora o tratamento seja desenvolvido visando recursos contínuos, ele também é aplicável a problemas com recursos discretos. Estuda-se o Método de Decomposição de Benders como ferramenta para soluções computacionais. Testes computacionais e exemplos ilustrativos modelados segundo a abordagem proposta são apresentados.

Palavras Chaves: Programação Não Linear, Programação Linear Inteira Mista, Restrição de Recurso, Sequenciamento.

Abstract

This thesis proposes a formulation for the Resource Constrained Project Scheduling Problem (RCPSP) that considers processing time varying nonlinearly with the amount of resource applied and that incorporates two additional types of precedence constraints: the prediction of activities that can not be executed simultaneously and the existence of waiting time after the end of some activities. Linear approximations using arbitrary numbers of line segments are used to reformulate the problem as a Mixed Linear Programming Problem. Although the treatment is initially aimed to continuous resources, it presents itself as an effective treatment for problems with discrete resources. Benders Decomposition Method is studied as a tool for computation solutions. Computational tests and illustrative examples modeled with the proposed approach are presented.

Keywords: Nonlinear Program, Linear Mixed Program, Resource Constrained, Scheduling.

Sumário

Lista de figuras	xiii
Lista de tabelas	xv
1 Introdução	1
1.1 Motivação do estudo	2
1.2 Objetivo	3
1.3 Contribuição científica	4
1.4 Contribuição tecnológica	4
1.5 Organização do trabalho	4
2 Contexto de estudo e fundamentos	6
2.1 Introdução	6
2.2 Administração de projetos	7
2.2.1 O Problema de alocação de recursos	9

2.2.2	O Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR)	10
2.3	Diagrama de Gantt	11
2.4	Teoria dos Grafos	12
2.4.1	Definições básicas	13
2.4.2	Representação gráfica de um grafo	13
2.4.3	Caminho crítico e atividades críticas	15
2.5	Programação Matemática	16
2.5.1	Programação Linear	17
2.5.2	Programação Linear Inteira Mista	18
2.6	Solução dos Problemas de Programação Matemática	19
2.6.1	O método Simplex	19
2.6.2	Método de Decomposição de Benders	21
2.7	Solução dos Problemas de Sequenciamento em Projeto com Restrição de Recursos	27
2.8	Comentários	29
3	O Problema de Sequenciamento em Projeto com Restrição de Recursos	30
3.1	Introdução	30
3.2	Modelos para o Problema de Sequenciamento em Projeto com Restrição de Recursos	30

3.2.1	Artigues <i>et al.</i>	31
3.2.2	Balas	33
3.2.3	Deblaere <i>et al.</i>	34
3.2.4	Brucker e Knust	35
3.2.5	Carlier e Néron	37
3.2.6	Christofides <i>et al.</i>	39
3.2.7	Oğuz e Bala	41
3.2.8	Klein e Scholl	41
3.3	Contribuições complementares	42
3.4	Conclusão	44
3.5	Contribuição da tese	44
4	Uma formulação estendida para o Problema de Sequenciamento em Projeto com Restrição de Recursos	46
4.1	Introdução	46
4.2	Formulação não linear do PSPRR com um único tipo de recurso	47
4.3	Manipulação das curvas de tempo de processamento das atividades	51
4.3.1	Aproximação linear por um segmento de reta	51
4.3.2	Aproximação linear por um número arbitrário de segmentos de reta	52
4.3.3	Aproximação linear por vários segmentos de reta aplicada ao PSPRR	55

4.4	Formulação do PSPRR como um problema de Programação Linear Inteira Mista com um único tipo recurso contínuo	58
4.5	Uma estimativa prática para o valor do parâmetro M_1	59
4.6	Uma estimativa prática para o valor do parâmetro M_2	60
4.7	Formulação do PSPRR como um problema de Programação Linear Inteira Mista com um único tipo de recurso discreto	60
4.8	Formulação não linear do PSPRR com vários tipos de recursos	62
4.9	Formulação do PSPRR como um problema de Programação Linear Inteira Mista com vários tipos de recursos contínuos	63
4.10	Formulação do PSPRR como um problema de Programação Linear Inteira Mista com vários tipos de recursos discretos	64
4.11	Conclusão	65
5	Aplicação do Método de Decomposição de Benders ao problema estudado	66
5.1	Introdução	66
5.2	O Método de Decomposição de Benders aplicado ao Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR)	68
5.3	Interpretação do Método de Benders pela Teoria dos Grafos	72
5.4	Estudo da inviabilidade do subproblema	73
5.5	Reescrevendo o Problema com uma variável de folga	75
5.5.1	Reescrevendo o Corte I	76

5.5.2	Reescrevendo o Corte II	78
5.6	Testes comparativos com a implementação do Método de Decomposição de Benders	79
5.6.1	Testes comparativos I	80
5.6.2	Testes comparativos II	86
5.7	Conclusão	88
6	Exemplos ilustrativos resolvidos computacionalmente	89
6.1	Introdução	89
6.2	Um problema com recurso contínuo	90
6.3	Um problema com recurso discreto	94
6.3.1	Um problema com resultado inteiro	94
6.3.2	Um problema com tratamento a posteriori	98
6.4	Um problema com múltiplos recursos	101
6.5	Um problema com tempo de espera	105
6.6	Comentários	112
7	Conclusão	113
7.1	Comentários finais	113
7.2	Conclusões	115
7.3	Perspectivas de continuidade do estudo	116

Lista de Figuras

2.1	Diagrama de Gantt	12
2.2	Grafo $G(N, E)$ de representação de um projeto	14
2.3	Rede orientada por eventos	15
4.1	Gráfico da função não linear $p_i(q_i)$	51
4.2	Gráfico da aproximação linear por um segmento de reta	53
4.3	Aproximação por três segmentos de reta	53
4.4	Segmentação arbitrária da curva do tempo de processamento	54
4.5	Aproximação da curva do tempo de processamento por dois segmentos	56
5.1	Grafo dos exemplos do grupo P01	80
5.2	Grafo dos exemplos do grupo P02	81
5.3	Grafo dos exemplos do grupo P03	81
6.1	Grafo do Exemplo 6.2	91
6.2	Diagrama de Gantt da solução com segmentos de reta	93

6.3	Diagrama de Gantt da solução com o dobro do número de segmentos de reta	94
6.4	Grafo do Exemplo 6.3.1	95
6.5	Diagrama de Gantt da solução pela aproximação com um segmento de reta	97
6.6	Diagrama de Gantt da solução pela aproximação com vários segmentos de reta	97
6.7	Grafo do Exemplo 6.3.2	99
6.8	Diagrama de Gantt da solução para recurso contínuo	101
6.9	Diagrama de Gantt da solução para recurso discreto	101
6.10	Grafo do Exemplo 6.4	102
6.11	Diagrama de Gantt do Exemplo 6.4	105
6.12	Grafo do Exemplo 6.5	106
6.13	Diagrama de Gantt do Exemplo 6.5 sem tempo de espera	108
6.14	Grafo do Exemplo 6.5 com tempo de espera	109
6.15	Diagrama de Gantt do Exemplo 6.5 com tempo de espera	111

Lista de Tabelas

2.1	Dados de um projeto	12
5.1	Resultados computacionais do P01 - CPLEX	81
5.2	Resultados computacionais do P02 - CPLEX	82
5.3	Resultados computacionais do P03 - CPLEX	82
5.4	Resultados computacionais do P01 - Benders	83
5.5	Resultados computacionais do P02 - Benders	84
5.6	Resultados computacionais do P03 - Benders	85
5.7	Resultados computacionais do EXPL01	87
5.8	Resultados computacionais do EXPL02	87
5.9	Resultados computacionais do EXPL03	88
6.1	Dados gerenciais do Exemplo 6.2	90
6.2	Dados técnicos do Exemplo 6.2	91
6.3	Solução do Exemplo 6.2 pela aproximação por múltiplos segmentos de reta	92

6.4	Solução do Exemplo 6.2 pela aproximação pelo dobro de segmentos de reta	93
6.5	Dados gerenciais do Exemplo 6.3.1	95
6.6	Dados técnicos do Exemplo 6.3.1	96
6.7	Solução do Exemplo 6.3.1 pela aproximação por um único segmento de reta	96
6.8	Solução do Exemplo 6.3.1 pela aproximação por múltiplos segmentos de reta	97
6.9	Dados gerenciais do Exemplo 6.3.2	98
6.10	Dados técnicos do Exemplo 6.3.2	99
6.11	Solução do Exemplo 6.3.2 pela aproximação por múltiplos segmentos de reta com recurso contínuo	100
6.12	Solução do Exemplo 6.3.2 pela aproximação por múltiplos segmentos de reta com recurso inteiro	100
6.13	Dados gerenciais do Exemplo 6.4	102
6.14	Dados técnicos do Exemplo 6.4	103
6.15	Solução do Exemplo 6.4 pela aproximação por múltiplos segmentos de reta para o recurso R_1	103
6.16	Solução do Exemplo 6.4 pela aproximação por múltiplos segmentos de reta para o recurso R_2	104
6.17	Solução do Exemplo 6.4 pela aproximação por múltiplos segmentos	104
6.18	Dados gerenciais do Exemplo 6.5	105
6.19	Dados técnicos do Exemplo 6.5	106
6.20	Alocação ótima do recurso R_1 para o Exemplo 6.5 sem tempo de espera	107

6.21	Alocação ótima do recurso R_2 para o Exemplo 6.5 sem tempo de espera . .	107
6.22	Solução ótima do Exemplo 6.5 sem tempo de espera	108
6.23	Alocação ótima do recurso R_1 para o Exemplo 6.5 com tempo de espera . .	110
6.24	Alocação ótima do recurso R_2 para o Exemplo 6.5 com tempo de espera . .	110
6.25	Solução ótima do Exemplo 6.5 com tempo de espera	111
6.26	Relação do tempo computacional com o valor de M_3	112

Capítulo 1

Introdução

A otimização do sequenciamento de atividades e da alocação de recursos é uma área da administração de projetos bastante atual e que desperta relevante interesse tanto no meio científico quanto em diversos setores industriais (Moumene e Ferland [47], Peteghem e Vanhoucke [51], Chen e Askin [12]). Uma das razões por este ser um assunto tão motivador é a possibilidade da redução dos custos de realização de projetos. Outra razão é a dificuldade na otimização de problemas reais, encontrados no âmbito empresarial, que se caracterizam por serem complexos e, conseqüentemente, difíceis de serem analisados (Oğuz e Bala [49], Garey e Johnson [25]). Esse obstáculo, ao contrário do que se imagina, é um verdadeiro estímulo para a busca de técnicas matemáticas modernas, que possam auxiliar na implementação e resolução desses tipos de problemas.

Com o advento da modelagem matemática, diferentes técnicas podem ser aplicadas em problemas práticos, conseguindo extrair do contexto real parâmetros necessários para transformá-los em problemas típicos da Pesquisa Operacional.

Dessa forma, cada vez mais, uma vasta variedade de programas matemáticos tem sido proposta para tentar solucionar os problemas de alocação de recursos e otimizar o sequenciamento das atividades, levando ao desenvolvimento de *softwares* dedicados.

Como regra geral, os problemas de alocação de recursos e sequenciamento de atividades têm sido modelados como problemas lineares de otimização envolvendo variáveis contínuas e variáveis inteiras. Por estar inserida nesse contexto, a Programação Linear Inteira Mista

é uma abordagem da Programação Matemática propícia para a resolução desta classe de problemas, sendo consistente o suficiente para gerar bons resultados práticos. Uma grande vantagem da utilização dessa técnica, segundo Pfaffenberger e Walker [52], é a capacidade de representar matematicamente as constantes e as variáveis dos problemas reais, permitindo obter soluções ótimas práticas. Segundo Valls *et al.* [61], tanto técnicas determinísticas quanto métodos heurísticos têm sido bastante utilizados para este fim.

1.1 Motivação do estudo

Muitas técnicas e métodos utilizados como ferramentas para o sequenciamento de atividades não levam em consideração as restrições operacionais associadas à alocação de recursos (Demeulemeester e Herroelen [22]). Isso é discutido mais detalhadamente, por exemplo, por Chen *et al.* [13] e Gümüsoglu e Tütek[30].

Por outro lado, mesmo as técnicas que consideram as restrições operacionais, desconsideram algumas restrições relevantes e, ao mesmo tempo, impõem algumas mais restritivas do que as que ocorrem nos problemas práticos. Por exemplo, boa parte dos trabalhos considera fixa a priori a quantidade de recurso alocada para executar uma atividade, fazendo com que o tempo para realizá-la seja fixo, como nos trabalhos de Artigues *et al.* [2], Deblaere *et al.* [21], Brucker e Knust [9] e Carlier e Néron [11]. Isso afeta a determinação do tempo ótimo de realização de todo o procedimento, ou seja, de todo o conjunto de atividades.

Outro aspecto que é abordado de forma limitada são as restrições que representam as relações de precedência entre as atividades. Tais restrições são consideradas, na maioria dos modelos atuais, em um número de possibilidades menor do que a prática exige, como pode ser visto nos trabalhos de Artigues *et al.* [2] e Deblaere *et al.*[21].

O presente estudo pretende reduzir estas limitações, expandindo o número de possibilidades para as relações de precedência e tratando o tempo de processamento de cada atividade como função da quantidade de recurso alocada, sendo ambos variáveis a serem otimizadas.

1.2 Objetivo

Esta tese tem o objetivo de estudar problemas de sequenciamento de atividades que compartilham recursos durante o processo de execução, visando obter modelos mais completos que levem em consideração alguns aspectos práticos ainda ausentes na literatura disponível. Estes aspectos devem ser introduzidos de tal forma que o problema matemático obtido continue segundo a formulação de um Problema de Programação Linear Inteira Mista.

O problema a ser analisado trata da determinação da alternativa ótima de alocação de recursos para uma sequência de atividades, respeitadas as quantidades disponíveis de cada tipo de recurso e os limites mínimos e máximos admitidos por cada atividade.

Dando prosseguimento ao estudo preliminar já realizado em Vieira [62], outro aspecto importante considerado é o tratamento das relações não lineares entre as quantidades de recurso utilizadas e os tempos de execução de cada atividade do projeto.

Além disso, são incorporadas na formulação do Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR) restrições adicionais de precedência entre as atividades. Um tipo de restrição incorporado considera a existência de atividades que não têm nenhuma relação de precedência pré estabelecida, mas que não podem ser realizadas simultaneamente. Este tipo de relação acontece, por exemplo, em atividades não correlacionadas, mas que devem ser realizadas em um mesmo local, havendo a impossibilidade da execução simultânea. O outro tipo de restrição incorporado considera a possibilidade de existir intervalos mínimos entre o final de uma atividade e o início de uma atividade seguinte. Este intervalo de tempo ocorre, por exemplo, após a aplicação de pintura. Um detalhe importante é que este tempo mínimo de espera não implica na interrupção do trabalho de uma equipe que, durante este período, pode estar executando outra atividade.

Exemplos do problema de otimização, modelados segundo a abordagem proposta, são resolvidos com a aplicação de técnicas computacionais apropriadas, desenvolvidas em parte pela autora da presente tese, cujos resultados são apresentados e discutidos.

1.3 Contribuição científica

De acordo com a pesquisa bibliográfica realizada, o presente estudo modela matematicamente um Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR) mais amplo do que os modelos atualmente disponíveis na literatura. Em primeiro lugar, diferente da grande parte dos estudos anteriores sobre a otimização do sequenciamento de atividades com recursos compartilhados, o presente estudo trata as quantidades de recursos alocadas e os tempos de execução como variáveis de otimização. Em segundo lugar, por considerar a possibilidade de atividades que não podem ser realizadas simultaneamente, mas sem nenhuma precedência estabelecida a priori. Finalmente, porque considera a existência de tempo de espera entre atividades sequenciadas a priori.

O problema, inicialmente formulado como um Problema de Programação Matemática Não Linear, é reformulado como um Problema de Programação Linear Inteira Mista, através de uma técnica de aproximação por múltiplos segmentos de reta proposta nesta tese. Isto favorece a obtenção de soluções através de técnicas computacionais bastante difundidas.

1.4 Contribuição tecnológica

Problemas práticos da indústria de transformação mecânica e de outros setores industriais poderão se beneficiar do presente estudo sobre a modelagem e otimização do sequenciamento de atividades com recursos compartilhados, levando a procedimentos mais eficientes e mais atrativos, não apenas no âmbito da indústria nacional, mas também mundial.

1.5 Organização do trabalho

Além desta introdução, a presente tese é constituída por mais seis capítulos.

No Capítulo 2, apresentam-se os conceitos fundamentais relacionados à administração de projetos e ao problema de alocação de recursos, além de técnicas utilizadas para a representação e métodos de resolução dos problemas estudados.

Descrições dos principais modelos matemáticos disponíveis na literatura são feitas no Capítulo 3.

O Capítulo 4 contém a modelagem proposta para o Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR). Todos os argumentos e desenvolvimentos matemáticos relevantes são apresentados neste Capítulo. O tratamento de recursos contínuos e de recursos discretos é considerado separadamente, em seções específicas.

O Capítulo 5 é dedicado às técnicas computacionais para a solução do Problema de Programação Linear Inteira Mista obtido, destacando-se um estudo sobre a aplicação do Método de Decomposição de Benders.

Alguns exemplos práticos de aplicação com resultados são apresentados e discutidos no Capítulo 6.

Finalmente, o trabalho é concluído no Capítulo 7, onde são apresentadas as principais conclusões e sugestões para trabalhos futuros.

Capítulo 2

Contexto de estudo e fundamentos

2.1 Introdução

Segundo Kolisch [39], o planejamento e o controle de grandes projetos são problemas complexos e importantes que fazem parte das organizações atuais. Nesse contexto, a utilização de técnicas de administração de projetos se tornam cada vez mais necessárias para lidar com tais problemas.

No final dos anos 50 e início dos anos 60, o desenvolvimento de técnicas de planejamento, tais como a Avaliação de Projeto e Técnica de Revisão - *Program Evaluation and Review Technique* (PERT) e o Método do Caminho Crítico - *Critical Path Method* (CPM), tornou possível determinar o sequenciamento das atividades para minimizar o tempo total de duração dos projetos. Porém, tais técnicas consideram que os recursos necessários para executar as atividades não são um fator de restrição (Demeulemeester e Herroelen [22]).

Na prática, entretanto, a execução dos projetos requer a utilização de vários recursos, os quais, frequentemente, estão disponíveis em quantidades limitadas e influenciam diretamente nos objetivos do planejamento, no tempo de execução e no sequenciamento das atividades. Desse modo, segundo Demeulemeester e Herroelen [22], esforços em pesquisas começaram a expandir tais ferramentas, incluindo os problemas de alocação de recursos.

Este capítulo contextualiza o problema estudado, apresentando conceitos fundamentais

inerentes à administração de projetos.

2.2 Administração de projetos

Segundo Steiner [60], um projeto é definido como uma organização de pessoas dedicadas a um propósito ou objetivo específico. Os projetos geralmente envolvem empreendimentos de grande porte, caros, únicos e de alto risco, que têm que ser finalizados dentro de um prazo definido, utilizando uma certa quantidade de recurso e atendendo a um nível desejado de desempenho.

Um projeto é composto por um conjunto de atividades simultâneas e sequenciais que visam à obtenção de um resultado específico (Williams [66]). Segundo Anthony e Govindarajan [1], os projetos podem incluir a construção de uma obra de engenharia, a produção de um determinado produto importante, o desenvolvimento e a comercialização de um novo produto, trabalhos de pesquisa, a instalação de sistemas de informações e muitos outros.

Atualmente o conceito de projeto vem sendo empregado com bastante frequência no cotidiano empresarial. Isso se deve por várias razões, principalmente no que se refere ao ponto de vista da competitividade. Pois a racionalização da utilização dos recursos e um melhor desempenho na execução das atividades de um projeto levam, conseqüentemente, a melhores resultados financeiros para as empresas.

O conceito de administração de projetos surgiu no começo da década de 60, nos Estados Unidos. De acordo com Hax e Candea [31], a administração de projetos provia a flexibilidade e a agilidade necessárias para melhorar o desempenho organizacional.

Segundo Pich *et al.* [53], administrar um projeto significa, sucintamente, planejar, executar e monitorar as atividades que o compõem. No planejamento são estabelecidos os objetivos, as atividades a serem realizadas e a seqüência de execução dessas atividades, com base nos recursos necessários e disponíveis. O monitoramento implica o acompanhamento do progresso do projeto.

As atividades são definidas, por Willis [67], como tarefas ou operações que necessitam de um certo número de unidades de tempo e/ou recursos para serem executadas. Na

prática, muitas dessas atividades requerem vários tipos de recursos, que geralmente estão em quantidade limitada.

Os recursos são todas as pessoas, matérias-primas e equipamentos primordiais para a execução dessas atividades. Segundo Patterson *et al.* [50], os recursos utilizados em um determinado projeto podem ser classificados em três categorias:

- renovável: é limitado em quantidade e pode ser reutilizado ao longo do projeto. Dentre esse tipo de recurso estão, segundo Heilmann [32], máquinas e mão-de-obra. Trabalhos considerando este tipo de recurso podem ser encontrados em Artigues *et al.* [2], Brucker e Knust [9] e Carlier e Néron [11];
- não renovável: quando a quantidade total do recurso analisado é limitada para o projeto como um todo, não havendo renovação durante o tempo. A utilização desse recurso, ao longo do projeto, implica na redução de sua disponibilidade. Como exemplo, Slowinski [59] cita matéria-prima e capital financeiro. Tais autores podem ser referência quando a utilização de recursos refere-se aos não renováveis;
- duplamente restrito: quando o recurso considerado é limitado duplamente, ou seja, na soma total e no projeto como um todo. Segundo Kolisch e Sprecher [40], o capital financeiro pode representar um recurso desta categoria se, tanto o orçamento do projeto, quanto a utilização do dinheiro ao longo do tempo, estiverem limitados. Trabalhos com recursos duplamente restritos podem ser encontrados em Weglarz [63].

O processo de planejamento de tempo em um projeto pode apresentar uma série de dificuldades que emergem durante o período de execução das atividades, como por exemplo, a escassez de recursos em determinados momentos e recursos ociosos em outros. Essas situações podem gerar prejuízos e uma adequada redistribuição desses recursos pode minimizar esses problemas. Esses fatos salientam que a administração de recursos é essencial na administração de projetos.

De acordo com Hax e Candea [31], a administração de projetos proporciona inúmeros benefícios a um empreendimento, como, por exemplo, a antecipação das situações desfavoráveis que poderão ser encontradas para que ações preventivas e corretivas possam ser tomadas antes que essas situações se consolidem. Isso prova, segundo Kerzner [37], que

a crescente utilização das técnicas de administração de projetos representa um indício do contínuo desenvolvimento das organizações.

Apesar dessas técnicas assumirem implicitamente que os recursos apropriados estarão disponíveis quando necessários, Gümüsoglu e Tütek [30] afirmam que isso pode não ocorrer na prática.

Esse fato vem impulsionando pesquisadores e administradores a desenvolverem uma considerável quantidade de trabalhos, relacionados com o problema de alocação de recursos limitados ao longo da execução do projeto.

2.2.1 O Problema de alocação de recursos

Alocar um recurso significa designá-lo para a execução de uma atividade por um determinado período de tempo.

Em um processo de alocação de recursos, podem ocorrer duas situações:

- a oferta ser maior do que a demanda, e neste caso, os recursos não são fatores limitantes na implementação do projeto;
- a demanda ser maior do que a oferta em uma ou mais unidades de tempo.

Na segunda situação, segundo Davis [18], podem ocorrer três tipos importantes de problemas:

- problema de nivelamento de recursos: neste caso, há necessidade de reduzir a variação nos perfis de demanda;
- problema de compressão de projetos: sendo necessário reduzir a duração do projeto, e para isto, ter que adicionar recursos ao menor custo;
- problema de sequenciamento em projeto com restrição de recursos: é necessário gerar uma combinação das datas de início das atividades, de modo que as quantidades de recursos disponíveis não sejam ultrapassadas em nem um período de tempo.

O problema estudado aqui é o Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR) - *Resource Constrained Project Scheduling Problem* (RCPSP) definido como segue.

2.2.2 O Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR)

De acordo Demeulemeester e Herroelen [22], um dos problemas de alocação de recursos de maior relevância é o problema de determinar a melhor sequência para a execução das atividades de maneira que os recursos sejam alocados de forma eficiente. Este problema, segundo Boctor [8], foi introduzido por Kelley [36] e Wiest [64] e é conhecido na literatura como Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR) - *Resource Constrained Project Scheduling Problem* (RCPSP). O PSPRR consiste no sequenciamento das atividades do projeto e na alocação de recursos para a execução dessas atividades, de maneira que o tempo total de duração do projeto seja minimizado. O sequenciamento das atividades deve atender às relações tecnológicas de execução, que são definidas previamente pelas restrições de precedência. Por outro lado, a alocação de recursos durante a execução das atividades deve ser feita de maneira que não exceda as quantidades disponíveis de recursos em cada intervalo de tempo e durante todo o horizonte de execução do projeto.

O PSPRR é um problema importante e desafiador para os profissionais responsáveis pela gestão dos projetos como também para os pesquisadores de áreas afins (Oğuz e Bala [49]). Uma das razões de sua importância é por este ser um problema comum a um grande número de situações reais de tomada de decisão, tais como os problemas que surgem nas distribuições de processos em redes de computadores e na administração de projetos na construção civil. Desafiador, do ponto de vista teórico, por pertencer à classe *NP-difícil* dos problemas de otimização combinatória (Garey e Johnson [25]).

As seções seguintes apresentam as principais definições do Diagrama de Gantt, da Teoria dos Grafos e da Programação Matemática. Essas três ferramentas matemáticas serão utilizadas posteriormente para representar os problemas aqui estudados.

2.3 Diagrama de Gantt

Um modo recorrente para representar graficamente uma proposição para a execução de projeto, vigentes na maioria dos *softwares*, é o Diagrama de Gantt (também conhecido como Diagrama de Barras).

O Diagrama de Gantt representa as atividades de um projeto através de linhas horizontais (ou barras ou setas) em uma sequência que corresponde à ordem em que elas serão executadas. O comprimento da linha é proporcional ao tempo de execução da atividade correspondente (Wisniewski e Klein [68]).

Dentre as vantagens do Diagrama de Gantt advém a representação gráfica da sequência e duração das atividades que são facilmente interpretadas. Essa é, provavelmente, uma das mais compreensíveis representações de sequenciamento de projetos. No entanto, as relações de precedência entre as atividades não são nitidamente identificadas.

A primeira etapa da elaboração do diagrama consiste no levantamento de todas as atividades que compõem o projeto, bem como os tempos de execução e as relações de precedência entre elas.

A tabela 2.1 exemplifica um conjunto de atividades necessárias para a realização de um projeto. Para cada atividade são dados os tempos de execução (em semanas) e as relações de precedências.

A etapa seguinte consiste em ordenar essas atividades, ou seja, estabelecer uma sequência de execução entre elas. Isso pode ser realizado através do lançamento das atividades no diagrama. A figura 2.1 mostra o diagrama do exemplo em questão.

Observa-se no Diagrama de Gantt que, se o intervalo de tempo entre as atividades com relações de precedência for nulo, todas as atividades do projeto podem ser executadas em 17 semanas.

Atividade	Tempo de Execução	Atividades Precedentes
1	2	–
2	9	1
3	4	1
4	1	3,6
5	1	1
6	2	5
7	3	2,4
8	1	7
9	2	8

Tabela 2.1: Dados de um projeto

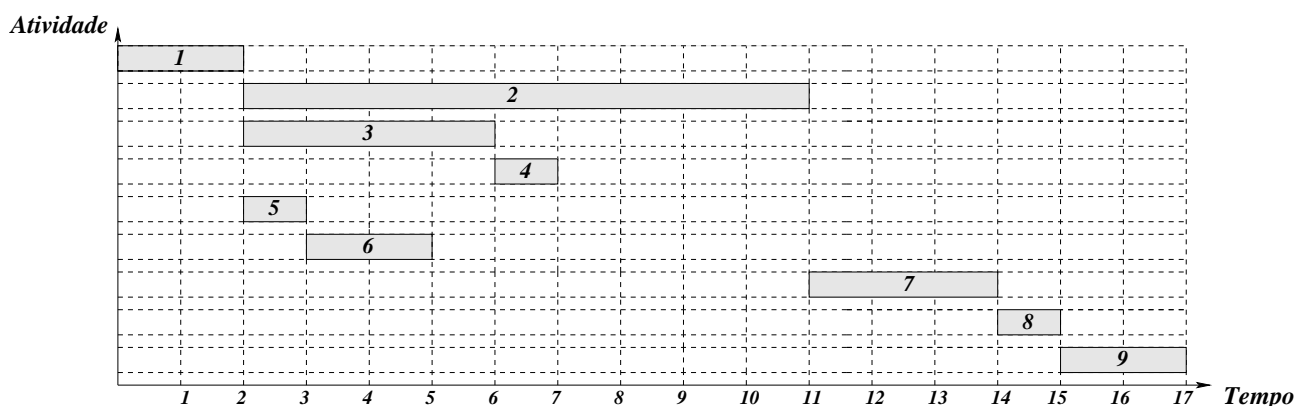


Figura 2.1: Diagrama de Gantt

2.4 Teoria dos Grafos

Segundo Gondran e Minoux [29], a linguagem dos grafos torna possível representar um determinado problema de uma maneira mais simples e objetiva. Problemas de roteirização, transporte e de administração de projetos são exemplos clássicos de aplicação do grafo. A definição de um grafo pode ser associada ao problema que se deseja representar. Existe na literatura especializada várias obras de diversos autores dedicadas à Teoria dos Grafos (TG). Cita-se, por exemplo, os livros do Boaventura Netto [7] e Gondran e Minoux [29].

Baseada nos livros citados ([7] e [29]), esta seção apresenta as definições e os conceitos necessários para a modelagem, pela Teoria dos Grafos, do problema estudado.

2.4.1 Definições básicas

Define-se aqui um grafo $G(N, E)$ como uma estrutura matemática composta por:

- N : conjunto dos n elementos discretos denominados vértices ou nós. N pode ser escrito então :

$$N = \{1, \dots, n\}$$

onde $|N| = n$ é a ordem do grafo.

- E : conjunto dos pares ordenados (i, j) denominados arcos:

$$E = \{(i, j) : i \in N, j \in N\}$$

Assim definido, o grafo $G(N, E)$ é dito orientado e, no arco (i, j) do $G(N, E)$, i é a extremidade inicial e j a extremidade final. Note aqui que $(i, j) \neq (j, i)$ e (i, j) define uma relação de precedência entre i e j , que pode ser representado por $i \prec j$.

2.4.2 Representação gráfica de um grafo

Em um grafo $G(N, E)$, os n vértices de N são representados por círculos e os arcos (i, j) de E são representados por setas, saindo de i e chegando em j .

Dessa forma, o exemplo do projeto dado pela tabela 2.1, pode ser também representado através de um grafo $G(N, E)$ mostrado pela figura 2.2.

Neste grafo, bem como nos problemas aqui considerados, as atividades são definidas pelo conjunto $N \cup \{s\} \cup \{t\}$, onde $s = 0$ e $t = n + 1$ são atividades fictícias e representam, respectivamente, o início e o término do projeto:

$$N \cup \{s\} \cup \{t\} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

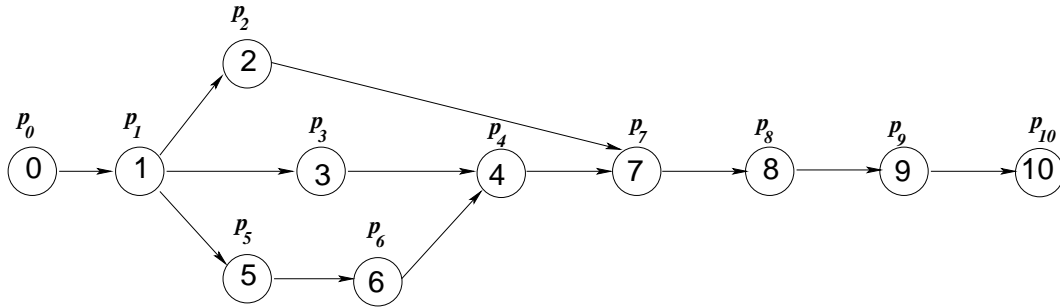


Figura 2.2: Grafo $G(N, E)$ de representação de um projeto

As relações de precedência entre as atividades do projeto são definidas pelo conjunto E de pares de atividades:

$$E = \{(0, 1), (1, 2), (1, 3), (1, 5), (2, 7), (3, 4), (4, 7), (5, 6), (6, 4), (7, 8), (8, 9), (9, 10)\}$$

Dessa forma, note que ao representar um problema por um grafo, as atividades de um determinado projeto, bem como suas respectivas relações de precedências, ficam bem definidas e podem ser facilmente visualizadas.

Segundo Goldberg e Luna [28], o grafo representado pela figura 2.2 é conhecido por rede orientada por tarefas ou rede PERT e pode ser representado por um modelo matemático como segue:

$$\text{minimizar } z = t_{n+1} - t_0 \quad (2.1)$$

$$\text{sujeito a } t_j \geq t_i + p_i \quad \forall (i, j) \in E \quad (2.2)$$

$$t_i \geq 0 \quad i = 0, 1, \dots, n + 1 \quad (2.3)$$

O problema escrito dessa forma não é um problema de fluxo em rede, mas seu dual o é. A formulação dual da rede PERT é denominada rede orientada por eventos e é exemplificada pela figura 2.3.

Nesta rede, o nó representa o início ou o término de uma atividade. No modelo primal, os valores da duração das atividades estão nos nós e no modelo dual, os valores da duração das atividades estão nos arcos.

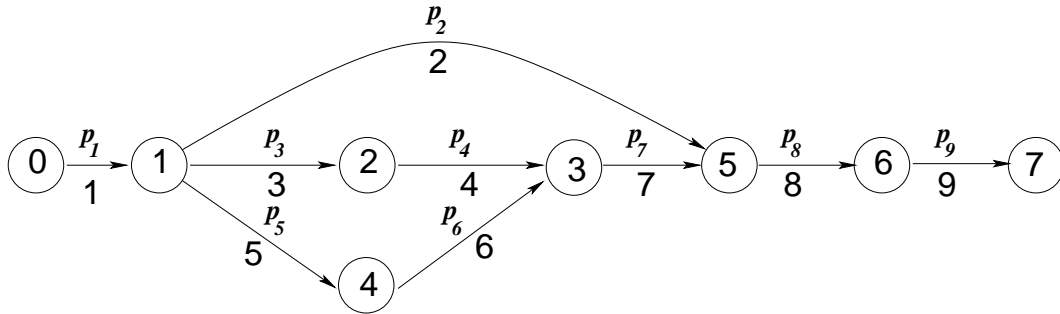


Figura 2.3: Rede orientada por eventos

Esse problema pode ser representado por um modelo matemático como segue:

$$\text{maximizar } \sum_{(i,j) \in E} p_i y_{ij} \quad (2.4)$$

$$\text{sujeito a } \sum_{(l,i) \in E} y_{li} - \sum_{(i,j) \in E} y_{ij} \leq -1 \quad i = 0 \quad (2.5)$$

$$\sum_{(l,i) \in E} y_{li} - \sum_{(i,j) \in E} y_{ij} \leq 0 \quad i = 1, \dots, n \quad (2.6)$$

$$\sum_{(l,i) \in E} y_{li} - \sum_{(i,j) \in E} y_{ij} \leq 1 \quad i = n + 1 \quad (2.7)$$

$$y_{ij} \geq 0 \quad \forall (i, j) \in E \quad (2.8)$$

Note que o número de arcos da rede orientada por eventos é igual ao número de nós da rede orientada por tarefas, excluindo os nós inicial e final. Os tempos do modelo primal correspondem ao número de nós da rede dual. Na rede orientada por eventos, o problema de determinar o maior tempo mais cedo se reduz a encontrar o caminho mais longo entre os nós extremos da rede, considerando o comprimento de cada arco (i, j) igual à duração da atividade a ele associada.

2.4.3 Caminho crítico e atividades críticas

Em um projeto, geralmente existem algumas atividades que podem ser iniciadas em determinadas datas, dentro de intervalos de tempo bem definidos, sem afetar a data de conclusão do projeto; já outras atividades devem ser iniciadas num instante específico,

pois qualquer data posterior a esse instante significaria um aumento na data de conclusão do projeto. As atividades que possuem uma data precisa para serem inicializadas são denominadas de atividades críticas.

Todo projeto, segundo Roy [57], possui ao menos uma sequência, do seu início ao seu término, de atividades críticas. Esta sequência de atividades é denominada caminho crítico.

Para a determinação do caminho crítico, calcula-se, para cada atividade, a data mais cedo e a data mais tarde para a sua execução, de maneira que a data de término do projeto seja mínima. As atividades cuja diferença entre as suas datas mais cedo e mais tarde é nula, são as atividades críticas. Assim, em outras palavras, qualquer atraso na execução de uma atividade crítica implica um atraso na conclusão do projeto como um todo (Wisniewski e Klein [68]).

Observa-se que um mesmo grafo pode possuir vários caminhos críticos.

2.5 Programação Matemática

A modelagem por Programação Matemática (PM) considera que os fatores relevantes ao problema de decisão possam ser quantificados. Esses fatores incluem as variáveis controláveis ou de decisão, as restrições sobre as variáveis e o objetivo para definir uma solução. Isso permite representá-los através de símbolos matemáticos e relacioná-los por meio de funções matemáticas.

Dentre os tipos de modelo de Programação Matemática está o modelo de otimização. Esse modelo não permite flexibilidade na escolha da alternativa, pois é estruturado para selecionar apenas uma, que será considerada ótima, segundo o critério pré-estabelecido. Esse critério integra a estrutura do modelo, que encontra a melhor solução para o problema através de uma análise matemática. Essa análise é processada por métodos sistemáticos de solução, denominados algoritmos.

Os modelos de otimização consideram que as variáveis de decisão possam assumir muitos valores ou variar dentro de grandes intervalos. A solução ótima obtida é então utilizada

como apoio ao processo de tomada de decisão.

2.5.1 Programação Linear

Os modelos de Programação Linear (PL) são um tipo especial de modelos de otimização. De acordo com Minoux [46], a Programação Linear considera os problemas de planejamento onde todas as relações são lineares tanto nas restrições como na função objetivo.

Segundo Goldberg e Luna [28], as características dos modelos de Programação Linear são:

- Proporcionalidade: a quantidade de recurso consumido por uma dada atividade deve ser proporcional ao nível dessa atividade na solução final do problema. O custo de cada atividade também é proporcional ao nível de operação da atividade;
- Não negatividade: deve ser sempre possível desenvolver dada atividade em qualquer nível não negativo e qualquer proporção de um dado recurso deve sempre poder ser utilizado;
- Aditividade: o custo total é igual a soma das parcelas associadas a cada atividade;
- Separabilidade: o custo ou o consumo de recursos, específico das operações de cada atividade, pode ser identificado de forma separada.

O modelo de Programação Linear é definido da seguinte maneira por Goldberg e Luna [28]:

$$\text{minimizar } z = \sum_{j=1}^n c_j x_j \quad (2.9)$$

$$\text{sujeito a } \sum_{j=1}^n a_{ij} x_j \geq b_i \quad i = 1, 2, \dots, p \quad (2.10)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad i = p + 1, p + 2, \dots, m \quad (2.11)$$

$$x_j \geq 0 \quad j = 1, 2, \dots, q \quad (2.12)$$

$$x_j \in \mathbb{R} \quad j = q + 1, q + 2, \dots, n \quad (2.13)$$

onde:

- z : denota o valor da medida de desempenho global;
- x_j : denota o nível de atividade j ;
- c_j : denota o incremento em z que resulta de cada incremento unitário no nível de atividade j ;
- b_i : denota a quantidade do recurso i que se encontra disponível para alocação nas atividades;
- a_{ij} : denota a quantidade do recurso i consumido por unidade da atividade j .

Segundo Hillier e Lieberman [33], a Programação Linear é bastante utilizada no tratamento de problemas de alocação de recursos limitados na execução de atividades competitivas. Consequentemente, existem muitos algoritmos ou classes de algoritmos para resolver problemas de Programação Linear. Um dos mais importantes é o método Simplex que será abordado posteriormente, neste capítulo.

2.5.2 Programação Linear Inteira Mista

O método de Programação Matemática escolhido para solucionar a complexidade dos problemas abordados neste estudo será a Programação Linear Inteira Mista (PLIM). De acordo com Goldbarg e Luna [28], esse método apresenta grande aplicabilidade e capacidade de retratar a realidade.

O problema geral de Programação Linear Inteira Mista considera que algumas variáveis estão restritas a assumirem valores inteiros e outras valores contínuos (Goldbarg e Luna [28]).

2.6 Solução dos Problemas de Programação Matemática

Segundo Minoux [46], a busca por métodos de solução eficazes para os problemas de Programação Matemática tem recebido grande atenção durante muitos anos. O desenvolvimento destes métodos, juntamente com o progresso computacional, tem contribuído para a obtenção eficiente de soluções desses problemas.

Os métodos de solução dos problemas de Programação Matemática podem ser classificados de diferentes formas, de acordo com alguns critérios. Uma das classificações desses métodos se refere a garantia ou não da obtenção de soluções ótimas para o problema. Os métodos que garantem a obtenção de soluções ótimas, na ausência de erros de arredondamento, são denominados métodos exatos. Já os métodos em que esta garantia não existe, são denominados métodos heurísticos.

Os métodos exatos, quando as variáveis são contínuas, utilizam conceitos do Cálculo Diferencial, tais como continuidade, diferenciabilidade, suavidade ou convexidade das funções. Já quando os problemas de Programação Matemática possuem apenas variáveis inteiras, os métodos exatos podem se basear em enumerações inteligentes, para as quais, a cada iteração, uma parte do domínio de definição do problema é descartada da investigação futura. Este é o caso da Programação Dinâmica e do algoritmo *Branch and Bound* (Cunha [15]).

Os métodos heurísticos se baseiam em regras de transição determinísticas e probabilísticas, sendo bastante utilizados em problemas para os quais não se dispõe de algoritmos exatos eficientes, como, por exemplo, problemas com descontinuidade nas funções envolvidas ou em suas derivadas, problemas em que as funções ou os domínios não são convexos ou em alguns problemas combinatórios (Cunha [15]).

2.6.1 O método Simplex

Um dos métodos exatos, notavelmente eficiente, que é usado rotineiramente para resolver problemas de Programação Linear é o método Simplex.

A idéia básica do método Simplex consiste em resolver repetidamente o sistema de equações lineares a fim de obter uma sequência de soluções básicas viáveis para o problema. Cada uma dessas soluções é melhor do que a sua predecessora. Esse processo se repete até que seja obtida uma solução (básica viável) ótima.

Além de encontrar a solução ótima, segundo Hillier e Lieberman [33], o método Simplex também fornece outras informações valiosas para análises adicionais do modelo. Como por exemplo, para a análise de sensibilidade e para a teoria da dualidade.

A análise de sensibilidade é importante para investigar o efeito do método Simplex sobre a solução ótima, caso os parâmetros do problema possam assumir outros valores possíveis. Dessa forma, o objetivo da análise de sensibilidade é identificar estes parâmetros, particularmente sensíveis, para que se possa analisar cuidadosamente o desempenho da solução correspondente aos valores utilizados.

A teoria da dualidade foi uma das mais importantes descobertas no início do desenvolvimento da Programação Linear. Esta descoberta revelou que todo o problema de Programação Linear tem associado a ele outro problema de Programação Linear chamado de dual. As relações entre o problema dual e o problema original, denominado primal, provam ser extremamente úteis de diversas maneiras. Uma delas é que, caso o problema primal requera mais esforço computacional do que o dual, pode-se resolver o dual no lugar do primal.

O Simplex, portanto, é um método que resolve grande parte dos problemas de Programação Linear. Entretanto, alguns destes tipos de problemas, que possuem formulações próprias, podem ser resolvidos muito mais eficientemente através de versões específicas do método Simplex, as quais exploram suas estruturas especiais. Estas versões podem reduzir de maneira significativa o tempo computacional requerido para grandes problemas e, às vezes, fazem com que problemas enormes se tornem viáveis de serem resolvidos pelo computador. Segundo Hillier e Lieberman [33], isto é particularmente verdade para problemas de transporte, problemas de alocação e sequenciamento.

2.6.2 Método de Decomposição de Benders

Segundo Belfiore *et al.* [4], os modelos de Programação Linear Inteira Mista de grande porte, devido ao grande número de variáveis inteiras, são de difícil resolução. Para contornar essa dificuldade, uma alternativa seria a implementação do Método de Decomposição de Benders.

Benders [5], em 1962, propôs um método exato de decomposição que pode ser utilizado para resolver problemas de Programação Linear Inteira Mista. Nessa abordagem, Benders define um algoritmo que divide o problema original em dois mais simples, o problema mestre e o subproblema.

O problema mestre é uma versão relaxada do problema original com um conjunto de variáveis inteiras, enquanto que o subproblema é o problema original com os valores das variáveis inteiras temporariamente fixados.

O algoritmo resolve cada um dos dois problemas, mestre e subproblema, de forma iterativa, baseando-se na aplicação dos seguintes conceitos:

- Projeção:

Considere um problema descrito como:

$$\min_{y \in Y, x \in X} f(x, y) \quad (2.14)$$

$$\text{sujeito a } g(x, y) \geq 0 \quad (2.15)$$

que envolve a otimização sobre o espaço das variáveis x e y .

A projeção sobre o espaço das variáveis y é definida como:

$$\min_{y \in Y} Q(y) \quad (2.16)$$

onde:

$$Q(y) = \min_{x \in X} f(x, y) \quad (2.17)$$

$$\text{sujeito a } g(x, y) \geq 0 \quad (2.18)$$

- Dualidade:

Segundo Hillier e Lieberman [33], define-se dualidade através do par de problemas de Programação Linear abaixo:

Problema primal:

$$\min_{x \geq 0} c^T x \quad (2.19)$$

$$\text{sujeito a } Ax \geq b \quad (2.20)$$

Problema dual:

$$\max_{u \geq 0} u^T b \quad (2.21)$$

$$\text{sujeito a } u^T A \leq c^T \quad (2.22)$$

Pelo teorema da dualidade, se um dos problemas (primal ou dual) tem uma solução ótima finita então o outro também tem solução finita e os valores das respectivas funções objetivo são iguais.

Por outro lado, se um dos problemas é ilimitado o outro não tem solução viável.

- Pontos extremos e raios extremos:

Considere o politopo convexo $C = \{u : Au = b, u \geq 0\}$, onde A é uma matriz de dimensão $m \times n$, com posto igual a m , e b é um vetor de dimensão m . Um politopo convexo limitado é denominado de poliedro.

Um ponto extremo do politopo C ou do poliedro é qualquer ponto $u \in C$ que não pode ser expresso como uma combinação convexa de outros pontos $p \in C$, com $p \neq u$.

Todo ponto extremo ou vértice de um politopo corresponde a uma solução básica viável.

Se o politopo for ilimitado, existe uma direção extrema ou vetor $v \geq 0$ que é denominado de raio extremo.

- Relaxação:

A relaxação é uma forma de resolver problemas de grande porte, onde o número de pontos extremos e raios extremos é muito grande. Neste processo, o problema passa

a não considerar todas as restrições inicialmente definidas. Dessa forma, o domínio do problema original é um subconjunto do domínio do problema relaxado.

Com base nesses conceitos, considere o seguinte problema de Programação Linear Inteira Mista descrito de acordo com (2.14) e (2.15):

$$\min_{y \in \{0,1\}, x \geq 0} c^T x + d^T y \quad (2.23)$$

$$\text{sujeito a } Ax + By \geq b \quad (2.24)$$

Utilizando o conceito de projeção, o problema completo de minimização, parametrizado pelas variáveis y , pode ser escrito da seguinte forma:

$$\min_{y \in \{0,1\}} d^T y + Q(y) \quad (2.25)$$

Fixando os valores das variáveis y , $Q(\bar{y})$ é definido como:

$$Q(\bar{y}) = \min_{x \geq 0} c^T x \quad (2.26)$$

$$\text{sujeito a } Ax \geq b - B\bar{y} \quad (2.27)$$

Este problema de Programação Linear é denominado subproblema de Benders.

Pelo teorema da dualidade, tem-se o subproblema dual dado por:

$$Q(\bar{y}) = \max_{u \geq 0} u^T (b - B\bar{y}) \quad (2.28)$$

$$\text{sujeito a } u^T A \leq c^T \quad (2.29)$$

Onde, u^T é um vetor que representa as variáveis duais.

O conjunto de restrições (2.29) define uma região viável para o problema que não depende de y . Esta região é um polítopo convexo independente de y , $C = \{u : u^T A \leq c^T, u \geq 0\}$.

Este polítopo poderá ser limitado, supondo ser um poliedro com p pontos extremos, u_p^T ($p = 1, \dots, P$), ou um polítopo convexo não limitado com p pontos extremos, u_p^T ($p = 1, \dots, P$), e s raios extremos, v_s^T ($s = 1, \dots, S$).

De acordo com Minoux [46], para aplicar a técnica de particionamento sugerida acima, as variáveis y não podem ser escolhidas arbitrariamente. Sendo necessário pelo menos que o subproblema primal $Q(y)$ tenha um conjunto de soluções não vazio e, conseqüentemente, o subproblema dual seja limitado.

Para expressar essa condição, utiliza-se o teorema de Farkas e Minkowski [46]:

$Q(y)$ tem uma solução $x \geq 0$ se e somente se $u^T(b - B\bar{y}) \leq 0$ para todo u para o qual $u^T A \leq 0$.

Analogamente:

- se $v_s^T(b - B\bar{y}) > 0$, para algum s , então a solução do subproblema dual é ilimitada e, conseqüentemente pelo teorema da dualidade, o subproblema primal não possui solução viável. Ocorrendo o mesmo para o problema original, com o valor de y dado;
- se $v_s^T(b - B\bar{y}) \leq 0$, para todo s , então o subproblema dual tem solução ótima finita e o subproblema primal também possui solução finita.

Dessa forma, todo valor de y deve satisfazer a condição:

$$v_s^T(b - By) \leq 0 \text{ para } (s = 1, \dots, S) \quad (2.30)$$

O problema completo de minimização pode então ser escrito usando o teorema da dualidade:

$$\min_{y \in \{0,1\}} [d^T y + \max_{u \geq 0} \{u^T(b - By) : u^T A \leq c^T\}] \quad (2.31)$$

Como a solução ótima de um problema de programação linear corresponde a um vértice da região viável, o subproblema dual pode ser resolvido por enumeração e escrito de uma maneira diferente:

$$Q(\bar{y}) = \max_{u \geq 0} u_p^T (b - B\bar{y}) \text{ para } (p = 1, \dots, P) \quad (2.32)$$

Dessa forma, o problema linear pode ser reescrito como:

$$\min_{y \in \{0,1\}} [d^T y + \max_{u \geq 0} \{u_p^T (b - By)\}] \text{ para } (p = 1, \dots, P) \quad (2.33)$$

Assumindo que os valores de y devem satisfazer a condição (2.30) e adicionando e definindo a variável auxiliar y_0 como o menor limite superior de $Q(y)$, o problema inicial geral passa a ser equivalente ao problema linear inteiro misto denominado de problema mestre:

$$\min_{y \in \{0,1\}} y_0 \quad (2.34)$$

$$\text{sujeito a } d^T y + \bar{u}_p^T (b - By) \leq y_0 \text{ para } (p = 1, \dots, P) \quad (2.35)$$

$$\bar{v}_s^T (b - By) \leq 0 \text{ para } (s = 1, \dots, S) \quad (2.36)$$

$$y_0 \geq 0 \quad (2.37)$$

Onde, y_0 é uma variável escalar maior ou igual a cada $d^T y + u_p^T (b - By)$, para $p = 1, \dots, P$, e, conseqüentemente, maior ou igual ao maior valor deles. Como a função objetivo do problema mestre, (2.34), é minimizar y_0 , pelo menos uma restrição (2.35) estará ativa na solução ótima.

Assim, as restrições (2.35) do problema mestre definem a função de $Q(y)$ do problema original. Logo, o problema mestre, escrito em função das variáveis y e y_0 , também é equivalente ao subproblema e os valores das funções objetivo coincidem na solução ótima.

O conjunto de restrições (2.35) e (2.36) pode ter grandes dimensões. Entretanto, segundo Minoux [46], o número de restrições ativas na solução ótima nunca excede $m + 1$, onde

m é o número de variáveis y . Portanto, utilizando o princípio da programação linear generalizada (geração de restrições), o problema mestre pode ser resolvido considerando explicitamente um número relativamente pequeno dessas restrições. O problema formado a partir de um subconjunto das restrições do problema mestre é denominado problema mestre relaxado. O procedimento de geração de restrições para o problema mestre relaxado consiste no algoritmo Simplex aplicado ao subproblema dual $Q(y)$.

Resumidamente, o algoritmo de Benders baseia-se nos seguintes passos:

- Passo 1)

Inicialmente, atribui-se valores arbitrários para as variáveis y .

- Passo 2)

Para testar a viabilidade dos valores de y , resolve-se o subproblema dual, substituindo tais valores em $Q(y)$.

- Passo 3)

Se o subproblema dual possuir solução ótima limitada, com os valores de y , acha-se \bar{u}_p^T . Neste caso, duas situações podem ocorrer:

- a restrição $d^T y + \bar{u}_p^T (b - By) \leq y_0$ é atendida para os atuais valores de y e y_0 do problema mestre relaxado. Consequentemente, (\bar{y}, \bar{y}_0) é solução para o problema mestre e o algoritmo termina;
- a restrição $d^T y + \bar{u}_p^T (b - By) \leq y_0$ não é atendida para os atuais valores de y e y_0 do problema mestre relaxado. Consequentemente, (\bar{y}, \bar{y}_0) não é solução para o problema mestre. Dessa forma, adiciona-se uma restrição do tipo $d^T y + \bar{u}_p^T (b - By) \leq y_0$ ao problema mestre relaxado, com os valores de \bar{u}_p^T encontrados no subproblema dual.

Se o subproblema dual possuir solução ilimitada, com os valores de y , acha-se \bar{v}_s^T , tal que $\bar{v}_s^T (b - B\bar{y}) > 0$. Dessa forma, a restrição $\bar{v}_s^T (b - By) \leq 0$ não é válida para o atual valor de y do problema mestre relaxado. Consequentemente, (\bar{y}, \bar{y}_0) não é solução para o problema mestre. Dessa forma, deve-se adicionar a restrição $\bar{v}_s^T (b - By) \leq 0$ ao problema mestre relaxado, com os valores de \bar{v}_s^T encontrados no subproblema dual.

Cada nova restrição adicionada ao problema mestre relaxado é denominada de corte de Benders.

- Passo 4)

Resolve-se o problema mestre relaxado com as restrições adicionais definidas pelo passo anterior.

Com a determinação dos novos valores das variáveis y , retorna-se ao passo 2.

Dessa forma, cada iteração do Método de Benders consiste em resolver o problema mestre relaxado e obter a solução ótima para y . Esta solução encontrada no problema mestre relaxado é enviada ao subproblema dual, que gera um novo ponto extremo ou um novo raio extremo para o problema.

Conclui-se, portanto, que, por se tratar de um problema com um número menor de restrições do que o problema mestre, o problema mestre relaxado é um limite inferior para a função objetivo do problema original. Já o subproblema, desde que tenha solução viável para um determinado valor de y , representa um limite superior para o problema completo de programação linear inteira mista.

Assim, cada iteração do método visa diminuir o valor do gap, ou seja, a diferença entre os dois limites. Ao final do processo, os limites, superior e inferior, convergirão para a solução ótima do problema original de Programação Inteira Mista, caso exista.

2.7 Solução dos Problemas de Sequenciamento em Projeto com Restrição de Recursos

Durante quatro décadas, desde os pioneiros trabalhos de Kelley e Walker [36] e Wiest [64], até os dias atuais, muitos pesquisadores tem se dedicado à investigação dos Problemas de Sequenciamento em Projeto com Restrição de Recursos (Oğuz e Bala [49]).

A Otimização Combinatória estuda problemas que são caracterizados por um número finito de soluções possíveis. Embora, em princípio, a solução ótima para um problema finito possa ser encontrada por simples enumeração, na prática esta tarefa é frequentemente impossível, especialmente para problemas de tamanho realístico, onde o número de soluções possíveis pode ser extremamente alto (Blazewicz *et al.* [6]).

Devido à complexidade dos PSPRR, muitos autores desenvolveram procedimentos heurísticos para solucioná-los, como explicita Valls *et al.* [61]. Os estudos mais antigos nessa linha apareceram na década de 60, realizados por Wiest [65] e Fendly [23]. Desde então muitos outros surgiram, como os propostos por Boctor [8], Oğuz e Bala [49] e Kolisch e Sprecher [40]. Oğuz e Bala [49], por exemplo, propõem um algoritmo de busca usando sub-gradientes e técnicas de busca n -dimensionais. Estes métodos de busca encontram um mínimo local, e não conseguem garantir um mínimo global.

Para superar algumas das desvantagens dos métodos de busca, tem sido frequentemente utilizadas heurísticas como o *Simulated Annealing*, os algoritmos Genéticos e o Busca Tabu. Nesse campo, Icmeli e Erenguc [34] apresentam um algoritmo Busca Tabu.

Segundo Kolisch [39], grande parte das heurísticas existentes emprega regras de prioridades para sequenciar as atividades, as quais permitem solucionar conflitos de recursos através da execução de atividades com mais alta prioridade, no tempo mais cedo de disponibilidade do recurso restrito. Mas, embora tenham sido elaboradas muitas regras de prioridade, de acordo com Boctor [8], não há regras que dominem outras ou sejam executadas com o desempenho constantemente melhor do que de outras regras. Os *softwares* comerciais de administração de projetos, como por exemplo Primavera, empregam, usualmente, regras simples de prioridade para resolverem os problemas de sequenciamento. Além disso, sistemas comerciais permitem ao usuário definir suas próprias regras. Consequentemente, os gerentes devem saber sob quais condições a aplicabilidade das regras de prioridade gerará bons resultados.

Uma revisão mais completa da literatura e a compreensão das heurísticas podem ser vistas em Davis e Patterson [17].

Já autores como Schrage [58], Fisher [24], Christofides *et al.* [14] e Kolisch *et al.* [41], apresentam o desenvolvimento de algoritmos exatos para os PSPRR. Dentre os quais, destacam-se os procedimentos baseados na técnica *Branch and Bound*, afirma Brucker *et al.* [10]. De acordo com Oğuz e Bala [49], Pritsker *et al.* [56] formularam o primeiro modelo de Programação Inteira. Nos procedimentos exatos, um dos maiores inconvenientes é que o número de variáveis do modelo cresce muito rapidamente com o tamanho do problema. No entanto, numerosas abordagens para resolver certas versões dos problemas de sequenciamento têm sido desenvolvidas por diversos autores. Dentre os mais competitivos, segundo Valls *et al.* [61], estão os propostos por Brucker *et al.* [10] e Mingozzi *et*

al. [45]. Entretanto, segundo Klein e Scholl [38], somente problemas de pequena e média instâncias, entre trinta atividades, são resolvidos de maneira satisfatória.

2.8 Comentários

Grande parte dos problemas em administração de projetos está relacionada à atribuição e distribuição de recursos limitados entre as diversas atividades que compõem um projeto. Na prática, normalmente, as quantidades disponíveis dos recursos necessários não são suficientes para executar todas essas atividades nos seus menores tempos de processamento. Neste caso, deve-se determinar a melhor sequência de execução das atividades de tal maneira que os recursos disponíveis sejam alocados de forma ótima, ou seja, minimizando o tempo total de execução e considerando as limitações de recursos.

Os diagramas, grafos e modelos são ferramentas matemáticas indispensáveis para uma boa representação e compreensão dos problemas reais. A utilização dessas ferramentas, bem como de um método capaz de resolver tais problemas, torna possível garantir a tomada de decisão que maximize a eficiência da execução de todo o projeto.

Capítulo 3

O Problema de Sequenciamento em Projeto com Restrição de Recursos

3.1 Introdução

Devido a importância dos modelos matemáticos, principalmente no contexto estudado, as próximas seções deste capítulo apresentam alguns problemas relevantes da literatura, modelados pela Programação Matemática.

3.2 Modelos para o Problema de Sequenciamento em Projeto com Restrição de Recursos

Vários modelos de Programação Matemática são propostos para o Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR) - *Resource Constrained Project Scheduling Problem* (RCPSP), como por exemplo em, Artigues *et al.* [2], Balas [3], Deblanc *et al.* [21], Brucker e Knust [9], Carlier e Néron [11], Mingozzi *et al.* [45], Christofides *et al.* [14], Oğuz e Bala [49] e Klein e Scholl [38].

Este problema é constituído, genericamente, por um conjunto de atividades, $N = \{1, 2, \dots, n\}$, e de recursos renováveis, $\mathcal{R} = \{1, \dots, m\}$. Cada tipo de recurso $k \in \mathcal{R}$ está disponível

em uma quantidade constante de R_k unidades. Cada atividade $i \in N$ necessita de uma quantidade constante q_{ik} de unidades do recurso k para ser executada. A atividade i é executada em p_i unidades de tempo sem ser interrompida, ou seja, a restrição de não preempção deve ser respeitada. Os valores de R_k , q_{ik} e p_i são não negativos. As relações de precedências são definidas entre as atividades. O objetivo é determinar a data de início t_i para cada atividade i do projeto, tal que:

- a quantidade de cada tipo de recurso utilizada, durante um determinado período de execução, seja menor ou igual à quantidade total disponível deste recurso;
- todas as relações de precedência sejam satisfeitas;
- o *makespan* $C_{\max} = \max_{i=1}^n C_i$ seja minimizado, onde $C_i = t_i + p_i$ é a data de conclusão (término) da atividade i .

A seguir serão apresentados alguns conceitos e modelos de Programação Matemática propostos na literatura para variações do PSPRR.

3.2.1 Artigues *et al.*

Artigues *et al.* [2] propõem uma formulação para o Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR), a qual, segundo os próprios autores, é uma extensão do modelo matemático clássico do problema de sequenciamento do *Job Shop*:

Dados do problema:

- $N = \{1, \dots, n\}$ é o conjunto das atividades i do projeto;
- $\mathcal{R} = \{1, \dots, r\}$ é o conjunto dos k recursos renováveis.

Cada recurso $k \in \mathcal{R}$ tem uma capacidade finita R_k .

Cada atividade $i \in N$ requer uma quantidade não negativa q_{ik} do recurso $k \in \mathcal{R}$ para ser executada e tem a duração de p_i unidades de tempo.

As restrições de precedência entre as atividades do projeto são modeladas por um conjunto de arcos E , tal que se $(i, j) \in E$, significa que a atividade j tem que começar depois que a execução da atividade i for finalizada.

Variáveis de decisão:

- $t = \{t_1, \dots, t_n\}$, onde t_i é a data de início da atividade i .

A dificuldade do PSPRR resulta da restrição de limitação de recursos, impedindo que algumas atividades, que necessitam do mesmo recurso, comecem a ser executadas simultaneamente. Essa restrição pode ser especificada definindo que uma unidade de recurso dada não pode ser alocada, ao mesmo tempo, para mais de uma atividade. Assim, em uma solução viável, uma unidade do recurso alocada para uma atividade i tem que ser transferida diretamente, após a sua execução, para uma única atividade j .

Desse modo, desde que todas as unidades do mesmo recurso sejam equivalentes, deve-se saber o número de unidades transferidas diretamente de uma atividade para outra. Assim, definem-se duas outras variáveis de decisão do problema como sendo:

- f_{ijk} , o número de unidades do recurso k transferidas diretamente da atividade i para a atividade j ;
- y_{ij} , uma variável binária, fixada em 1, se a atividade j é restringida a começar depois da completa execução da atividade i , e igual a 0, caso contrário.

Além disso, $s = 0$ e $t = n + 1$ são duas atividades fictícias tal que $q_{sk} = q_{tk} = R_k, \forall k \in \mathcal{R}$ e $p_s = p_t = 0$.

M_1 e M_2 são dois números inteiros grandes arbitrários.

Modelo:

$$\text{minimizar } C_{max} \quad (3.1)$$

$$\text{sujeito a } C_{max} \geq t_i + p_i \quad \forall i \in N \quad (3.2)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E \quad (3.3)$$

$$t_j - t_i - M_1 y_{ij} \geq p_i - M_1 \quad \forall i \in N \cup \{s\}, \forall j \in N \cup \{t\} \quad (3.4)$$

$$f_{ijk} - M_2 y_{ij} \leq 0 \quad \forall i \in N \cup \{s\}, \forall j \in N \cup \{t\}, \forall k \in \mathcal{R} \quad (3.5)$$

$$\sum_{j \in N \cup \{t\}} f_{ijk} = q_{ik} \quad \forall i \in N \cup \{s\}, \forall k \in \mathcal{R} \quad (3.6)$$

$$\sum_{i \in N \cup \{s\}} f_{ijk} = q_{jk} \quad \forall j \in N \cup \{t\}, \forall k \in \mathcal{R} \quad (3.7)$$

$$t_i \geq 0 \quad \forall i \in N \quad (3.8)$$

$$f_{ijk} \geq 0 \quad \forall i \in N \cup \{s\}, \forall j \in N \cup \{t\}, \forall k \in \mathcal{R} \quad (3.9)$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in N \cup \{s\}, \forall j \in N \cup \{t\} \quad (3.10)$$

A função objetivo (3.1) é o critério de otimização do problema, ou seja, é minimizar o *makespan*. As restrições (3.2) dão a expressão do *makespan*. As restrições (3.3) garantem as relações de precedência entre as atividades do projeto. As restrições (3.4) são restrições disjuntivas que impedem que duas atividades, relacionadas através de uma unidade de recurso, sejam executadas simultaneamente. As restrições (3.5) descrevem a relação entre as variáveis f_{ijk} e y_{ij} , que implica em $y_{ij} = 0, f_{ijk} = 0, \forall k \in \mathcal{R}$. As restrições (3.6) e (3.7) expressam que o fluxo do recurso k que entra e que sai de uma atividade deve ser igual à capacidade requerida desse recurso (propriedade de conservação do fluxo). O domínio das variáveis t_i, f_{ijk} e y_{ij} é representado pelas restrições (3.8), (3.9) e (3.10).

3.2.2 Balas

A formulação apresentada por Artigues *et al.* [2], baseada no conceito do grafo disjuntivo do problema do *Job Shop*, foi inspirada nas idéias propostas por Balas [3], em 1967. Nesse contexto, Balas [3] afirmou que se há somente uma máquina de cada tipo, duas atividades, i e j , que necessitam da mesma máquina, não podem ser executadas simultaneamente. Para evitar essa possibilidade, o par de arcos disjuntivos, $(i, j), (j, i)$, é adicionado ao

grafo original do problema, criando, assim, o grafo disjuntivo. Considerando a seleção de exatamente um arco de cada par dos arcos disjuntivos, juntamente com o conjunto de arcos fixos, que representam as relações de precedência entre as atividades do projeto, pode-se propor uma solução viável para o problema original. O problema passa a ser então selecionar o conjunto de arcos disjuntivos, tal que o tempo total de execução do projeto seja minimizado. Posteriormente, essa idéia foi generalizada para os Problemas de Sequenciamento em Projeto com Restrição de Recursos. Porém, nestes casos, raramente a quantidade de recursos disponível é igual a uma unidade e duas atividades, que necessitam de um mesmo recurso, não são, necessariamente, incompatíveis. Essas condições, no entanto, modificam a complexidade do problema original, bem como a implementação de algoritmos de solução.

3.2.3 Deblaere *et al.*

Deblaere *et al.* [21], assim como Artigues *et al.* [2], consideram as restrições relativas ao fluxo de recursos. Porém, as restrições de fluxo (3.6) e (3.7), apresentadas por Artigues *et al.* [2], são substituídas formalmente por outras quatro, na formulação proposta por Deblaere *et al.* [21]:

$$\sum_{j \in N \cup \{t\}} f_{sjk} = R_k \quad \forall k \in \mathcal{R} \quad (3.11)$$

$$\sum_{j \in N \cup \{t\}} f_{ijk} = q_{ik} \quad \forall i \in N, \forall k \in \mathcal{R} \quad (3.12)$$

$$\sum_{i \in N \cup \{s\}} f_{ijk} = q_{jk} \quad \forall j \in N, \forall k \in \mathcal{R} \quad (3.13)$$

$$\sum_{i \in N \cup \{s\}} f_{itk} = R_k \quad \forall k \in \mathcal{R} \quad (3.14)$$

As restrições (3.11) e (3.14) garantem que, para todo o tipo de recurso $k \in \mathcal{R}$, o somatório de todo o fluxo que sai da atividade inicial s é igual ao somatório de todo o fluxo que chega na atividade final t . Ambos os somatórios são iguais à quantidade total do recurso disponível, R_k .

Já as restrições (3.12) e (3.13) expressam que, para todo o tipo de recurso $k \in \mathcal{R}$ e para

toda atividade i , tal que $i \neq s$ e $i \neq t$, o somatório do fluxo que entra nessa atividade deve ser igual ao somatório do fluxo que sai (propriedade de conservação do fluxo). Ambos os somatórios devem ser iguais à quantidade de recurso necessária para executar essa atividade.

Dessa forma, irão existir no problema, além das relações de precedência originais, outras relações de precedência impostas pelo fluxo de recursos.

3.2.4 Brucker e Knust

A formulação de Brucker e Knust [9] considera que o horizonte de planejamento pode ser dividido em sucessivos intervalos de tempo e que as atividades de um projeto podem ser particionadas. As partes de atividades diferentes podem ser executadas simultaneamente ou sequencialmente dentro de um mesmo intervalo de tempo, desde que não ultrapasse a sua duração. Porém, as partes de uma mesma atividade, desde que estejam dentro de um mesmo intervalo, só podem ser executadas sequencialmente. O tempo total de processamento e a quantidade disponível e necessária de recurso para a execução de cada atividade são definidos anteriormente no projeto. Janelas de tempo, com as datas de início e término para as atividades serem processadas, também são dados. 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 minimizando o tempo total do projeto.

Brucker e Knust [9] propõem uma maneira de calcular o limite inferior para o Problema de Sequenciamento em Projetos com Restrição de Recursos (PSPRR) baseada na formulação de Programação Linear proposta por Mingozi *et al.* [45].

Na formulação feita por Mingozi *et al.* [45], a execução das atividades que compõem um projeto pode ser interrompida, ou seja, a condição de preempção é permitida. As restrições de precedência são parcialmente relaxadas e tratadas como disjunção.

O método de cálculo do limite inferior é do tipo destrutivo, ou seja, tenta-se encontrar o limite inferior pelo caminho inverso. Sendo T um valor inicial para o limite superior, tenta-se provar a inexistência de um sequenciamento preemptivo com o *makespan* $C_{max} \leq T$. Comprovando a inviabilidade do problema preemptivo, $T + 1$ será o valor do limite inferior

para o PSPRR.

Brucker e Knust [9] e Mingozzi *et al.* [45] associam uma variável artificial a cada subconjunto de atividades que podem ser executadas simultaneamente sem violar as restrições de recurso e precedência. Isso permite a formulação de Programação Linear para o problema.

Dados:

- n denota o número de atividades do projeto;
- $I_l = \{[t_1, t_2], \dots, [t_{l-1}, t_l]\}$ é o conjunto dos $l = \{1, \dots, L\}$ sucessivos intervalos de tempo de comprimento igual a $t_l - t_{l-1}$;
- N_{jl} é um subconjunto $j = \{1, \dots, n_l\}$ de atividades que podem ser executadas simultaneamente no intervalo $l = \{1, \dots, L\}$, não havendo restrições de precedência ou disjunções entre elas e respeitando as restrições de recursos;
- N_l é o conjunto de todos os subconjuntos viáveis N_{jl} que podem ser executados no mesmo intervalo I_l .

A cada subconjunto N_{jl} é associado um vetor de incidência $\alpha_i^{jl} \in \{0, 1\}$ definido por:

$$\alpha_i^{jl} = \begin{cases} 1 & \text{se } i \in N_{jl}, \\ 0 & \text{caso contrário.} \end{cases}$$

Variáveis de decisão:

- u_l , variável artificial para $l = \{1, \dots, L\}$;
- p_{jl} , número de unidades de tempo para que todas as atividades do subconjunto N_{jl} sejam executadas simultaneamente.

Modelo:

$$\text{minimizar } \sum_{l=1}^L u_l \quad (3.15)$$

$$\text{sujeito a } \sum_{l=1}^L \sum_{j=1}^{n_l} a_i^{jl} p_{jl} \geq p_i \quad \forall i = 1, \dots, n \quad (3.16)$$

$$- \sum_{j=1}^{n_l} p_{jl} + u_l \geq -t_l + t_{l-1} \quad \forall l = 1, \dots, L \quad (3.17)$$

$$p_{jl} \geq 0 \quad \forall l = 1, \dots, L \quad \forall j = 1, \dots, n_l \quad (3.18)$$

$$u_l \geq 0 \quad \forall l = 1, \dots, L \quad (3.19)$$

As restrições (3.16) garantem que todas as atividades i serão executadas em pelo menos p_i unidades de tempo. Ou seja, todas as atividades serão executadas completamente. As restrições (3.17) asseguram que o tempo da execução das atividades sequenciadas dentro do intervalo I_l não excede o comprimento desse intervalo. As restrições (3.18) e (3.19) definem o domínio das variáveis.

Uma solução irá existir para a viabilidade do problema preemptivo se e somente se a Programação Linear tiver uma solução ótima de valor igual zero, com todos os valores das variáveis artificiais u_l também iguais a zero. Se o valor ótimo da função objetivo for maior do que zero, o problema preemptivo será inviável e o valor da função objetivo mais um será o limite inferior para o PSPRR.

3.2.5 Carlier e Néron

Carlier e Néron [11], assim como Brucker e Knust [9], propõem um método destrutivo para calcular o valor do limite inferior para o PSPRR.

Esse método possibilita provar que não existe nenhuma solução viável com o valor do *makespan* menor ou igual a um dado valor do limite superior. Assim, o limite superior, acrescentado de uma unidade, passa a ser o limite inferior do *makespan*.

A formulação de Carlier e Néron [11] 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. As partes das atividades são executadas em intervalos consecutivos de tempo. 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.

A seguir será apresentado o modelo proposto por Carlier e Néron [11] que é semelhante ao do Brucker e Knust [9]:

Dados do problema:

- n denota o número de atividades do projeto;
- $I_l = \{[t_1, t_2], \dots, [t_{l-1}, t_l]\}$ é o conjunto dos $l = \{1, \dots, L\}$ sucessivos intervalos de tempo de comprimento igual a $t_l - t_{l-1}$;
- $h = \{1, \dots, H\}$ é o conjunto dos Limites Lineares Inferiores (LLI).

Variáveis de decisão:

- t_l , data de início de cada intervalo l ;
- p_{il} , tempo de processamento da parte da i -ésima atividade executada no intervalo de tempo I_l .

Modelo:

$$\text{minimizar } t_{l+1} \quad (3.20)$$

$$\text{sujeito a } \sum_{l=1}^L p_{il} = p_i \quad \forall i = 1, \dots, n \quad (3.21)$$

$$p_{il} \leq t_{l+1} - t_l \quad \forall i = 1, \dots, n, \forall l = 1, \dots, L \quad (3.22)$$

$$LLI_h(p_{1l}, p_{2l}, \dots, p_{nl}) \leq t_{l+1} - t_l \quad \forall l = 1, \dots, L, \forall h = 1, \dots, H \quad (3.23)$$

$$p_{il} \geq 0 \quad \forall i = 1, \dots, n, \forall l = 1, \dots, L \quad (3.24)$$

$$t_1 = 0 \quad \leq t_2 \leq t_3 \leq \dots \leq t_{L+1} \quad (3.25)$$

$$\text{outras restrições lineares} \quad (3.26)$$

A função objetivo (3.20) é minimizar a data de término do projeto. De acordo com a restrição (3.21), todas as atividades são executadas completamente, ou seja, o somatório de todas as partes executadas da atividade i é igual ao tempo total de processamento dessa atividade. A restrição (3.22) considera que o tempo de processamento da parte da i -ésima atividade, executada dentro do intervalo $[t_l, t_{l+1}]$, não pode ser maior do que o tamanho deste intervalo. A restrição (3.23) define que o intervalo l , de tamanho $t_{l+1} - t_l$, determina um *LLI* para o problema.

Analisando o modelo proposto, a restrição (3.21) estabelece que o somatório de todas as partes que compõem uma atividade i é igual a p_i , tempo total de execução dessa atividade. Esse tempo de processamento está relacionado com a quantidade de recurso utilizada para a execução da atividade i , não variando durante o processo. Assim, conclui-se que tanto a quantidade total de recurso disponível quanto a quantidade de recurso alocada a cada atividade são definidas no início da formulação do problema.

Como descrito anteriormente, Calier e Néron [11] propoem que, caso exista o valor ótimo obtido na resolução do modelo, este será um limite inferior para o valor do *makespan* do PSPRR.

3.2.6 Christofides *et al.*

Segundo Christofides *et al.* [14], se não houvesse a restrição relativa ao recurso em um problema de sequenciamento, este poderia ser resolvido por técnicas como PERT/CPM. Porém, a introdução dessa restrição muda o problema, aumentando o seu grau de dificuldade, sendo necessária a utilização de técnicas de Programação Matemática mais elaboradas.

O Problema de Sequenciamento em Projeto com Restrição de Recursos pode ser formulado, de acordo com Christofides *et al.* [14], como:

$$\text{minimizar } t_n + 1 \quad (3.27)$$

$$\text{sujeito a } t_j - t_i \geq p_i \quad \forall (i, j) \in E \quad (3.28)$$

$$\sum_{N_l} q_{ik} \leq R_k \quad \forall l = 1, \dots, L, \forall k = 1, \dots, K \quad (3.29)$$

onde:

- n : denota o número de atividades do projeto;
- t_i : denota a data de início da i -ésima atividade;
- $E = \{(i, j) : i = 0, 1, \dots, n, j = 1, \dots, n + 1, i \neq j, i \prec j\}$: representa o conjunto de indexadores de pares de atividades cuja precedência é pré-estabelecida;
- p_i : denota o tempo de execução da i -ésima atividade;
- q_{ik} : denota a quantidade do recurso k utilizada na i -ésima atividade;
- N_l é o conjunto de atividades processadas no intervalo $l = \{1, \dots, L\}$;
- r : denota os diferentes tipos de recursos renováveis presentes no projeto;
- R_k : denota a quantidade máxima disponível do k -ésimo recurso.

Esses autores assumem que o tempo de execução de cada atividade é conhecido e independe do momento em que a atividade é realizada. As quantidades de recursos também são constantes conhecidas. O mesmo ocorre com a quantidade total disponível de cada tipo de recurso. Finalmente, os tempos de *setup* são insignificantes ou são incluídos nos tempos de execução.

Posteriormente, devido às limitações computacionais do modelo anterior, Christofides *et al.* [14] apresentam ainda uma formulação de Programação Inteira para o problema:

$$\text{minimizar } \sum_{l=1}^L l y_{nl} \quad (3.30)$$

$$\text{sujeito a } \sum_{l=1}^L y_{il} = 1 \quad \forall i = 1, \dots, n \quad (3.31)$$

$$\sum_{l=1}^L l (y_{jl} - y_{il}) \geq p_i \quad \forall (i, j) \in E \quad (3.32)$$

$$\sum_{i=1}^n q_{ik} \left(\sum_{m=l-p_i+1}^l y_{im} \right) \leq R_k \quad \forall k = 1, \dots, r, \forall l = 1, \dots, L \quad (3.33)$$

$$y_{il} \in \{0, 1\} \quad (3.34)$$

onde:

$$y_{il} = \begin{cases} 1 & \text{se a atividade } i \text{ começa no intervalo } l, \\ 0 & \text{caso contrário.} \end{cases}$$

A função objetivo (3.30) é minimizar o tempo total de execução do projeto. As restrições (3.31) indicam que cada atividade deve começar somente uma vez. As restrições (3.32) garantem que as relações de precedência entre duas atividades sejam respeitadas e as restrições (3.33) garantem que as limitações referentes às quantidades de recursos utilizadas sejam respeitadas.

Uma formulação de Programação Linear é proposta para o problema inteiro original, relaxando as restrições de integralidade (3.34) e adicionando, posteriormente, novas restrições na respectiva formulação linear relaxada (planos de corte). Outras relaxações também são propostas para as restrições (3.33) da formulação inteira, com a introdução de multiplicadores de Lagrange (relaxação Lagrangeana).

3.2.7 Oğuz e Bala

Oğuz e Bala [49] propõem uma formulação para o PSPRR bem próxima da apresentada por Christofides *et al.* [14]. Porém, acrescentam a cada restrição, limites inferiores e superiores para as datas de execução das atividades.

Tal formulação também é modificada por uma equivalente, a fim de reduzir o número total de variáveis do problema.

3.2.8 Klein e Scholl

Klein e Scholl [38] apresentam alguns conceitos para limites inferiores do PSPRR, baseados em aproximações feitas por Christofides *et al.* [14], Mingozzi *et al.* [45] e Brucker *et al.* [10]. Klein e Scholl [38] consideram uma reformulação do PSPRR baseada em todos os subconjuntos viáveis de atividades, ou seja, todas as combinações de atividades, as quais podem ser executadas simultaneamente, sem violar as restrições de precedências

e recursos. Essa reformulação relaxa a restrição de não preempção, permitindo que a execução das atividades seja interrompida. Desde que o número de subconjuntos viáveis cresça exponencialmente com o número de atividades, o esforço computacional é muito grande, embora a qualidade dos limites obtidos seja satisfatória.

3.3 Contribuições complementares

Os problemas estudados pelos autores citados na seção 3.2 deste Capítulo 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. Como constata Konstantinidis [42], essas condições são usuais em muitos outros trabalhos e publicações.

Por outro lado, segundo Poder *et al.* [55], em muitas aplicações concretas, as quantidades de recursos consumidas e, conseqüentemente, os tempos de execução das atividades são raramente constantes.

Dessa forma, o modelo proposto a seguir para o PSPRR, bem os casos gerais apresentados em Dautère-Pèrés *et al.* [16], Poder *et al.* [55] e Pinedo [54], consideram o tempo de execução da atividade i como sendo uma variável contínua, ou seja, podendo variar em função da quantidade de recurso alocada.

Dautère-Pèrés *et al.* [16] definem o tempo de execução da atividade i utilizando o recurso k como sendo p_{ik} . Cada atividade pode utilizar mais de um tipo de recurso e o maior tempo de execução definirá o tempo total de conclusão de uma atividade. Ou seja, P_i é o tempo total de execução da atividade i , onde $P_i = \max p_{ik}$.

Em Poder *et al.* [55], cada atividade i é particionada e composta por uma sequência de n consecutivos intervalos i_1, i_2, \dots, i_n . Cada intervalo da atividade i é definido por um tempo de execução mínimo, \underline{p} , um tempo de execução máximo, \bar{p} , e uma função que representa a relação entre a quantidade de recurso alocada e o tempo, $R(t)$.

Pinedo [54] considera que o tempo de execução de uma atividade é uma função linear da quantidade de recurso alocada. Ou seja, quanto mais recurso alocado, menor será o tempo

de execução da atividade. Ele considera ainda que existem, para cada atividade, um tempo de execução mínimo, \underline{p}_i , e um tempo de execução máximo, \bar{p}_i . Ambos correspondem, respectivamente, a uma quantidade de recurso alocada máxima, \bar{q}_i , e a uma quantidade de recurso alocada mínima, \underline{q}_i . O coeficiente de redução do tempo de execução da atividade i em função da quantidade de recurso alocada é definido por a_i , onde:

$$a_i = \frac{\bar{p}_i - \underline{p}_i}{\bar{q}_i - \underline{q}_i};$$

Alguns dos conceitos e definições considerados a seguir podem ser fundamentados a partir dos exemplos apresentados anteriormente em Dauzère-Pèrés *et al.* [16], Poder *et al.* [55] e Pinedo [54]. Porém, o contexto de aplicação de tais conceitos e definições passa a ser bastante específico. Pois, a formulação proposta se refere especificamente ao Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR).

A nova formulação proposta generaliza o problema estudado por Artigues *et al.* [2], considerando adicionalmente:

- as restrições apresentadas em Deblaere *et al.* [21];
- a determinação dos melhores tempos de execução das atividades em função da alocação dos recursos disponíveis.

Essa relação é não linear, ou seja, o tempo de execução, p_{ik} , de uma atividade i é inversamente proporcional ao aumento da quantidade, q_{ik} , de recurso k alocada.

Em Vieira [62], uma relação não linear entre o tempo de execução e a quantidade de recurso alocada para cada atividade foi linearizada, aproximando-se a curva por uma secante única. No entanto, este procedimento gera imprecisões relevantes ao problema, fornecendo soluções muitas vezes bastante diferentes da solução ótima exata.

3.4 Conclusão

Os problemas de sequenciamento classificados como PSPRR são problemas relevantes da literatura que motivam diversos autores. Devido, porém, à complexidade e dificuldade de resolução de tais problemas, muitas variações da formulação original podem ser observadas.

Balas [3] introduz uma abordagem diferente, 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, como por exemplo, a apresentada por Artigues *et al.* [2].

A formulação proposta por Artigues *et al.* [2] considera ainda o PSPRR como um problema de fluxo. O que também pode ser constatado em Deblaere *et al.* [21].

Brucker e Knust [9], assim como, Carlier e Néron [11], Mingozzi *et al.* [45], Christofides *et al.* [14], Oğuz e Bala [49] e Klein e Scholl [38] modificam a formulação do problema original relaxando algumas restrições.

Brucker e Knust [9], Carlier e Néron [11], Klein e Scholl [38] também propõem o cálculo de limites inferiores 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 dadas do problema. Porém, o tempo de execução deve variar em função da quantidade de recurso alocada para a realização da atividade.

3.5 Contribuição da tese

No intuito de explorar a relação entre o tempo de processamento e a quantidade de recurso alocada para cada atividade do projeto, no próximo capítulo é apresentada uma nova

formulação para o PSPRR. Essa formulação é baseada no modelo proposto por Artigues *et al.* [2], onde o problema de sequenciamento é também, como descrito anteriormente, um problema de fluxo de recurso. Porém, tanto o tempo de processamento como a quantidade de recurso alocada não serão mais dados do problema. Agora, esses parâmetros passam a ser tratados como variáveis contínuas de otimização.

Nesta pesquisa, 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, capazes de fornecer soluções subótimas tão próximas da solução ótima quanto se queira.

Outros tipos de restrição de precedência, além daquelas presentes na literatura citada, são incorporados ao problema, conforme será explicado no Capítulo 4.

Um aspecto relevante e complementar do presente estudo é a sua aplicabilidade a problemas com recursos discretos. Neste caso, obtêm-se soluções relaxadas. Essas soluções, aparentemente, podem ser computacionalmente mais fáceis de serem obtidas e, se necessário, podem ser facilmente ajustadas a posteriori para atenderem ao quesito de recurso discreto no tempo.

Capítulo 4

Uma formulação estendida para o Problema de Sequenciamento em Projeto com Restrição de Recursos

4.1 Introdução

Os problemas estudados neste trabalho consistem em definir temporalmente quais as quantidades de cada tipo de recurso devem ser utilizadas para realizar cada uma das atividades do projeto, de maneira que o objetivo do planejamento seja atingido, obedecendo a sequência tecnológica de execução. Exemplos de projetos são construções de fábricas, pontes, estradas e prédios, manutenções em equipamentos e aeronaves, desenvolvimento de *softwares* e experimentos de pesquisa, entre muitos outros. Na construção de uma ponte, por exemplo, a compra de material, contratação de mão de obra e as diversas etapas que compõem a construção, como preparação do terreno, fundação, estruturas de pilares, vigas e lajes, pavimentação, são algumas das atividades do projeto. Essas atividades têm uma duração prevista e para serem executadas estão sujeitas a relações de precedência. Isto é, certas atividades só podem ser realizadas após outras terem sido completadas. Ao serem executadas, as atividades consomem recursos, como máquinas e mão de obra. O objetivo desta classe de problemas é realizar as atividades de forma a minimizar a data de entrega do projeto, obedecendo às relações de precedência entre as atividades e a disponibilidade de recursos. Outro exemplo de projeto pode ser visto em

Gemmill e Edwards [26] que estudam o problema de manutenção de aeronaves de carga em uma base da força aérea americana. A manutenção destas aeronaves envolve uma grande quantidade de atividades, como desmontar e inspecionar quase todos os sistemas. O problema consiste em encontrar a melhor sequência para completar as atividades de forma que a aeronave retorne da manutenção o mais rápido possível. Outros exemplos também podem ser encontrados em Neumann *et al.* [48].

Para esses problemas, a decisão do administrador do projeto consiste em definir as quantidades de cada tipo de recurso que devem ser utilizadas para executar cada uma dessas atividades. Note que as consequências desta decisão influenciará diretamente no tempo total de execução do projeto.

O tempo de execução de cada atividade do projeto é escrito em função da quantidade de recursos alocados para a realização da atividade.

A fim de explorar essa relação no contexto dos Problemas de Sequenciamento em Projeto com Restrição de Recursos (PSPRR), a seção 4.2 apresenta a formulação não linear do PSPRR para um tipo de recurso. A seção 4.3 contém toda a argumentação e o desenvolvimento para uma reformulação linear aproximada do PSPRR. A formulação do PSPRR como um problema de Programação Linear Inteira Mista com um único tipo de recurso contínuo está na seção 4.4. A utilidade da aplicação do procedimento em problemas com recursos discretos é discutida na seção 4.7. A seção 4.8 apresenta a formulação não linear do PSPRR para múltiplos de recursos. A utilização de mais de um tipo de recurso contínuo e discreto no problema de Programação Linear Inteira Mista estão na seção 4.9 e na seção 4.10, respectivamente. Finalmente, a seção 4.11 apresenta as principais conclusões sobre o trabalho.

4.2 Formulação não linear do PSPRR com um único tipo de recurso

Considere o problema de realizar em tempo mínimo uma sequência de n atividades que necessitam compartilhar um mesmo recurso R , sendo que o tempo de realização de uma atividade varia de forma não linear com a quantidade de recurso aplicada.

No contexto da teoria da otimização, trata-se de um problema não linear de sequenciamento de atividades, para o qual se pretende minimizar o tempo total de execução, através da alocação ótima dos recursos. Matematicamente, o problema pode ser formulado como:

$$\text{minimizar } t_{n+1} \quad (4.1)$$

$$\text{sujeito a } t_{n+1} - t_i - p_i - \delta_i \geq 0 \quad i = 0, \dots, n \quad (4.2)$$

$$p_i = \phi_i(q_i) \quad i = 1, \dots, n \quad (4.3)$$

$$(t_j - t_i - p_i - \delta_i r_{ij}) y_{ij} \geq 0 \quad \forall (i, j) \in E \quad (4.4)$$

$$0 \leq f_{ij} \leq M_2 y_{ij} \quad \forall (i, j) \in E \quad (4.5)$$

$$\sum_{j=1}^{n+1} f_{0j} = R \quad (4.6)$$

$$\sum_{j=1}^{n+1} f_{ij} - x_i = \underline{q}_i \quad i = 1, \dots, n \quad (4.7)$$

$$\sum_{i=0}^n f_{ij} - x_j = \underline{q}_j \quad j = 1, \dots, n \quad (4.8)$$

$$\sum_{i=0}^n f_{in+1} = R \quad (4.9)$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall (i, j) \in E - \{E_0 \cup E_1\} \quad (4.10)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (4.11)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (4.12)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (4.13)$$

$$t_i \geq 0 \quad i = 0, 1, \dots, n+1 \quad (4.14)$$

$$0 \leq x_i \leq \bar{q}_i - \underline{q}_i \quad i = 1, \dots, n \quad (4.15)$$

onde:

- t_i : 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_{n+1} : denota a data de término do projeto, correspondente a data de início de uma atividade artificial, com índice $n+1$, a qual é precedida por todas as demais atividades;

- q_i : denota a quantidade de recurso utilizada na execução da i -ésima atividade;
- \underline{q}_i e \bar{q}_i : denotam, respectivamente, a quantidade mínima e a quantidade máxima de 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_{ij} : denota a quantidade de recurso que será transferida da i -ésima atividade para a j -ésima atividade;
- x_i : denota a quantidade do recurso que será alocada para acelerar a execução da i -ésima atividade;
- ϕ_i : representa fator de proporcionalidade;
- M_2 : representa uma constante positiva com valor suficientemente grande;
- R : denota a quantidade disponível do recurso utilizado. Esta quantidade deve ser no mínimo igual ao valor da quantidade mínima de recurso permitida na execução da i -ésima atividade, ou seja, $R \geq \max \underline{q}_i$;
- $E = \{(i, j) : i = 0, 1, \dots, n, j = 1, \dots, n+1, i \neq j, i \prec j\}$: representa o conjunto de indexadores de pares de atividades (todo o conjunto disjuntivo);
- $E_0 = \{(i, j) : i = 0, 1, \dots, n, j = 1, \dots, n+1, i \neq j, i \prec j\}$: representa o conjunto de indexadores de pares de atividades que não podem ser executadas simultaneamente;
- $E_1 = \{(i, j) : i = 0, 1, \dots, n, j = 1, \dots, n+1, i \neq j, i \prec j\}$: 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 ;
- $r_{ij} = \begin{cases} 1 & \text{se } (i, j) \in E_0 \cup E_1; \\ 0 & \text{caso contrário;} \end{cases}$

Note 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

(por exemplo, tempo de cura de um material aplicado). Ou seja, entre duas atividades sequenciadas a priori (a última só pode 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 r_{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.

A função objetivo (4.1) indica a minimização da data de finalização do projeto (t_{n+1}). As restrições (4.2) impõem que a atividade artificial (de índice $n + 1$) deve suceder todas as demais atividades ($y_{in+1} = 1, i = 0, 1, \dots, n$). As restrições (4.3) definem o tempo de execução de cada atividade do projeto como uma função não linear da quantidade de recurso alocada. As restrições (4.4) 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 (4.5) 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 $f_{ij} = 0$. As restrições (4.6) estabelecem que a somatória dos fluxos de recurso que saem da atividade inicial é igual à quantidade total disponível desse recurso. As restrições (4.7) e (4.8) 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 (4.9) garantem que a quantidade total de recurso que chega na atividade final é igual à quantidade total disponível desse recurso. As restrições (4.10) e (4.11) impossibilitam que a variável y_{ij} e a variável y_{ji} sejam simultaneamente iguais a um. Adicionalmente, as restrições (4.10) definem uma relação disjuntiva entre duas atividades e as restrições (4.11) definem a sequência entre duas atividades. As restrições (4.12) garantem a precedência estabelecida a priori. As restrições (4.13) definem o domínio das variáveis y_{ij} . Finalmente, as restrições (4.14) e (4.15) estabelecem, respectivamente, a positividade das datas de início das atividades e os limites mínimo (\underline{q}_i) e máximo (\bar{q}_i) para a quantidade de recurso.

Note que o problema possui não linearidade nas restrições (4.3) e (4.4). As restrições (4.4), conforme será mostrado adiante, serão manipuladas no presente trabalho de acordo com procedimento já consagrado na literatura (ver por exemplo, Artigues *et al.* [2]). Por outro lado, as restrições (4.3), foco do presente trabalho, receberão um tratamento original, cujo desenvolvimento será apresentado a seguir.

4.3 Manipulação das curvas de tempo de processamento das atividades

Na presente pesquisa, considera-se o tempo de execução de uma atividade (p_i) inversamente proporcional à quantidade de recurso alocada (q_i), conforme ilustrado na figura 4.1, obedecendo à equação:

$$p_i(q_i) = \alpha_i/q_i, q_i \in [\underline{q}_i, \bar{q}_i];$$

onde, α_i representa um fator de proporcionalidade, correspondente à atividade i , sendo (\underline{q}_i) e (\bar{q}_i) as quantidades mínima e máxima admissíveis de recurso para executar a atividade.

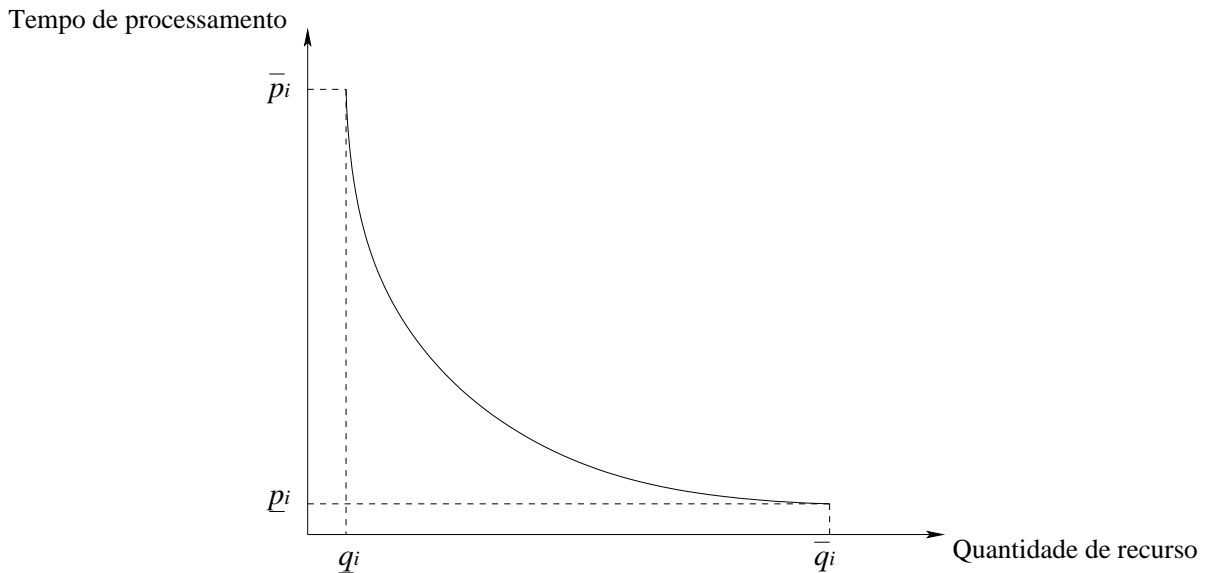


Figura 4.1: Gráfico da função não linear $p_i(q_i)$

4.3.1 Aproximação linear por um segmento de reta

Uma forma simples de aproximação linear para a função $p_i(q_i)$, adotada em Vieira [62], consiste em substituí-la pela reta secante, conforme mostrado na figura 4.2.

A linha mostrada na figura 4.2 tem a equação:

$$p_i = \bar{p}_i - a_i x_i,$$

com:

$$a_i = \frac{\bar{p}_i - \underline{p}_i}{\bar{q}_i - \underline{q}_i},$$

$$0 \leq x_i \leq \bar{q}_i - \underline{q}_i,$$

com:

- \bar{p}_i : o tempo de execução da atividade i , quando a quantidade de recurso \underline{q}_i é alocada;
- \underline{p}_i : o tempo de execução da atividade i , quando a quantidade de recurso \bar{q}_i é alocada.

Note que a_i representa o fator de redução do tempo de processamento em relação a x_i , que por sua vez representa a quantidade de recurso alocada para a execução da atividade i , além da quantidade mínima \underline{q}_i . Os valores de \bar{p}_i e \underline{p}_i correspondem, respectivamente, ao máximo e ao mínimo tempo possível para a alocação do recurso na execução da atividade i .

Conforme já comentado, essa aproximação gera algumas dificuldades, como por exemplo, uma discrepância muito grande entre o tempo de execução aproximado e o real, o que altera significativamente a solução ótima obtida.

4.3.2 Aproximação linear por um número arbitrário de segmentos de reta

Com o objetivo de controlar os erros de aproximação e melhorar a precisão dos resultados, na presente pesquisa propõe-se uma aproximação linear através de um número arbitrário

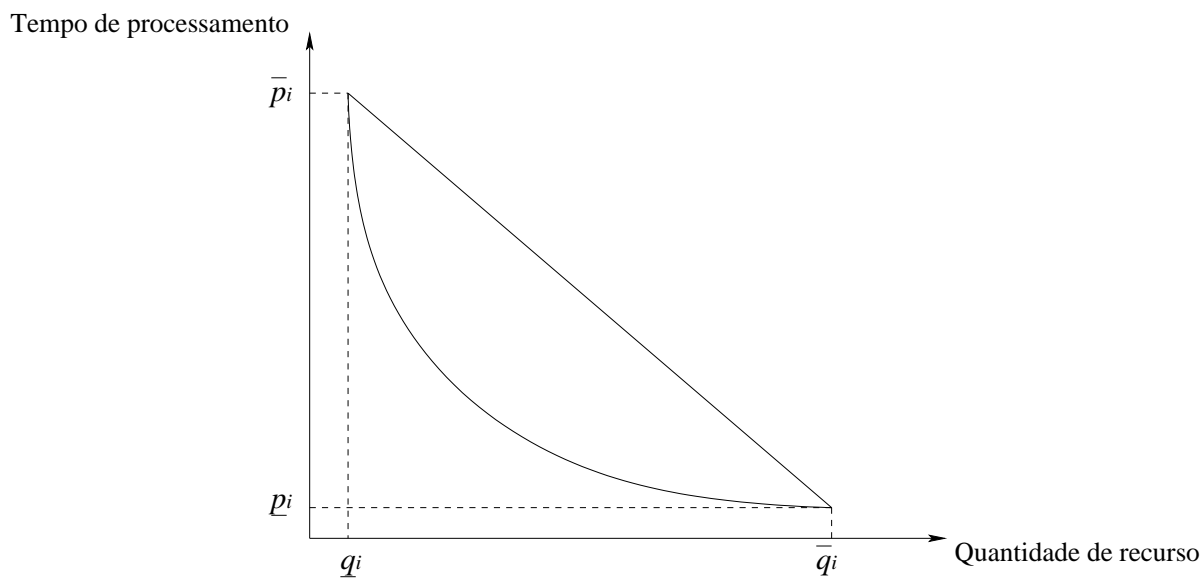


Figura 4.2: Gráfico da aproximação linear por um segmento de reta

de segmentos de reta, tal que as aproximações dos tempos de execução sejam suficientemente próximas dos seus valores exatos. Para compreender o procedimento, considere, como na figura 4.3, uma aproximação através de três segmentos de reta: o primeiro para o intervalo $[q_i, q_{i1}]$, o segundo para o intervalo $[q_{i1}, q_{i2}]$ e o terceiro para o intervalo $[q_{i2}, \bar{q}_i]$.

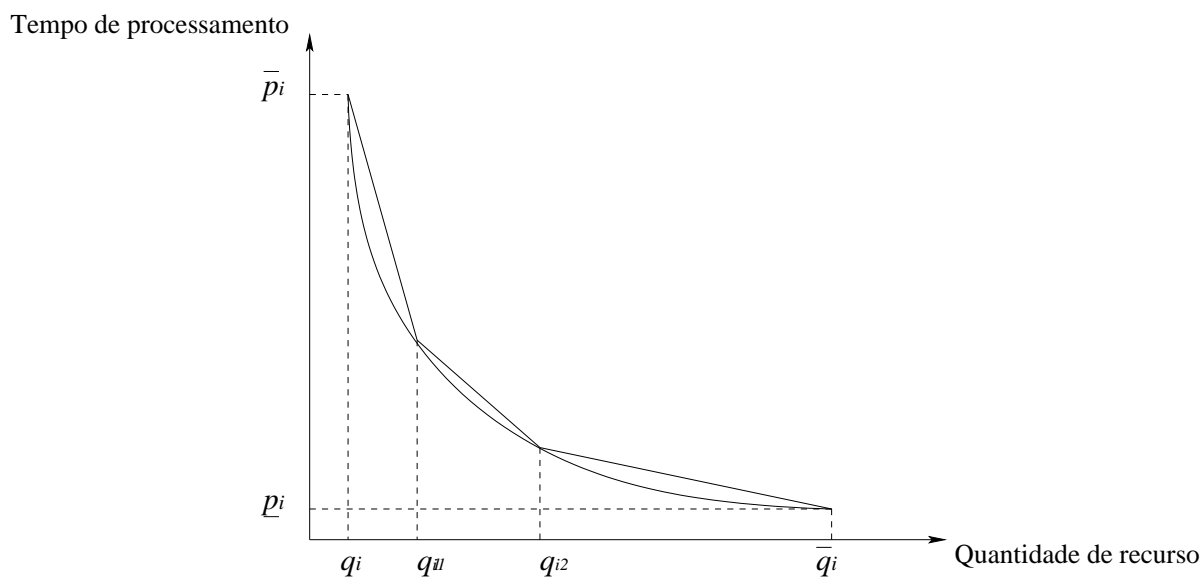


Figura 4.3: Aproximação por três segmentos de reta

Observe que o tempo calculado pelas aproximações lineares será sempre maior ou igual ao valor exato. Assim, a solução ótima aproximada será sempre viável para o problema original. Note ainda que, com a aproximação da figura 4.3, a diferença entre os tempos de processamento aproximado e real fica bastante reduzida em relação à figura 4.2. Note também que a aproximação por três segmentos é apenas uma exemplificação.

Para compreender a segmentação arbitrária da curva não linear de tempo de processamento como função dos recursos alocados considere agora a figura 4.4.

Na figura 4.4 representa-se um segmento arbitrário aproximando a curva sobre o intervalo arbitrário $q_i \in [q_{iw}, q_{iw+1}]$.

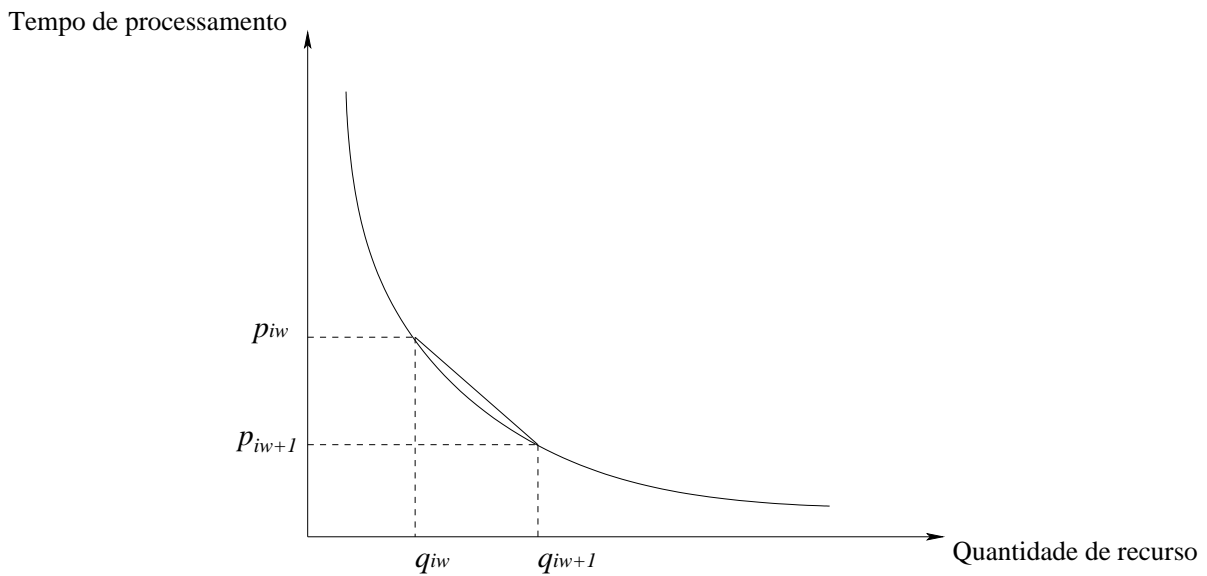


Figura 4.4: Segmentação arbitrária da curva do tempo de processamento

Neste caso, o segmento indicado na figura 4.4 possui como equação:

$$p_i = p_{iw} - a_{iw}x_i,$$

com:

$$a_{iw} = \frac{p_{iw} - p_{iw+1}}{q_{iw+1} - q_{iw}},$$

$$0 \leq x_i \leq q_{iw+1} - q_{iw}$$

Note que q_{iw} e q_{iw+1} correspondem à menor quantidade e à maior quantidade de recurso para o intervalo considerado, respectivamente. Por sua vez, p_{iw} e p_{iw+1} correspondem, respectivamente, ao maior tempo e ao menor tempo de processamento para o intervalo considerado.

4.3.3 Aproximação linear por vários segmentos de reta aplicada ao PSPRR

A aplicação da aproximação da curva de recursos alocados ao Problema de Sequenciamento em Projeto com Restrição de Recurso (PSPRR) é feita estendendo-se cada segmento sobre todo o intervalo do problema $q_i \in [q_{iw}, q_{iw+1}]$. Para compreender a estratégia, considere inicialmente uma aproximação por dois segmentos como mostrado na figura 4.5.

Note que, a reta suporte de cada segmento está estendida sobre todo o intervalo de pertinência, ou seja, $q_i \in [q_{iw}, q_{iw+1}]$. Assim, o primeiro segmento estendido tem a seguinte representação matemática:

$$p_i = p_{i1} - a_{i1}x_i,$$

com:

$$a_{i1} = \frac{p_{i1} - p_{i2}}{q_{i2} - q_{i1}},$$

$$0 \leq x_i \leq \bar{q}_i - \underline{q}_i = q_{i3} - q_{i1},$$

Tempo de processamento

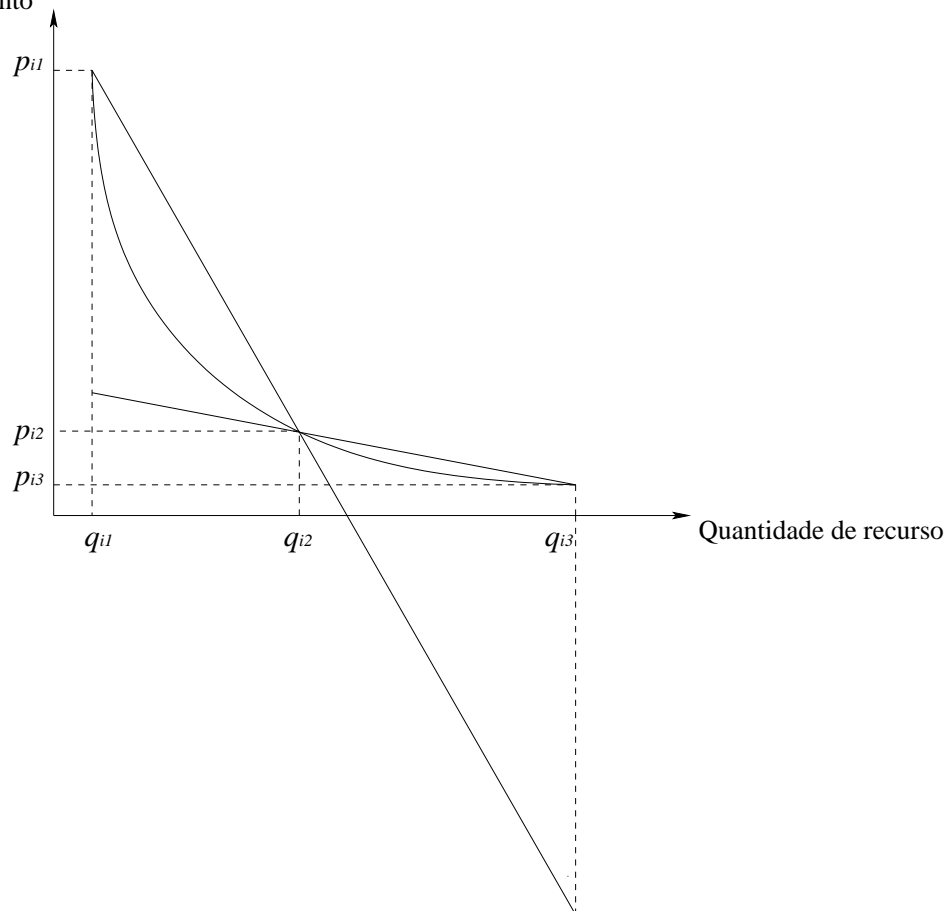


Figura 4.5: Aproximação da curva do tempo de processamento por dois segmentos

Por sua vez, para o segundo segmento estendido temos:

$$p_i = p_{i2} - a_{i2}x_i,$$

com:

$$a_{i2} = \frac{p_{i2} - p_{i3}}{q_{i3} - q_{i2}},$$

$$0 \leq x_i \leq \bar{q}_i - \underline{q}_i = q_{i3} - q_{i1},$$

A idéia é incorporar a aproximação por dois segmentos no problema de otimização através

das desigualdades:

$$p_i = p_{i1} - a_{i1}x_i, \quad 0 \leq x_i \leq q_{i3} - q_{i1}, \quad (4.16)$$

$$p_i = p_{i2} - a_{i2}x_i, \quad 0 \leq x_i \leq q_{i3} - q_{i1} \quad (4.17)$$

Note que o intervalo estabelecido para x_i , para ambas as aproximações, corresponde à $\underline{q}_i \leq q_i \leq \bar{q}_i$. Observe que o menor valor de p_i satisfazendo a ambas as desigualdades, estará sobre o primeiro segmento se $q_i \in [q_{i1}, q_{i2}]$ e estará sobre o segundo segmento se $q_i \in [q_{i2}, q_{i3}]$.

Não é difícil observar que, para o problema de otimização cujo objetivo é minimizar o tempo necessário para realização das atividades, o valor de p_i deverá ser o menor valor satisfazendo as desigualdades (4.16) e (4.17). Portanto, na solução ótima, o valor de p_i estará de acordo com a equação (4.16), no caso de q_i ótimo pertencer ao primeiro intervalo, ou de acordo com a equação (4.17), no caso de q_i ótimo pertencer ao segundo intervalo.

Este raciocínio pode ser estendido a um número arbitrário de segmentos, levando às seguintes desigualdades:

$$\left\{ \begin{array}{l} p_i \geq p_{i1} - a_{i1}x_i \\ p_i \geq p_{i2} - a_{i2}x_i \\ \cdot \\ \cdot \\ \cdot \\ p_i \geq p_{iw^i} - a_{iw^i}x_i \end{array} \right. \quad 0 \leq x_i \leq \bar{q}_i - \underline{q}_i$$

onde, w^i denota o número de segmentos adotados para a aproximação do tempo de processamento da atividade i .

4.4 Formulação do PSPRR como um problema de Programação Linear Inteira Mista com um único tipo recurso contínuo

Aplicando a segmentação múltipla para cada atividade do projeto, podemos reescrever o PSPRR como o seguinte problema de programação linear inteira mista:

$$\text{minimizar } t_{n+1} \quad (4.18)$$

$$\text{sujeito a } t_{n+1} - t_i - p_i - \delta_i \geq 0 \quad i = 1, \dots, n \quad (4.19)$$

$$p_i \geq p_{iw^i} - a_{iw^i} x_i \quad i = 1, \dots, n \quad (4.20)$$

$$t_j - t_i - p_i - \delta_i r_{ij} - M_1 y_{ij} \geq -M_1 \quad \forall (i, j) \in E \quad (4.21)$$

$$f_{ij} - M_2 y_{ij} \leq 0 \quad \forall (i, j) \in E \quad (4.22)$$

$$\sum_{j=1}^{n+1} f_{0j} = R \quad (4.23)$$

$$\sum_{j=1}^{n+1} f_{ij} - x_i = \underline{q}_i \quad i = 1, \dots, n \quad (4.24)$$

$$\sum_{i=0}^n f_{ij} - x_j = \underline{q}_j \quad j = 1, \dots, n \quad (4.25)$$

$$\sum_{i=0}^n f_{in+1} = R \quad (4.26)$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall (i, j) \in E - \{E_0 \cup E_1\} \quad (4.27)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (4.28)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (4.29)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (4.30)$$

$$t_i \geq 0 \quad i = 0, 1, \dots, n+1 \quad (4.31)$$

$$0 \leq x_i \leq \bar{q}_i - \underline{q}_i \quad i = 1, \dots, n \quad (4.32)$$

$$f_{ij} \geq 0 \quad \forall (i, j) \in E \quad (4.33)$$

onde t_i , n , t_{n+1} , \underline{q}_i , \bar{q}_i , p_i , y_{ij} , f_{ij} , x_i , M_2 , R , E_0 , E_1 , δ_i e r_{ij} são como definidos no modelo da seção 4.2 e:

- w^i : denota o número de segmentos para aproximar o tempo de processamento da i -ésima atividade;
- M_1 : representa uma constante positiva com valor suficientemente grande.

Note também que, de acordo com as restrições (4.20), o tempo de processamento de cada atividade do projeto agora é definido como uma função linear da quantidade de recurso alocada e o valor ótimo de p_i será fornecido pelo segmento de reta apropriado. Observe também que as restrições não lineares (4.4), apresentadas na formulação original, foram substituídas pelas restrições lineares (4.21), de acordo com as manipulações apresentadas em Artigues *et al.* [2].

4.5 Uma estimativa prática para o valor do parâmetro M_1

Na literatura, ver Artigues *et al.* [2], o parâmetro M_1 tem sido adotado arbitrariamente grande, sem referências práticas sobre o mesmo.

Com o intuito de estabelecer valores mais justos para M_1 e, além disso, introduzir uma forma automática de atribuir um valor para M_1 , não suscetível de equívocos, propõe-se que ele seja adotado como:

$$M_1 = \sum_{i=1}^n (\bar{p}_i + \delta_i)$$

onde \bar{p}_i representa o tempo máximo de execução da atividade i . Trata-se de uma atribuição válida para o parâmetro M_1 .

4.6 Uma estimativa prática para o valor do parâmetro M_2

Na literatura, ver Artigues *et al.* [2], o parâmetro M_2 também tem sido adotado arbitrariamente grande, sem referências práticas sobre o mesmo.

Com o mesmo intuito de estabelecer valores válidos, mais realistas, para M_2 , propõe-se que ele seja adotado como:

$$M_2 = l_{ij},$$

onde, $l_{ij} = l_{ji} = \min\{\bar{q}_i, \bar{q}_j\}$, ou seja, l_{ij} representa o menor entre os limites superiores para os fluxos f_{ij} e f_{ji} . Novamente, pode-se perceber a validade do valor adotado.

4.7 Formulação do PSPRR como um problema de Programação Linear Inteira Mista com um único tipo de recurso discreto

Embora a formulação acima tenha sido desenvolvida para problemas com recurso contínuo (podendo assumir qualquer valor real dentro dos intervalos de validade), ela pode também ser utilizada para obter soluções relaxadas de problemas com recurso discreto (podendo assumir apenas valores inteiros).

Para isso deve-se inicialmente modelar o problema considerando o tempo de execução de cada atividade como uma função contínua do recurso. Em seguida, cada função deve ser linearizada através de vários segmentos de reta. A princípio, parece interessante adotar um segmento de reta para cada unidade inteira de recurso. Desta maneira, espera-se a obtenção de soluções relaxadas que, ou já são naturalmente inteiras apesar da relaxação, ou podem migrar para soluções com recursos inteiros através de pequenos ajustes realizados a posteriori. Esses ajustes consistem, inicialmente, nas aproximações (arredondamentos) que deverão ser feitas nos valores obtidos das variáveis x_{ik} , a partir da solução relaxada,

tornando-os valores inteiros.

Posteriormente, recalcula-se os respectivos tempos de processamento e as relativas datas de início de cada atividade do projeto, obedecendo a sequência tecnológica de execução definida previamente pela solução relaxada. Em outras palavras, as soluções relaxadas são tratadas a posteriori.

A idéia de tratar recurso discreto através de funções contínuas pode ser interessante nos problemas de otimização inteira mista. Por exemplo, na aplicação do método de Benders, a redução no número de variáveis discretas, ainda que aumentando o número de variáveis contínuas, a princípio, facilita a obtenção de soluções ótimas.

De fato, no caso do problema de sequenciamento com recursos discretos, uma abordagem relaxada terá como variáveis inteiras somente as variáveis binárias (y_{ij}). Todas as demais são contínuas. Considerando o método de resolução proposto, o Método de Decomposição de Benders, esta forma de resolver o problema torna-se mais atrativa, uma vez que o problema mestre, que é o que determina o tempo de resolução do método, fica com um número menor de variáveis inteiras, aliviando o tempo de resolução do problema como um todo.

Assim, a formulação proposta se aplica principalmente no tratamento de recursos contínuos, embora também possa ser aplicada no tratamento de recursos discretos. Essa característica a torna bastante genérica, embora exista na literatura proposições para o caso de recursos discretos, como por exemplo as propostas por Artigues *et al.* [2] e Deblaere *et al.* [21]. Mas, não temos conhecimento, na literatura, de proposições feitas para recursos contínuos, no contexto do PSPRR, considerando o tempo de processamento e as quantidades de recursos variáveis do problema. Pois, tanto em trabalhos apresentados pelos autores citados acima como em Brucker e Knust [9], Carlier e Néron [11] e Christofides *et al.* [14], a duração das atividades e a quantidade de recursos utilizada são dados do problema.

4.8 Formulação não linear do PSPRR com vários tipos de recursos

Podemos reescrever o modelo de Programação não linear do PSPRR, proposto na seção 4.2, considerando agora vários tipos de recursos renováveis:

$$\text{minimizar } t_{n+1} \quad (4.34)$$

$$\text{sujeito a } t_{n+1} - t_i - p_i - \delta_i \geq 0 \quad i = 0, \dots, n \quad (4.35)$$

$$p_i = \max_{k \in \mathcal{R}} \phi_{ik}(q_{ik}) \quad i = 1, \dots, n \quad (4.36)$$

$$(t_j - t_i - p_i - \delta_i r_{ij}) y_{ij} \geq 0 \quad \forall (i, j) \in E \quad (4.37)$$

$$0 \leq f_{ijk} \leq M_2 y_{ij} \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (4.38)$$

$$\sum_{j=1}^{n+1} f_{0jk} = R_k \quad \forall k \in \mathcal{R} \quad (4.39)$$

$$\sum_{j=1}^{n+1} f_{ijk} - x_{ik} = \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (4.40)$$

$$\sum_{i=0}^n f_{ijk} - x_{jk} = \underline{q}_{jk} \quad j = 1, \dots, n, \forall k \in \mathcal{R} \quad (4.41)$$

$$\sum_{i=0}^n f_{in+1k} = R_k \quad \forall k \in \mathcal{R} \quad (4.42)$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall (i, j) \in E - \{E_0 \cup E_1\} \quad (4.43)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (4.44)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (4.45)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (4.46)$$

$$t_i \geq 0 \quad i = 0, 1, \dots, n+1 \quad (4.47)$$

$$0 \leq x_{ik} \leq \bar{q}_{ik} - \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (4.48)$$

Os parâmetros t_i , n , t_{n+1} , q_{ik} , \underline{q}_{ik} , \bar{q}_{ik} , p_{ik} , y_{ij} , f_{ijk} , x_{ik} , ϕ_{ik} , M_2 , R_k , E , E_0 , E_1 , δ_i , r_{ij} são definidos como anteriormente, na seção 4.2, e os parâmetros:

- $\mathcal{R} = \{1, \dots, \bar{k}\}$: denota o conjunto dos diferentes tipos de recursos renováveis.
- R_k : denota a quantidade disponível do k -ésimo recurso.

Note que o raciocínio utilizado para o desenvolvimento do modelo com apenas um tipo

de recurso agora é generalizado para um modelo que engloba a utilização de mais de um tipo de recurso no Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR).

4.9 Formulação do PSPRR como um problema de Programação Linear Inteira Mista com vários tipos de recursos contínuos

Podemos também reescrever o modelo de Programação Linear Inteira Mista para o PSPRR considerando agora vários tipos de recursos renováveis:

$$\text{minimizar } t_{n+1} \quad (4.49)$$

$$\text{sujeito a } t_{n+1} - t_i - p_i - \delta_i \geq 0 \quad i = 1, \dots, n \quad (4.50)$$

$$p_i \geq p_{iw^i k} - a_{iw^i k} x_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (4.51)$$

$$t_j - t_i - p_i - \delta_i r_{ij} - M_1 y_{ij} \geq -M_1 \quad \forall (i, j) \in E \quad (4.52)$$

$$f_{ijk} - l_{ijk} y_{ij} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (4.53)$$

$$\sum_{j=1}^{n+1} f_{0jk} = R_k \quad \forall k \in \mathcal{R} \quad (4.54)$$

$$\sum_{j=1}^{n+1} f_{ijk} - x_{ik} = \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (4.55)$$

$$\sum_{i=0}^n f_{ijk} - x_{jk} = \underline{q}_{jk} \quad j = 1, \dots, n, \forall k \in \mathcal{R} \quad (4.56)$$

$$\sum_{i=0}^n f_{in+1k} = R_k \quad \forall k \in \mathcal{R} \quad (4.57)$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall (i, j) \in E - \{E_0 \cup E_1\} \quad (4.58)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (4.59)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (4.60)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (4.61)$$

$$t_i \geq 0 \quad i = 0, 1, \dots, n+1 \quad (4.62)$$

$$0 \leq x_{ik} \leq \bar{q}_{ik} - \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (4.63)$$

$$f_{ijk} \geq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (4.64)$$

onde $t_i, n, t_{n+1}, \underline{q}_{ik}, \bar{q}_{ik}, p_i, y_{ij}, f_{ijk}, x_{ik}, l_{ijk}, R, E_0, E_1, \delta_i, r_{ij}, x_{ik}, w_k^i$ e M_1 são como definidos no modelo da seção 4.4 e:

- $\mathcal{R} = \{1, \dots, \bar{k}\}$: denota o conjunto dos diferentes tipos de recursos renováveis.
- R_k : denota a quantidade disponível do k -ésimo recurso.

Note que, de acordo com as restrições (4.51), o tempo de processamento de cada atividade do projeto agora é definido como uma função linear da quantidade de cada tipo de recurso alocada e o valor ótimo de p_i será fornecido pelo segmento de reta apropriado, considerando os tipos de recursos utilizados na execução de cada atividade.

Observe que a definição do valor de M_1 leva em consideração agora os tipos de recursos. Dessa forma, $\bar{p}_i = \max_k \bar{p}_{ik}$ representa o maior tempo de execução da atividade i entre todos os tempos máximos de execução de cada atividade (quando elas são executadas com a quantidade mínima permitida do k -ésimo recurso).

O mesmo ocorre para o valor de M_2 que passa a ser definido como $M_2 = l_{ijk}$ onde, $l_{ijk} = l_{jik} = \min\{\bar{q}_{ik}, \bar{q}_{jk}\}$, ou seja, l_{ijk} representa o menor entre os limites superiores para os fluxos f_{ijk} e f_{jik} .

4.10 Formulação do PSPRR como um problema de Programação Linear Inteira Mista com vários tipos de recursos discretos

Embora a formulação acima tenha sido desenvolvida para problemas com vários tipos de recursos contínuos (podendo assumir qualquer valor real dentro dos intervalos de validade), ela pode também ser utilizada para obter soluções relaxadas de problemas com vários tipos de recursos discretos (podendo assumir apenas valores inteiros).

Para isso, similarmente ao descrito na seção 4.7, deve-se inicialmente modelar o problema considerando o tempo de execução de cada atividade como uma função contínua de cada tipo de recurso utilizado. Em seguida, cada função deve ser linearizada através de vários

segmentos de reta. Esse procedimento é feito para cada atividade associada a cada tipo de recurso utilizado na sua execução.

Assim, a formulação originalmente proposta também pode ser estendida para o tratamento de problemas com vários tipos de recursos, tanto contínuos como discretos.

4.11 Conclusão

Este capítulo abordou a solução do Problema de Sequenciamento em Projeto com Restrição de Recurso (PSPRR) não linear que considera como variáveis contínuas tanto o tempo de execução das atividades quanto as quantidades de recurso alocadas.

Aproximações por múltiplos segmentos lineares foram utilizadas para representar o tempo de processamento de cada atividade como função linear da quantidade de recurso alocada. Com auxílio de desigualdades lineares, estas aproximações tornaram possível reformular o problema original como um problema de Programação Linear Inteira Mista.

Embora o procedimento tenha sido desenvolvido para recursos contínuos, ele pode também ser aplicado para obter soluções relaxadas de problemas com recursos discretos. Neste caso, a solução pode ser facilmente ajustada a posteriori, se necessário, para resgatar a característica discreta dos recursos. A relaxação do problema reduz substancialmente o número de variáveis inteiras, o que pode ser uma grande vantagem do ponto de vista computacional.

A aproximação por segmentos de reta foi proposta para relações entre o tempo de processamento e a quantidade de recurso alocada do tipo hiperbólica ($p_{ik} = \alpha_{ik}/q_{ik}$). No entanto, o procedimento é perfeitamente aplicável a outras formas de não linearidade do tipo:

$$p_{ik} = \phi_{ik}(q_{ik})$$

onde $\phi_{ik}(\cdot)$ pode ser qualquer função contínua, monotônica e convexa no domínio $q_{ik} \in [\underline{q}_{ik}, \bar{q}_{ik}]$.

Capítulo 5

Aplicação do Método de Decomposição de Benders ao problema estudado

5.1 Introdução

A formulação proposta para o Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR), no Capítulo 4, evidencia a relação não linear entre o tempo de processamento de cada atividade do projeto e a quantidade de recurso alocada para sua execução. A linearização desta relação, também proposta no Capítulo 4, considera o número de segmentos de reta definido conforme a precisão necessária do projeto. Entende-se que esta é uma contribuição importante, não só para a formulação do problema estudado, como também para outros problemas de sequenciamento. Assim, a definição do tempo de processamento como função linear da quantidade de recurso alocada para cada atividade pode tornar a modelagem e a resolução de problemas práticos mais realistas.

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 Problemas de Sequenciamento em Projeto com Restrição de Recursos (PSPRR), podem ser explorados com a utilização de procedimentos

computacionais de resolução (Luenberger [44]). Tais procedimentos englobam os métodos matemáticos de decomposição. Como já conhecido da literatura, de acordo com Lasdon [43] e Minoux [46], 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 melhor desempenho computacional na obtenção da solução de alguns problemas, como pode ser verificado no trabalho pioneiro de Geoffrion [27] e, por exemplo, nos trabalhos de Carvalho [20] e Camargo *et al.* [19].

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.

Dando continuidade à pesquisa, este Capítulo propõe a aplicação do Método de Decomposição de Benders ao PSPRR. Este Método também será implementado computacionalmente e utilizado na resolução de exemplos.

5.2 O Método de Decomposição de Benders aplicado ao Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR)

Como apresentado no Capítulo 4, seja o problema (\mathcal{P}) definido como:

$$\text{minimizar } t_{n+1} \quad (5.1)$$

$$\text{sujeito a } t_{n+1} - t_i - p_i - \delta_i \geq 0 \quad i = 1, \dots, n \quad (5.2)$$

$$p_i \geq p_{iw^i k} - a_{iw^i k} x_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.3)$$

$$t_j - t_i - p_i - \delta_i r_{ij} - M_1 y_{ij} \geq -M_1 \quad \forall (i, j) \in E \quad (5.4)$$

$$f_{ijk} - l_{ijk} y_{ij} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.5)$$

$$\sum_{j=1}^{n+1} f_{0jk} = R_k \quad \forall k \in \mathcal{R} \quad (5.6)$$

$$\sum_{j=1}^{n+1} f_{ijk} - x_{ik} = \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.7)$$

$$\sum_{i=0}^n f_{ijk} - x_{jk} = \underline{q}_{jk} \quad j = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.8)$$

$$\sum_{i=0}^n f_{in+1k} = R_k \quad \forall k \in \mathcal{R} \quad (5.9)$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall (i, j) \in E - \{E_0 \cup E_1\} \quad (5.10)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (5.11)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (5.12)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (5.13)$$

$$t_i \geq 0 \quad i = 0, 1, \dots, n+1 \quad (5.14)$$

$$0 \leq x_{ik} \leq \bar{q}_{ik} - \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.15)$$

$$f_{ijk} \geq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.16)$$

Considere o poliedro $\mathcal{X} = \{(x, p, f, t)\}$ que satisfaz as restrições (5.2), (5.3), (5.6), (5.7), (5.8), (5.9), (5.14), (5.15) e (5.16), isto é:

$$\begin{aligned} \mathcal{X} = \{(x, P, f, t)\} = \{ & \\ & t_{n+1} - t_i - p_i - \delta_i \geq 0 && i = 1, \dots, n \\ & p_i \geq p_{iw^i k} - a_{iw^i k} x_{ik} && i = 1, \dots, n, \forall k \in \mathcal{R} \\ & \sum_{j=1}^{n+1} f_{0jk} = R_k && \forall k \in \mathcal{R} \\ & \sum_{j=1}^{n+1} f_{ijk} - x_{ik} = \underline{q}_{ik} && i = 1, \dots, n, \forall k \in \mathcal{R} \\ & \sum_{i=0}^n f_{ijk} - x_{jk} = \underline{q}_{jk} && j = 1, \dots, n, \forall k \in \mathcal{R} \\ & \sum_{i=0}^n f_{in+1k} = R_k && \forall k \in \mathcal{R} \\ & t_i \geq 0 && i = 0, 1, \dots, n+1 \\ & 0 \leq x_{ik} \leq \bar{q}_{ik} - \underline{q}_{ik} && i = 1, \dots, n, \forall k \in \mathcal{R} \\ & f_{ijk} \geq 0 && \forall (i, j) \in E, \forall k \in \mathcal{R} \\ & \} \end{aligned}$$

Observe que a definição deste poliedro não inclui as variáveis binárias, y_{ij} . Assim, o problema (\mathcal{P}) pode então ser reescrito como:

$$\min_{(x, p, f, t) \in \mathcal{X}} t_{n+1} \quad (5.17)$$

$$\text{sujeito a } t_j - t_i - p_i - \delta_i r_{ij} - M_1 y_{ij} \geq -M_1 \quad \forall (i, j) \in E \quad (5.18)$$

$$f_{ijk} - l_{ijk} y_{ij} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.19)$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall (i, j) \in E - \{E_0 \cup E_1\} \quad (5.20)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (5.21)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (5.22)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (5.23)$$

Considere agora um conjunto de vetores definido como:

$$\mathcal{Y} = \left\{ \begin{array}{ll} y_{ij} + y_{ji} \leq 1 & \forall (i, j) \in E - \{E_0 \cup E_1\} \\ y_{ij} + y_{ji} = 1 & \forall (i, j) \in E_0 \cup E_1 \\ y_{ij} = 1 & \forall (i, j) \in E_1 \\ y_{ij} \in \{0, 1\} & \forall (i, j) \in E \end{array} \right\}$$

O problema (\mathcal{P}) fica então:

$$\min_{(x,p,f,t) \in \mathcal{X}} t_{n+1} \quad (5.24)$$

$$\text{sujeito a } t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - y_{ij}) \leq 0 \quad \forall (i, j) \in E \quad (5.25)$$

$$f_{ijk} - l_{ijk} y_{ij} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.26)$$

$$y_{ij} \in \mathcal{Y} \quad (5.27)$$

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

$$Q(\bar{y}) = \min_{(x,p,f,t) \in \mathcal{X}} t_{n+1} \quad (5.28)$$

$$\text{sujeito a } t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - \bar{y}_{ij}) \leq 0 \quad \forall (i, j) \in E \quad (5.29)$$

$$f_{ijk} - l_{ijk} \bar{y}_{ij} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.30)$$

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

Dessa forma, de acordo com Geoffrion [27], utilizando o conceito de projeção, o problema completo de minimização (\mathcal{P}) parametrizado pelas variáveis y é equivalente ao problema (\mathcal{P}') definido como:

$$\min Q(y) \quad \text{sujeito a } y \in \mathcal{Y} \cap \mathcal{V}$$

onde \mathcal{V} é dado por:

$$\mathcal{V} = \{y : \exists (x, p, f, t) \in \mathcal{X}, t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - y_{ij}) \leq 0 \quad \forall (i, j) \in E, f_{ijk} - l_{ijk} y_{ij} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R}\}$$

A teoria da dualidade, ver [27], mostra que \mathcal{V} pode ser descrito da seguinte forma:

$$\mathcal{V} = \{y : \min_{(x,p,f,t) \in \mathcal{X}} \{ \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - y_{ij})] \cdot v_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} [f_{ijk} - l_{ijk} y_{ij}] \cdot v_{2ijk} \} \leq 0\}$$

Onde, $v_{1ij} \geq 0$ e $v_{2ijk} \geq 0$, são os raios extremos do politopo definido pelo problema.

Dessa forma, para garantir a viabilidade do problema (\mathcal{P}), todo o valor de y deve satisfazer a condição:

$$\min_{(x,p,f,t) \in \mathcal{X}} \left\{ \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - y_{ij})] \cdot v_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} [f_{ijk} - l_{ijk} y_{ij}] \cdot v_{2ijk} \right\} \leq 0$$

Geoffrion [27] também afirma que técnicas de dualização podem ser aplicadas em $Q(y)$, desde que sua função objetivo seja convexa e possua restrições lineares, como é o caso. Dessa forma, $Q(\bar{y})$ passa a ser denominado de subproblema dual e pode ser representado como:

$$Q(\bar{y}) = \max_{u \geq 0} \left\{ \min_{(x,p,f,t) \in \mathcal{X}} \left\{ t_{n+1} + \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - \bar{y}_{ij})] \cdot u_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} [f_{ijk} - l_{ijk} \bar{y}_{ij}] \cdot u_{2ijk} \right\} \right\}$$

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

Assumindo que os valores de y devem satisfazer à condição de viabilidade e adicionando a variável auxiliar y_0 , o problema inicial geral passa a ser equivalente ao problema linear inteiro misto denominado de problema mestre:

$$\min_{y \in \mathcal{Y}} y_0 \tag{5.31}$$

sujeito a

$$\min_{(x,p,f,t) \in \mathcal{X}} \left\{ t_{n+1} + \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - y_{ij})] \cdot \bar{u}_{1ij}^c + \sum_{(i,j) \in E'} \sum_{k \in \mathcal{R}} [f_{ijk} - l_{ijk} y_{ij}] \cdot \bar{u}_{2ijk}^c \right\} \leq y_0 \tag{5.32}$$

$$\min_{(x,p,f,t) \in \mathcal{X}} \left\{ \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - M_1(1 - y_{ij})] \cdot \bar{v}_{1ij}^s + \sum_{(i,j) \in E'} \sum_{k \in \mathcal{R}} [f_{ijk} - l_{ijk} y_{ij}] \cdot \bar{v}_{2ijk}^s \right\} \leq 0 \tag{5.33}$$

$$y_0 \geq 0 \tag{5.34}$$

Para $(c = 1, \dots, C)$ e $(s = 1, \dots, S)$. Sendo C o número máximo de restrições do tipo (5.32) e S o número máximo de restrições do tipo (5.33). C e S são valores grandes e não conhecidos a priori.

As restrições (5.32) representam o corte I de Benders, que será gerado quando o subproblema dual tiver solução ótima.

As restrições (5.33) representam o corte II de Benders, que será gerado quando o subproblema dual tiver solução ilimitada.

5.3 Interpretação do Método de Benders pela Teoria dos Grafos

Utilizando os conceitos da Teoria dos Grafos expostos na seção 2.4, define-se aqui o grafo disjuntivo $G(N, E)$ associado ao PSPRR, onde as atividades i são os vértices e os elementos de E são os arcos.

O subproblema de Benders gerado ao fixar $\bar{y} \in \mathcal{Y}$ é também representado por um grafo conjuntivo $G(N, E_{\bar{y}})$, onde todas as relações de precedência entre as atividades do projeto estão definidas.

Uma solução do problema mestre, por sua vez, estabelece a fixação das variáveis y_{ij} , e, conseqüentemente, está relacionada com a topologia do grafo. Dessa forma, ao resolver o problema mestre, um outro vetor $\bar{y} \in \mathcal{Y}$ é definido, fixando as variáveis $\bar{y}_{ij} = 1$, existe o arco (i, j) em $G(N, E_{\bar{y}})$, ou $\bar{y}_{ij} = 0$, caso contrário. Assim, uma nova topologia para o grafo é definida, estabelecendo novas relações de precedência e possibilidades de fluxo de recursos. Posteriormente, resolve-se, novamente, o subproblema até que se encontre o vetor \bar{y} ótimo e, conseqüentemente, os valores das demais variáveis.

5.4 Estudo da inviabilidade do subproblema

Supondo um problema (\mathcal{P}) que possui solução viável, inicia-se o Método fixando um vetor $\bar{y} \in \mathcal{Y}$ qualquer. Para as componentes de \bar{y} , tais que $\bar{y}_{ij} = 1$, existe o arco (i, j) que liga a atividade i à atividade j . Esse arco define a relação de precedência entre essas atividades, determinando que a atividade i deve ser executada antes da atividade j , o que permite que recursos utilizados na execução da atividade i podem ser utilizados na execução da atividade j . Similarmente, para os valores de $\bar{y}_{ij} = 0$, não existe relação de precedência entre a atividade i e a atividade j e, conseqüentemente, não se pode garantir que as atividades não serão executadas simultaneamente, o que impossibilita a reutilização dos recursos de i em j e torna nulo o fluxo de recurso entre elas.

Portanto, com \bar{y} fixado, todas as relações de precedência entre as atividades estão estabelecidas, definindo todas as possibilidades de fluxo de recurso entre elas. Com isso, a resolução do subproblema consiste em definir as quantidades de recurso que irão fluir entre as atividades, de tal maneira que a data de término da atividade $n + 1$ seja minimizada.

Se o subproblema for viável, para \bar{y} , tem-se em sua solução ótima, a data de início de execução, a quantidade de recurso alocada e o tempo de processamento para cada atividade do projeto.

Por outro lado, se (\mathcal{P}) possui solução viável, então a inviabilidade de um subproblema está necessariamente relacionada aos valores fixados de \bar{y} . Casos de inviabilidade podem estar associados a:

1. Existência de circuito no grafo conjuntivo que representa uma determinada solução do subproblema:

Considere, por exemplo, três atividades 1, 2 e 3 com os respectivos δ iguais a zero e que utilizam um mesmo tipo de recurso. Considere também os arcos $(1, 2)$, $(2, 1)$, $(1, 3)$, $(3, 1)$, $(2, 3)$, $(3, 2) \in E$. Se \bar{y} for fixado, tal que $\bar{y}_{12} = 1$, $\bar{y}_{21} = 0$, $\bar{y}_{13} = 0$, $\bar{y}_{31} = 1$, $\bar{y}_{23} = 1$, $\bar{y}_{32} = 0$, tem-se, pelas restrições do tipo (5.4), que:

$$t_2 \geq t_1 + p_1, \quad t_3 \geq t_2 + p_2, \quad t_1 \geq t_3 + p_3.$$

Isso significa que $t_3 \geq t_2 \geq t_1 \geq t_3$ e implica, portanto, na existência de circuito

no grafo conjuntivo. Consequentemente, essa condição só será válida se $p_1 = p_2 = p_3 = 0$, ou seja, quando as atividades não forem executadas. O que corresponde a um caso de inviabilidade.

2. Fluxo insuficiente de recurso para atender à necessidade de uma atividade:

Considere três atividades 1, 2 e 3, que utilizam um mesmo tipo de recurso k . Considere também os arcos $(1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2) \in E$. Ao fixar $\bar{y}_{12} = 0, \bar{y}_{21} = 0, \bar{y}_{13} = 0, \bar{y}_{31} = 0, \bar{y}_{23} = 0$ e $\bar{y}_{32} = 0$, tem-se, pelas restrições do tipo (5.5), que:

$$f_{ijk} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R}$$

Entretanto, pelas restrições (5.16):

$$f_{ijk} \geq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R}$$

Logo:

$$f_{ijk} = 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R}$$

Dessa forma, não existe fluxo de recurso chegando nas atividades 1, 2 e 3.

Entretanto, pelas restrições (5.7) e (5.8), respectivamente:

$$\sum_{j=1}^{n+1} f_{ijk} - x_{ik} = \underline{q}_{ik} \quad \forall i = 1, \dots, n, \forall k \in \mathcal{R},$$

$$\sum_{i=0}^n f_{ijk} - x_{jk} = \underline{q}_{jk} \quad \forall j = 1, \dots, n, \forall k \in \mathcal{R},$$

para cada atividade ser executada, uma quantidade mínima do recurso k deve ser utilizada.

Neste caso, os arcos fixados através das variáveis binárias y pelo problema mestre não possibilitam o fluxo suficiente de recurso para atender à necessidade de uma atividade, ou seja, a quantidade de recurso que pode chegar em uma atividade é menor do que a quantidade mínima de recurso necessária para executá-la, então o subproblema é inviável.

5.5 Reescrevendo o Problema com uma variável de folga

Considerando um problema viável, a inviabilidade de qualquer subproblema, gerado pela fixação de um vetor \bar{y} , pode ser contornada introduzindo uma variável artificial em determinadas restrições do modelo e penalizando-a na função objetivo, isto é, redefinindo o problema (\mathcal{P}) como:

$$\min \quad t_{n+1} + M_3 v_a \quad (5.35)$$

$$\text{s. a} \quad t_{n+1} - t_i - p_i - \delta_i \geq 0 \quad i = 1, \dots, n \quad (5.36)$$

$$p_i \geq p_{iw^i k} - a_{iw^i k} x_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.37)$$

$$t_j - t_i - p_i - \delta_i r_{ij} - M_1 y_{ij} + v_a \geq -M_1 \quad \forall (i, j) \in E \quad (5.38)$$

$$f_{ijk} - l_{ijk} y_{ij} - v_a \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.39)$$

$$\sum_{j=1}^{n+1} f_{0jk} = R_k \quad \forall k \in \mathcal{R} \quad (5.40)$$

$$\sum_{j=1}^{n+1} f_{ijk} - x_{ik} = \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.41)$$

$$\sum_{i=0}^n f_{ijk} - x_{jk} = \underline{q}_{jk} \quad j = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.42)$$

$$\sum_{i=0}^n f_{in+1k} = R_k \quad \forall k \in \mathcal{R} \quad (5.43)$$

$$y_{ij} + y_{ji} \leq 1 \quad \forall (i, j) \in E - \{E_0 \cup E_1\} \quad (5.44)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in E_0 \cup E_1 \quad (5.45)$$

$$y_{ij} = 1 \quad \forall (i, j) \in E_1 \quad (5.46)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (5.47)$$

$$t_i \geq 0 \quad i = 0, 1, \dots, n+1 \quad (5.48)$$

$$0 \leq x_{ik} \leq \bar{q}_{ik} - \underline{q}_{ik} \quad i = 1, \dots, n, \forall k \in \mathcal{R} \quad (5.49)$$

$$f_{ijk} \geq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.50)$$

$$v_a \geq 0 \quad (5.51)$$

Onde, v_a é uma variável artificial e M_3 é um número grande positivo.

A introdução dessa variável no problema, segundo Jensen [35], garante a viabilidade do

subproblema. Isso ocorre devido à geração de uma nova restrição no subproblema dual que limita o valor das variáveis duais, de modo que o somatório de todas essas variáveis seja menor ou igual ao valor de M_3 .

Observe que a variável artificial foi utilizada somente nas restrições (5.38) e (5.39) e que a função objetivo também foi modificada para impor uma penalidade no valor da solução.

Observe também que se o problema original sem as variáveis artificiais possui uma solução viável então, para um valor de M_3 suficientemente grande, o valor da variável de folga é igual a zero.

Dessa forma, o subproblema terá solução limitada e será escrito como:

$$Q(\bar{y}) = \min_{(x,p,f,t) \in \mathcal{X}} t_{n+1} + M_3 v_a \quad (5.52)$$

$$\text{sujeito a } t_i - t_j + p_i + \delta_i r_{ij} - v_a - M_1(1 - \bar{y}_{ij}) \leq 0 \quad \forall (i, j) \in E \quad (5.53)$$

$$f_{ijk} - v_a - l_{ijk} \bar{y}_{ij} \leq 0 \quad \forall (i, j) \in E, \forall k \in \mathcal{R} \quad (5.54)$$

Com isso, o problema mestre passa a ser constituído apenas pelo corte tipo I:

$$\min_{y \in \mathcal{Y}} y_0 \quad (5.55)$$

sujeito a

$$\min_{(x,p,f,t) \in \mathcal{X}} \{t_{n+1} + M_3 v_a + \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - v_a - M_1(1 - y_{ij})] \cdot \bar{u}_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} [f_{ijk} - v_a - l_{ijk} y_{ij}] \cdot \bar{u}_{2ijk}\} \leq y_0 \quad (5.56)$$

$$y_0 \geq 0 \quad (5.57)$$

5.5.1 Reescrevendo o Corte I

Analisando o corte I:

$$\min_{(x,p,f,t) \in \mathcal{X}} \{t_{n+1} + M_3 v_a + \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - v_a - M_1(1 - y_{ij})] \cdot \bar{u}_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} [f_{ijk} - v_a - l_{ijk} y_{ij}] \cdot \bar{u}_{2ijk}\} \leq y_0,$$

observa-se que a função objetivo é independente das variáveis y e que as restrições $t_i - t_j + p_i + \delta_i r_{ij} - v_a - M_1(1 - y_{ij}) \leq 0$ e $f_{ijk} - v_a - l_{ijk} y_{ij} \leq 0$ são linearmente separáveis em y .

Dessa forma, esse corte pode ser reescrito como:

$$\min_{(x,p,f,t) \in \mathcal{X}} \{t_{n+1} + M_3 v_a + \sum_{(i,j) \in E} (t_i - t_j + p_i + \delta_i r_{ij} - v_a) \cdot \bar{u}_{1ij} - M_1 \sum_{(i,j) \in E} (1 - y_{ij}) \cdot \bar{u}_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} (f_{ijk} - v_a) \cdot \bar{u}_{2ijk} - l_{ijk} \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} y_{ij} \cdot \bar{u}_{2ijk}\} \leq y_0$$

$$\min_{(x,p,f,t) \in \mathcal{X}} \{t_{n+1} + M_3 v_a + \sum_{(i,j) \in E} (t_i - t_j + p_i + \delta_i r_{ij} - v_a) \cdot \bar{u}_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} (f_{ijk} - v_a) \cdot \bar{u}_{2ijk} - M_1 \sum_{(i,j) \in E} (1 - y_{ij}) \cdot \bar{u}_{1ij} - l_{ijk} \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} y_{ij} \cdot \bar{u}_{2ijk}\} \leq y_0$$

O subproblema dual:

$$Q(\bar{y}) = \max_{u \geq 0} \{ \min_{(x,p,f,t) \in \mathcal{X}} \{t_{n+1} + M_3 v_a + \sum_{(i,j) \in E} [t_i - t_j + p_i + \delta_i r_{ij} - v_a - M_1(1 - \bar{y}_{ij})] \cdot u_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} [f_{ijk} - v_a - l_{ijk} \bar{y}_{ij}] \cdot u_{2ijk}\} \},$$

também pode ser reescrito como:

$$Q(\bar{y}) = \max_{u \geq 0} \{ [-M_1 \sum_{(i,j) \in E} (1 - \bar{y}_{ij})] \cdot u_{1ij} - [l_{ijk} \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} \bar{y}_{ij}] \cdot u_{2ijk} + \min_{(x,p,f,t) \in \mathcal{X}} \{t_{n+1} + M_3 v_a + \sum_{(i,j) \in E} (t_i - t_j + p_i + \delta_i r_{ij} - v_a) \cdot u_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} (f_{ijk} - v_a) \cdot u_{2ijk}\} \}$$

Considerando uma iteração qualquer do algoritmo de Benders, o vetor de multiplicadores ótimos \bar{u} é obtido resolvendo-se o subproblema $Q(\bar{y})$ com \bar{y} fixado. Quando $u = \bar{u}$, o valor ótimo para $Q(\bar{y})$ é dado por:

$$Q(\bar{y}) = -M_1 \sum_{(i,j) \in E} (1 - \bar{y}_{ij}) \cdot \bar{u}_{1ij} - l_{ijk} \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} \bar{y}_{ij} \cdot \bar{u}_{2ijk} + \min_{(x,p,f,t) \in \mathcal{X}} \{t_{n+1} + M_3 v_a + \sum_{(i,j) \in E} (t_i - t_j + p_i + \delta_i r_{ij} - v_a) \cdot \bar{u}_{1ij} + \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} (f_{ijk} - v_a) \cdot \bar{u}_{2ijk}\}$$

Utilizando a equação $Q(\bar{y})$ no corte I, obtém-se a expressão:

$$Q(\bar{y}) + M_1 \sum_{(i,j) \in E} (y_{ij} - \bar{y}_{ij}) \cdot \bar{u}_{1ij} + l_{ijk} \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} (\bar{y}_{ij} - y_{ij}) \cdot \bar{u}_{2ijk} \leq y_0$$

O corte I é reescrito agora em função de $Q(\bar{y})$ e das variáveis duais. Os valores das variáveis duais associadas às restrições do subproblema são geradas a partir de \bar{y} .

Considere que um ciclo do Método de Benders consiste na resolução de um subproblema seguida da resolução de um problema mestre relaxado. Assim, o problema mestre relaxado, após C ciclos do Método, passa a ser escrito como:

$$\min_{y \in \mathcal{Y}} y_0 \quad (5.58)$$

sujeito a

$$Q(\bar{y}) + M_1 \sum_{(i,j) \in E'} (y_{ij} - \bar{y}_{ij}) \cdot \bar{u}_{1ij}^c + l_{ijk} \sum_{(i,j) \in E} \sum_{k \in \mathcal{R}} (\bar{y}_{ij} - y_{ij}) \cdot \bar{u}_{2ijk}^c \leq y_0 \quad c = 1, \dots, C \quad (5.59)$$

$$y_0 \geq 0 \quad (5.60)$$

5.5.2 Reescrevendo o Corte II

Conhecendo as componentes de \bar{y} fixadas que geram um circuito no grafo que representa uma solução de um subproblema, pode-se eliminar tal circuito introduzindo no problema mestre a restrição:

$$\sum_{(i,j) \in \mu} y_{ij} \leq |\mu| - 1,$$

onde $y_{ij} = 1$, se o arco $(i, j) \in \mu$ e $|\mu|$ é o número de arcos que pertencem ao circuito μ .

Assim, observando que o corte I independe da variável artificial v_a , pode-se então eliminá-la da restrição (5.38), conservando-a somente na restrição (5.39). Feito isso, a inviabilidade associada a uma solução do subproblema está necessariamente associada à existência de um circuito no grafo.

Dessa forma, o problema mestre relaxado passa a ser escrito como:

$$\min_{y \in \mathcal{Y}} y_0 \quad (5.61)$$

sujeito a

$$Q(\bar{y}) - M_1 \sum_{(i,j) \in E'} (1 - y_{ij}) \cdot \bar{u}_{1ij}^c + l_{ijk} \sum_{(i,j) \in E'} \sum_{k \in \mathcal{R}} (1 - y_{ij}) \cdot \bar{u}_{2ijk}^c \leq y_0 \quad c = 1, \dots, C \quad (5.62)$$

$$\sum_{(i,j) \in \mu} y_{ij} \leq |\mu| - 1 \quad (5.63)$$

$$y_0 \geq 0 \quad (5.64)$$

5.6 Testes comparativos com a implementação do Método de Decomposição de Benders

Para analisar o desempenho do algoritmo proposto neste Capítulo 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. 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 5.6.1 foram propostos pela autora. Já os problemas testes da seção 5.6.2 foram gerados pelo programa Progen (Kolisch e Sprecher [40]), que é um gerador de problemas testes utilizado na literatura para o PSPRR, dentre outros.

5.6.1 Testes comparativos I

Foram criados três grupos de problemas: P01, P02 e P03. 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;
- a quantidade total disponível do recurso varia de forma aleatória;
- o número de atividades é igual à 7, 12 e 17 para os exemplos dos grupos 01, 02 e 03, respectivamente;
- os demais dados, como o tempo de execução e a quantidade de recurso alocada, variam de um exemplo para outro, de forma aleatória.

Note que os parâmetros de entrada de cada um dos cinco exemplos de cada grupo foram criados aleatoriamente, sem nenhum critério estabelecido para diferenciar uma instância da outra, o que influenciou diretamente no tempo computacional para obtenção da solução ótima.

Os exemplos de cada um dos grupos, P01, P02 e P03, são representados pelos grafos correspondentes das figuras 5.1, 5.2 e 5.3:

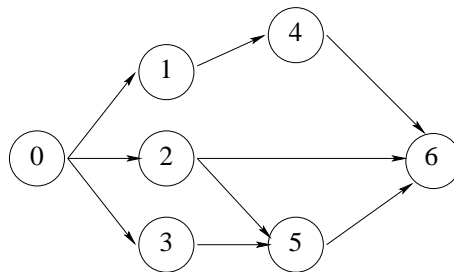


Figura 5.1: Grafo dos exemplos do grupo P01

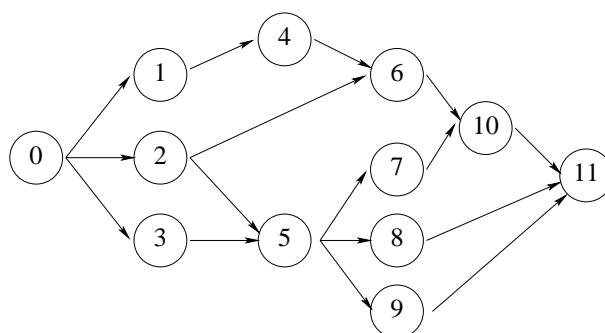


Figura 5.2: Grafo dos exemplos do grupo P02

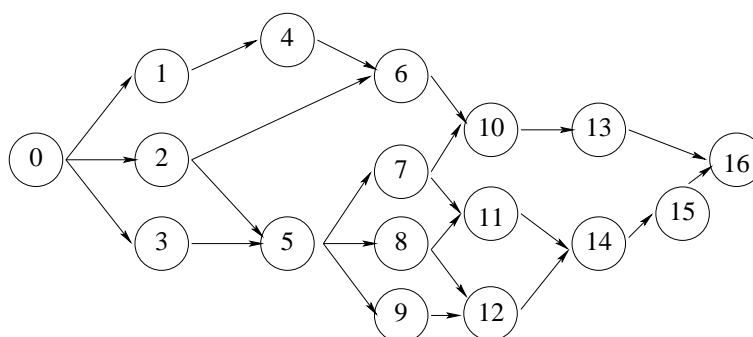


Figura 5.3: Grafo dos exemplos do grupo P03

Os resultados computacionais de cada grupo, utilizando o *solver* CPLEX 9.1, estão resumidos nas tabelas 5.1, 5.2 e 5.3.

Problemas	Solução Ótima	Tempo CPU [s]	Num. Var. y_{ij}	Num. Rest. / Num. Var.	Num. Iter.
P01 ^a	10	0,02	14	116/93	276
P01 ^b	16	0,02	14	116/93	444
P01 ^c	9	0,02	14	116/93	228
P01 ^d	7	0,01	14	116/93	48
P01 ^e	20	0,03	14	116/93	959

Tabela 5.1: Resultados computacionais do P01 - CPLEX

Problemas	Solução Ótima	Tempo CPU [s]	Num. Var. y_{ij}	Num. Rest. / Num. Var.	Num. Iter.
P02 ^a	16	0,01	46	352/278	90
P02 ^b	21	0,02	46	352/278	114
P02 ^c	16	0,01	46	352/278	129
P02 ^d	17	0,01	46	352/278	91
P02 ^e	20	0,01	46	352/278	114

Tabela 5.2: Resultados computacionais do P02 - CPLEX

Problemas	Solução Ótima	Tempo CPU [s]	Num. Var. y_{ij}	Num. Rest. / Num. Var.	Num. Iter.
P03 ^a	26	0,04	96	722/563	189
P03 ^b	26	0,03	96	722/563	118
P03 ^c	26	0,01	96	722/563	65
P03 ^d	30	0,03	96	722/563	116
P03 ^e	30	0,01	96	722/563	58

Tabela 5.3: Resultados computacionais do P03 - CPLEX

As variáveis y_{ij} , nas tabelas acima, 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.

A seguir, aplicando-se o Método de Decomposição de Benders, as tabelas 5.4, 5.5 e 5.6 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 à zero, para y_1 , todos iguais à um, para y_2 , os valores são aleatórios e para y^* , os valores correspondem aos da solução ótima.

Problemas	Tempo Total CPU [s]	Tempo Total Sub [s]	Tempo Total Mestre [s]	Num. Ciclos Para Obter Valor Sol. Ótima	Num. Ciclos Para Otimidade
P01 _{y0} ^a	0,04	0,01	0,03	13	53
P01 _{y1} ^a	0,22	0,02	0,20	9	50
P01 _{y2} ^a	0,20	0,01	0,19	1	53
P01 _{y*} ^a	0,20	0,01	0,19	1	53
P01 _{y0} ^b	0,17	0,01	0,16	3	52
P01 _{y1} ^b	0,25	0,04	0,21	3	60
P01 _{y2} ^b	0,28	0,05	0,23	18	65
P01 _{y*} ^b	0,14	0,01	0,13	1	53
P01 _{y0} ^c	0,23	0,07	0,16	15	62
P01 _{y1} ^c	0,36	0,02	0,34	46	74
P01 _{y2} ^c	0,27	0,01	0,26	5	60
P01 _{y*} ^c	0,23	0,04	0,19	1	58
P01 _{y0} ^d	0,02	0,01	0,01	7	7
P01 _{y1} ^d	0,02	0,01	0,01	7	7
P01 _{y2} ^d	0,02	0,01	0,01	9	9
P01 _{y*} ^d	0,02	0,01	0,01	1	1
P01 _{y0} ^e	0,32	0,02	0,30	17	64
P01 _{y1} ^e	0,21	0,02	0,19	40	58
P01 _{y2} ^e	0,26	0,03	0,23	15	63
P01 _{y*} ^e	0,26	0,02	0,24	1	59

Tabela 5.4: Resultados computacionais do P01 - Benders

Problemas	Tempo Total CPU [s]	Tempo Total Sub [s]	Tempo Total Mestre [s]	Num. Ciclos Para Obter Valor Sol. Ótima	Num. Ciclos Para Otimalidade
P02 _{y0} ^a	0,89	0,13	0,76	146	148
P02 _{y1} ^a	0,94	0,13	0,81	144	144
P02 _{y2} ^a	0,95	0,13	0,82	150	151
P02 _{y*} ^a	0,45	0,11	0,34	1	109
P02 _{y0} ^b	0,31	0,11	0,20	92	92
P02 _{y1} ^b	0,64	0,13	0,51	114	116
P02 _{y2} ^b	1,08	0,13	0,95	93	165
P02 _{y*} ^b	0,39	0,11	0,28	1	119
P02 _{y0} ^c	0,03	0,01	0,02	9	9
P02 _{y1} ^c	0,03	0,01	0,02	10	12
P02 _{y2} ^c	0,03	0,01	0,02	10	10
P02 _{y*} ^c	0,20	0,02	0,18	1	45
P02 _{y0} ^d	0,02	0,01	0,01	12	12
P02 _{y1} ^d	0,02	0,01	0,01	16	16
P02 _{y2} ^d	0,02	0,01	0,01	10	10
P02 _{y*} ^d	0,02	0,01	0,01	1	19
P02 _{y0} ^e	0,03	0,01	0,02	17	17
P02 _{y1} ^e	0,07	0,02	0,05	20	28
P02 _{y2} ^e	0,09	0,01	0,08	28	28
P02 _{y*} ^e	0,03	0,01	0,02	1	19

Tabela 5.5: Resultados computacionais do P02 - Benders

Problemas	Tempo Total CPU [s]	Tempo Total Sub [s]	Tempo Total Mestre [s]	Num. Ciclos Para Obter Valor Sol. Ótima	Num. Ciclos Para Otimalidade
P03 _{y0} ^a	3,58	0,47	3,11	268	268
P03 _{y1} ^a	5,76	0,58	5,18	353	353
P03 _{y2} ^a	7,65	0,30	7,35	277	277
P03 _{y*} ^a	19,14	0,88	18,26	530	543
P03 _{y0} ^b	0,32	0,11	0,21	87	124
P03 _{y1} ^b	2,39	0,15	2,24	24	145
P03 _{y2} ^b	4,81	0,63	4,18	134	355
P03 _{y*} ^b	14,03	0,78	13,27	1	397
P03 _{y0} ^c	0,28	0,08	0,20	68	88
P03 _{y1} ^c	0,42	0,12	0,32	31	69
P03 _{y2} ^c	3,90	0,31	3,59	241	241
P03 _{y*} ^c	0,02	0,01	0,01	288	301
P03 _{y0} ^d	2,52	0,10	2,42	249	258
P03 _{y1} ^d	3,48	0,08	3,40	127	214
P03 _{y2} ^d	3,53	0,10	3,43	261	261
P03 _{y*} ^d	6,78	0,08	6,70	1	336
P03 _{y0} ^e	0,32	0,08	0,24	74	105
P03 _{y1} ^e	0,28	0,08	0,20	103	202
P03 _{y2} ^e	1,90	0,10	1,80	116	268
P03 _{y*} ^e	3,20	0,80	2,40	12	311

Tabela 5.6: Resultados computacionais do P03 - Benders

5.6.2 Testes comparativos II

Os problemas testes desta seção foram gerados pelo programa Progen (Kolisch e Sprecher [40]), que é um gerador de problemas testes amplamente utilizado na literatura para problemas de sequenciamento de atividades em projetos com recursos limitados. 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 três grupos de problemas testes: EXPL01, EXPL02 e EXPL03. 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 à 7, 12 e 17 para os exemplos dos grupos EXPL01, EXPL02 e EXPL03, respectivamente.

As tabelas 5.7, 5.8 e 5.9 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 5.7: Resultados computacionais do EXPL01

Problemas	Solução Ótima	CPLEX Tempo CPU [s]	Benders 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	3,13	Não obtido
EXPL02(7)	41,97	3,96	Não obtido
EXPL02(8)	42,03	4,85	Não obtido

Tabela 5.8: Resultados computacionais do EXPL02

Problemas	Solução Ótima	CPLEX Tempo CPU [s]	Benders Tempo CPU [s]
EXPL03(1)	43,61	2,28	Não obtido
EXPL03(2)	37,58	2,32	Não obtido
EXPL03(3)	42,22	2,80	Não obtido
EXPL03(4)	48,97	2,95	Não obtido
EXPL03(5)	48,62	3,25	Não obtido
EXPL03(6)	51,97	4,75	Não obtido
EXPL03(7)	49,31	5,23	Não obtido
EXPL03(8)	52,03	7,50	Não obtido

Tabela 5.9: Resultados computacionais do EXPL03

5.7 Conclusão

Dentro de um conjunto de problemas de pequeno e médio porte (entre 7 e 12 atividades e sem grandes dificuldades na alocação de recursos e relações de precedência), obteve-se soluções ótimas utilizando o Método de Decomposição de Benders. A partir de problemas com 12 atividades, o Método mostrou limitações significativas em comparação ao *solver* CPLEX (versão 9.1).

Além dos problemas apresentados ao longo deste Capítulo, outros testes também foram realizados aplicando-se o Método de Decomposição de Benders com algoritmo desenvolvido pela autora. Como regra geral, o processamento via Método de Decomposição de Benders se mostrou comparativamente não atrativo, exigindo tempo computacional, muitas vezes, proibitivo, ou seja, não obtendo o valor da solução ótima. Talvez, após aperfeiçoamentos mais elaborados, esta realidade possa se alterar. O que sugere a continuidade do estudo, visando aprimorar a versão apresentada deste Método de Decomposição.

Adicionalmente, outras técnicas também poderão ser acrescentadas ao método, como forma de melhorar o processo de resolução. Pois, apesar do algoritmo de Benders convergir para uma solução ótima, a definição de uma região de viabilidade pode ser acelerada.

Capítulo 6

Exemplos ilustrativos resolvidos computacionalmente

6.1 Introdução

Neste Capítulo são apresentados e resolvidos computacionalmente alguns exemplos de problemas para ilustrar a aplicação do modelo proposto. O primeiro exemplo consiste em um problema com um recurso contínuo. O segundo e o terceiro exemplos consideram a utilização de um recurso discreto, apesar de serem modelados e resolvidos como problemas com um recurso contínuo. O quarto exemplo ilustra a formulação e solução de um problema com múltiplos recursos. Finalmente, um problema com recursos múltiplos é resolvido em duas versões: a primeira sem a existência de tempo de espera e a segunda considerando a existência de tempo de espera após algumas atividades.

Todos os exemplos foram implementados e resolvidos computacionalmente utilizando-se o MPL (*Mathematical Programming Language*) e o *solver* CPLEX 9.1.

6.2 Um problema com recurso contínuo

Considere o problema de usinar um conjunto de lotes de peças metálicas utilizando máquinas operatrizes. A fabricação de cada lote de peças representa uma atividade que deve ser executada com uma potência elétrica (kW) constante, respeitando limites mínimos e máximos. Como primeira aproximação, pode-se considerar que o processo de usinagem completo de cada lote demanda um consumo fixo de energia (kWh), independente da potência adotada (kW). Assim, o tempo de usinagem de cada lote é inversamente proporcional à potência elétrica aplicada. Compartilhando a mesma fonte de energia, as atividades têm que respeitar a capacidade máxima do fornecimento (no caso uma potência máxima de $10kW$). A energia necessária para usinar cada lote e os respectivos limites de potência estão mostrados na tabela 6.1. O problema a ser resolvido é o de otimizar a sequência de usinagem dos lotes, minimizando o tempo total de realização.

Atividade	kWh	Limites de potência
1	10,5	$1,0 \leq kW \leq 7,5$
2	11,0	$2,4 \leq kW \leq 6,0$
3	11,5	$2,9 \leq kW \leq 8,2$
4	12,0	$4,1 \leq kW \leq 8,9$
5	12,5	$5,2 \leq kW \leq 10,0$
6	15,0	$1,2 \leq kW \leq 5,4$
7	14,5	$2,4 \leq kW \leq 7,2$

Tabela 6.1: Dados gerenciais do Exemplo 6.2

Como parte da modelagem do problema são introduzidas duas atividades fictícias: atividade 0 e atividade 8. Essas atividades representam, respectivamente, o início e o término do projeto. O grafo da figura 6.1 apresenta as relações de precedência entre as atividades.

Note que somente as atividades inicial e final estabelecem a priori uma relação de precedência com as demais. As outras atividades não possuem a priori nenhuma relação de precedência entre elas.

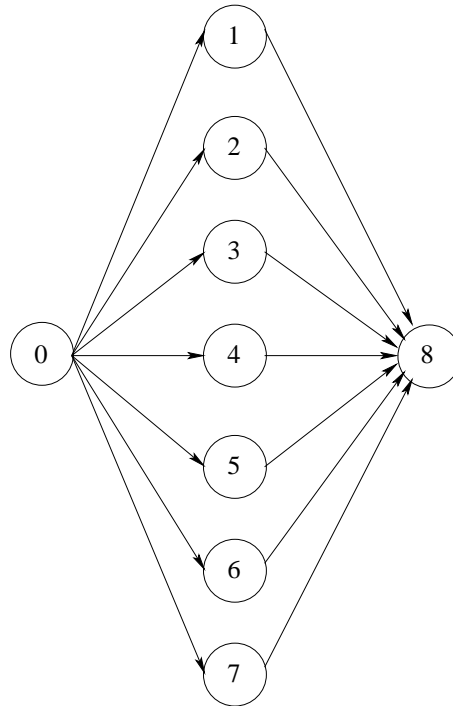


Figura 6.1: Grafo do Exemplo 6.2

A partir das quantidades mínimas (\underline{q}_i) e máximas (\bar{q}_i) de potência admissíveis para a execução de cada atividade do projeto, pode-se calcular os tempos de processamento máximos (\bar{p}_i) e mínimos (\underline{p}_i), conforme mostrado na tabela 6.2.

Atividade	R	\underline{q}_i	\bar{q}_i	\bar{p}_i	\underline{p}_i
1	10	1,0	7,5	10,5	1,4
2	10	2,4	6,0	4,6	1,8
3	10	2,9	8,2	4,0	1,4
4	10	4,1	8,9	2,9	1,3
5	10	5,2	10,0	2,4	1,3
6	10	1,2	5,4	12,5	2,8
7	10	2,4	7,2	6,0	2,0

Tabela 6.2: Dados técnicos do Exemplo 6.2

Uma solução deste problema foi obtida utilizando-se a técnica proposta de aproximação das curvas de tempo de processamento por múltiplos segmentos de reta (conforme tabela 6.3). Procurou-se segmentar cada curva de tempo de processamento com um número mínimo de intervalos suficiente para que o tamanho de cada intervalo fosse menor do que uma unidade.

Atividade	Número de segmentos	Tamanho do intervalo	t_i	p_i	x_i	$\underline{q}_i + x_i$
0	-	-	0	0	-	-
1	7	0,93	8,2	1,4	6,5	7,5
2	4	0,90	2,4	1,8	3,6	6,0
3	6	0,88	4,2	1,4	5,3	8,2
4	5	0,96	5,6	1,3	4,8	8,9
5	5	0,96	6,9	1,3	4,8	10,0
6	5	0,84	0	4,2	2,4	3,6
7	5	0,96	0	2,4	4,0	6,4
8	-	-	9,6	0	-	-

Tabela 6.3: Solução do Exemplo 6.2 pela aproximação por múltiplos segmentos de reta

Uma segunda solução foi obtida dobrando-se o número de segmentos para cada atividade, conforme mostrado na tabela 6.4. Consequentemente, o tamanho de cada intervalo passa a ser a metade do considerado na solução anterior.

Atividade	Número de segmentos	Tamanho do intervalo	t_i	p_i	x_i	$\underline{q}_i + x_i$
0	-	-	0	0	-	-
1	14	0,46	8,1	1,4	6,5	7,5
2	8	0,45	2,3	1,8	3,6	6,0
3	12	0,44	4,1	1,4	5,3	8,2
4	10	0,48	5,5	1,3	4,8	8,9
5	10	0,48	6,8	1,3	4,8	10,0
6	10	0,42	0	4,1	2,4	3,6
7	10	0,48	0	2,3	4,0	6,4
8	-	-	9,5	0	-	-

Tabela 6.4: Solução do Exemplo 6.2 pela aproximação pelo dobro de segmentos de reta

Ambas as soluções ótimas estão representadas pelos seus diagramas de Gantt como mostra as figuras 6.2 e 6.3, respectivamente.

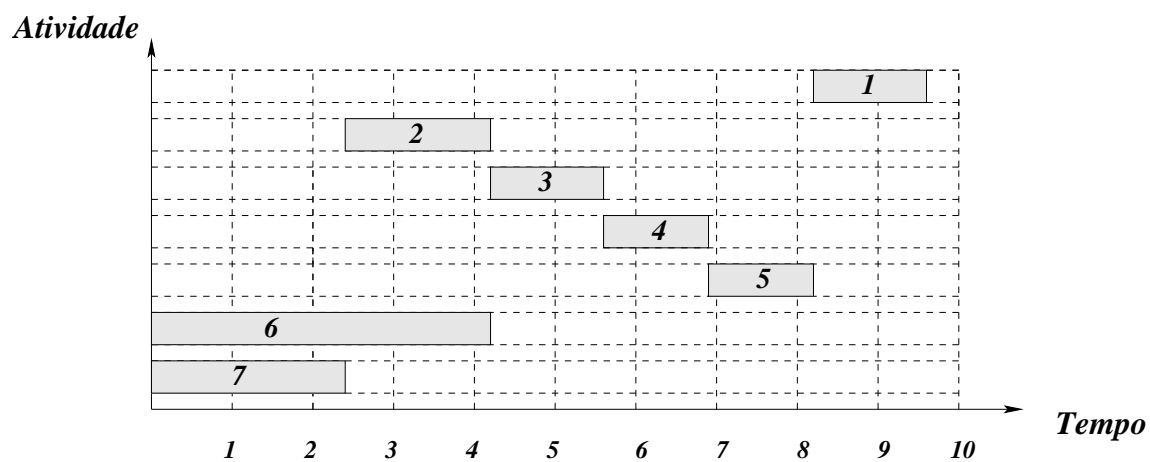


Figura 6.2: Diagrama de Gantt da solução com segmentos de reta

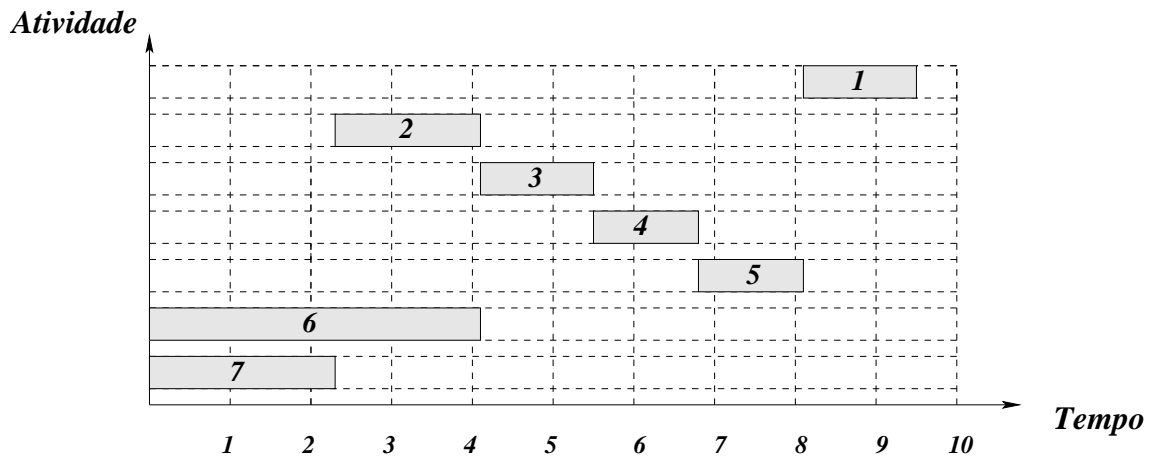


Figura 6.3: Diagrama de Gantt da solução com o dobro do número de segmentos de reta

Note que, apenas para as atividades 6 e 7, os tempos de processamento foram reduzidos em 0,1 unidades de tempo. Para as demais atividades o tempo permaneceu o mesmo. De fato, as quantidades de recurso utilizadas para executar cada uma das atividades, exceto as atividades 6 e 7, são, já na primeira solução, iguais aos valores máximos permitidos.

6.3 Um problema com recurso discreto

6.3.1 Um problema com resultado inteiro

Considere um projeto constituído da montagem de um conjunto de cinco equipamentos mecânicos. A montagem de cada equipamento corresponde a uma atividade com a necessidade de um número total de homens-hora para ser realizada (vide tabela 6.5).

De acordo com a especificidade de cada equipamento, existe um limite mínimo e máximo de número de homens que podem trabalhar simultaneamente sobre o equipamento, conforme mostra a tabela 6.5. A montagem deverá ser realizada por uma empresa que possui uma equipe de 10 trabalhadores.

Atividade	Homens-Hora	Limites de trabalhadores
1	20	$1 \leq \text{trabalhadores} \leq 7$
2	25	$2 \leq \text{trabalhadores} \leq 8$
3	25	$3 \leq \text{trabalhadores} \leq 9$
4	30	$4 \leq \text{trabalhadores} \leq 10$
5	15	$5 \leq \text{trabalhadores} \leq 11$

Tabela 6.5: Dados gerenciais do Exemplo 6.3.1

Note que a atividade 5, para ser executada, necessita no mínimo de 5 e no máximo de 11 trabalhadores. No entanto, tecnicamente, ela só poderá utilizar uma quantidade máxima de 10 trabalhadores, que é o número disponível do recurso. Dessa maneira, a quantidade máxima de trabalhadores para executar a atividade 5 é, automaticamente, substituída na modelagem por 10 trabalhadores.

Para modelar o problema são introduzidas artificialmente a atividade 0 e a atividade 6, conforme ilustra o grafo da figura 6.4. Neste exemplo, admite-se que a montagem dos equipamentos pode ser feita em qualquer ordem, sem nenhuma relação de precedência estabelecida a priori. Somente a título de ilustração, no grafo da figura 6.4, as cinco atividades estão sendo realizadas em paralelo.

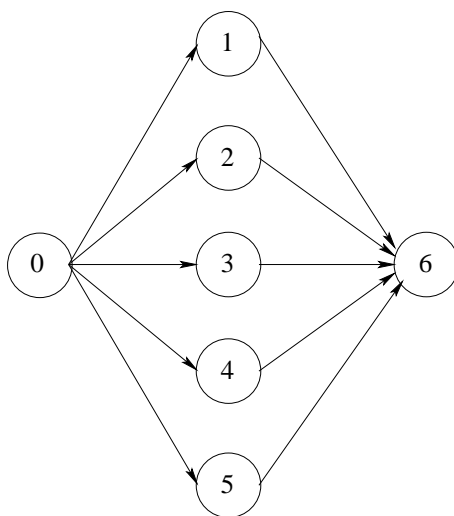


Figura 6.4: Grafo do Exemplo 6.3.1

A partir das quantidades de recurso mínimas (\underline{q}_i) e máximas (\bar{q}_i) admissíveis para a execução de cada atividade do projeto, calcula-se os tempos de processamento máximos (\bar{p}_i) e mínimos (\underline{p}_i), conforme mostrado na tabela 6.6.

Atividade	R	\underline{q}_i	\bar{q}_i	\bar{p}_i	\underline{p}_i
1	10	1	7	20,0	2,86
2	10	2	8	12,5	3,13
3	10	3	9	8,3	2,78
4	10	4	10	7,5	3,00
5	10	5	10	3,0	1,50

Tabela 6.6: Dados técnicos do Exemplo 6.3.1

Note que, como explicado anteriormente, para a atividade 5 o número máximo de trabalhadores está estabelecido como 10, correspondendo ao total de trabalhadores disponíveis.

Inicialmente, uma primeira solução foi obtida aproximando as curvas dos tempos de processamento por um único segmento de reta, cujos resultados estão apresentados na tabela 6.7.

Atividade	Número de segmentos	t_i	p_i	x_i	$\underline{q}_i + x_i$
0	-	0	0	-	-
1	1	0	2,9	6	7
2	1	0	12,5	0	2
3	1	5,0	7,5	1	4
4	1	5,0	7,5	0	4
5	1	2,9	2,1	3	8
6	-	12,5	0	-	-

Tabela 6.7: Solução do Exemplo 6.3.1 pela aproximação por um único segmento de reta

Uma segunda solução foi obtida aproximando as curvas dos tempos de processamento por múltiplos segmentos (um segmento para cada unidade de intervalo), como mostra a tabela 6.8.

Atividade	Número de segmentos	t_i	p_i	x_i	$\underline{q}_i + x_i$
0	-	0	0	-	-
1	6	0	5	3	4
2	6	6,5	5	3	5
3	6	6,5	5	2	5
4	6	0	5	2	6
5	5	5	1,5	5	10
6	-	11,5	0	-	-

Tabela 6.8: Solução do Exemplo 6.3.1 pela aproximação por múltiplos segmentos de reta

Ambas as soluções ótimas estão representadas pelos seus diagramas de Gantt como mostra as figuras 6.5 e 6.6, respectivamente.

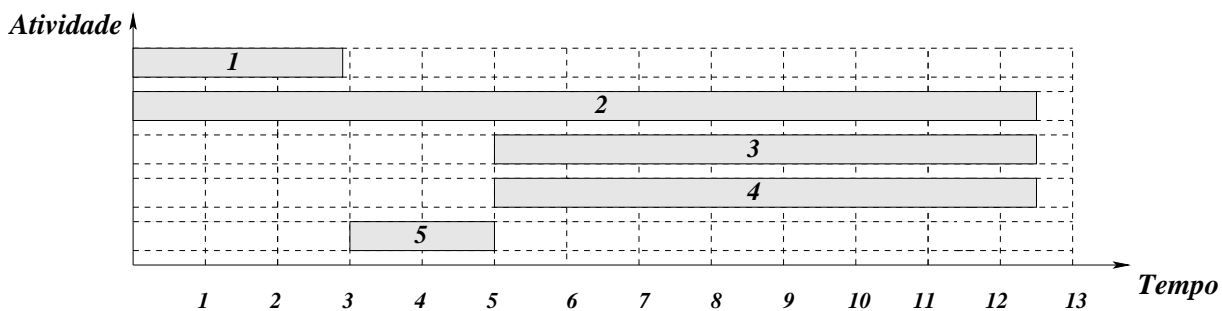


Figura 6.5: Diagrama de Gantt da solução pela aproximação com um segmento de reta

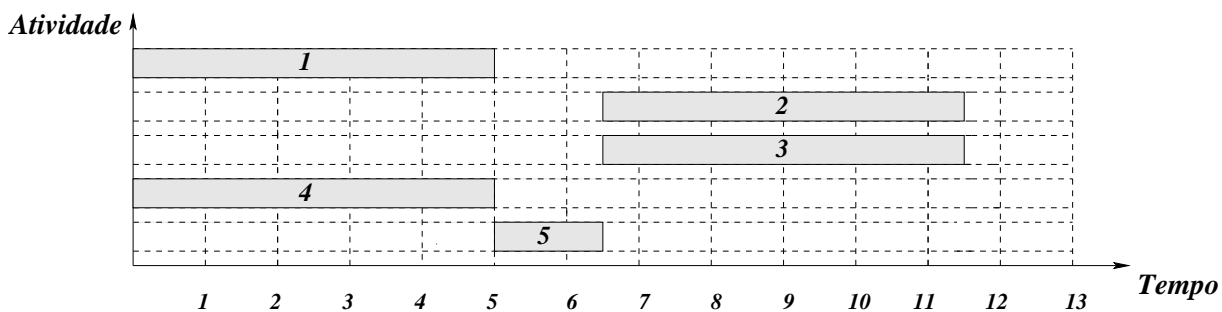


Figura 6.6: Diagrama de Gantt da solução pela aproximação com vários segmentos de reta

Note que a sequência das atividades para cada solução é bastante diferente, sendo que o tempo para realizá-las é menor para a segunda solução (12,5 para a primeira solução e 11,5 para a segunda solução).

Note também que os valores da solução obtida pela aproximação com vários segmentos, coincidentemente, estão sobre a curva exata de tempo de processamento \times quantidade de recurso. Desta forma, não é necessário realizar nenhum ajuste a posteriori nesta solução.

Observe que, apesar de ser um problema com recurso contínuo, os valores definidos para as quantidades de recurso alocadas para cada atividade são todos valores inteiros.

6.3.2 Um problema com tratamento a posteriori

Considere agora um outro projeto também constituído da montagem de um conjunto de cinco equipamentos mecânicos. Os valores da necessidade de homens-hora e dos limites mínimo e máximo de número de homens que podem trabalhar simultaneamente sobre o mesmo equipamento correspondem aqueles da tabela 6.5 do exemplo anterior que está reproduzida na tabela 6.9.

Atividade	Homens-Hora	Limites de trabalhadores
1	20	$1 \leq \text{trabalhadores} \leq 7$
2	25	$2 \leq \text{trabalhadores} \leq 8$
3	25	$3 \leq \text{trabalhadores} \leq 9$
4	30	$4 \leq \text{trabalhadores} \leq 10$
5	15	$5 \leq \text{trabalhadores} \leq 11$

Tabela 6.9: Dados gerenciais do Exemplo 6.3.2

O grafo de representação deste problema agora é definido como ilustra a figura 6.7. As atividades 4 e 5 são atividades disjuntivas que não podem ser executadas simultaneamente. A atividade 0 e a atividade 6 também são atividades fictícias que representam, respectivamente, o início e o término do projeto.

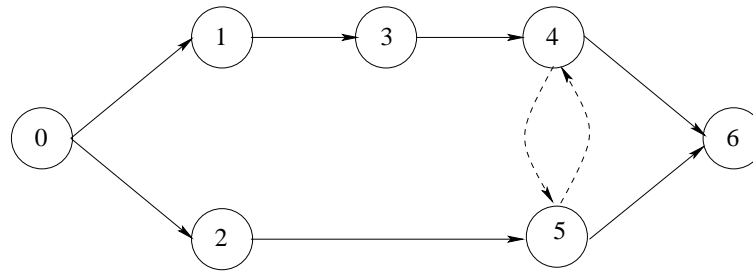


Figura 6.7: Grafo do Exemplo 6.3.2

As quantidades de recurso mínimas (\underline{q}_i) e máximas (\bar{q}_i) admissíveis para a execução de cada atividade do projeto e os tempos de processamento máximos (\bar{p}_i) e mínimos (\underline{p}_i) também correspondem ao da tabela 6.6 do exemplo anterior, reproduzida na tabela 6.10, conforme mostrado abaixo.

Atividade	R	\underline{q}_i	\bar{q}_i	\bar{p}_i	\underline{p}_i
1	10	1	7	20,0	2,86
2	10	2	8	12,5	3,13
3	10	3	9	8,3	2,78
4	10	4	10	7,5	3,00
5	10	5	10	3,0	1,50

Tabela 6.10: Dados técnicos do Exemplo 6.3.2

Inicialmente, uma solução foi obtida considerando os recursos contínuos e aproximando as curvas dos tempos de processamento por múltiplos segmentos (um segmento para cada unidade de intervalo), cujos resultados estão apresentados na tabela 6.11.

Uma solução modificada foi obtida ajustando-se a solução anterior para quantidades inteiras de recurso, como mostra a tabela 6.12 (mais precisamente, os valores não inteiros foram arredondados). Trata-se de uma solução viável, conforme é fácil de se verificar.

Atividade	Número de segmentos	t_i	p_i	x_i	$\underline{q}_i + x_i$
0	-	0	0	-	-
1	6	0	3,1	5,4	6,4
2	6	0	7,1	1,6	3,6
3	6	3,1	4	3,4	6,4
4	6	7,1	3	6	10
5	5	10,1	1,5	5	10
6	-	11,6	0	-	-

Tabela 6.11: Solução do Exemplo 6.3.2 pela aproximação por múltiplos segmentos de reta com recurso contínuo

Atividade	Número de segmentos	t_i	p_i	x_i	$\underline{q}_i + x_i$
0	-	0	0	-	-
1	6	0	3,3	5	6
2	6	0	6,3	2	4
3	6	3,3	4,2	3	6
4	6	7,5	3	6	10
5	5	10,5	1,5	5	10
6	-	12	0	-	-

Tabela 6.12: Solução do Exemplo 6.3.2 pela aproximação por múltiplos segmentos de reta com recurso inteiro

Ambas as soluções estão representadas pelos seus diagramas de Gantt, como mostra as figuras 6.8 e 6.9 respectivamente.

Note que a sequência de execução das atividades permanece igual para os dois casos. Porém, o valor do tempo de processamento de cada atividade foi recalculado em função da quantidade discreta de recurso correspondente. Note também que o tempo para realizar as atividades é maior para a solução modificada (11,6 para a solução original e 12 para a solução modificada).

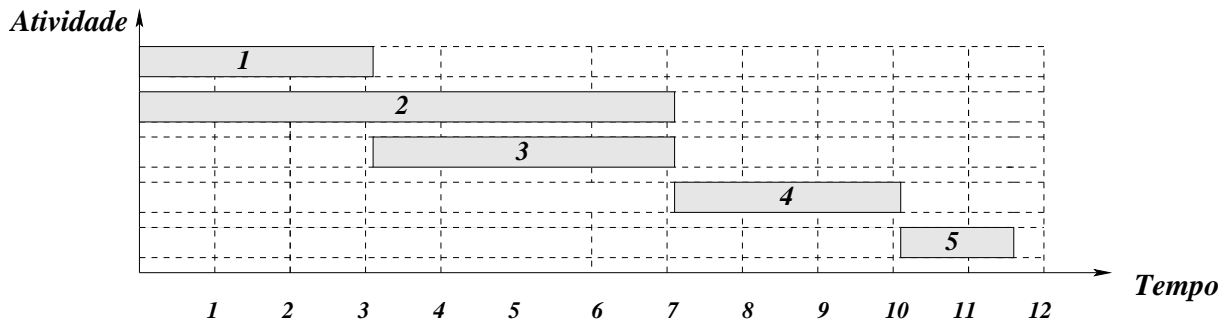


Figura 6.8: Diagrama de Gantt da solução para recurso contínuo

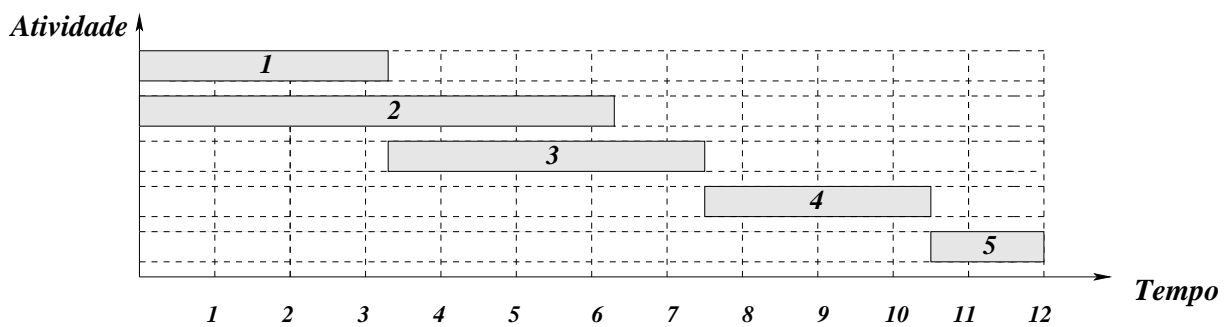


Figura 6.9: Diagrama de Gantt da solução para recurso discreto

6.4 Um problema com múltiplos recursos

Considere um projeto compartilhando dois recursos diferentes. Cada um deles corresponde a um tipo de mão-de-obra especializada (por exemplo, duas categorias de eletricitas). O projeto é constituído pela realização da manutenção de sete equipamentos elétricos. A manutenção de cada um corresponde a uma atividade. Cada atividade necessita de um número total de homens-hora para ser realizada, respeitando um limite mínimo e um limite máximo de trabalhadores (eletricitas). A tabela 6.13 apresenta os dados relativos ao problema. Pretende-se obter a sequência das atividades com a respectiva distribuição de trabalhadores que minimiza o tempo total de manutenção.

Atividade	Recurso 1		Recurso 2	
	Homens-Hora	Limites de trabalhadores	Homens-Hora	Limites de trabalhadores
1	10	$1 \leq \text{trabalhadores} \leq 4$	12	$2 \leq \text{trabalhadores} \leq 6$
2	12	$2 \leq \text{trabalhadores} \leq 5$	14	$4 \leq \text{trabalhadores} \leq 7$
3	14	$3 \leq \text{trabalhadores} \leq 6$	12	$6 \leq \text{trabalhadores} \leq 8$
4	20	$4 \leq \text{trabalhadores} \leq 8$	25	$3 \leq \text{trabalhadores} \leq 9$
5	25	$6 \leq \text{trabalhadores} \leq 9$	20	$5 \leq \text{trabalhadores} \leq 8$
6	30	$7 \leq \text{trabalhadores} \leq 10$	15	$5 \leq \text{trabalhadores} \leq 7$
7	15	$4 \leq \text{trabalhadores} \leq 7$	15	$7 \leq \text{trabalhadores} \leq 9$

Tabela 6.13: Dados gerenciais do Exemplo 6.4

O grafo de representação deste problema é ilustrado na figura 6.10, não havendo nenhuma relação de precedência estabelecida a priori. A atividade 0 e a atividade 8 são atividades fictícias.

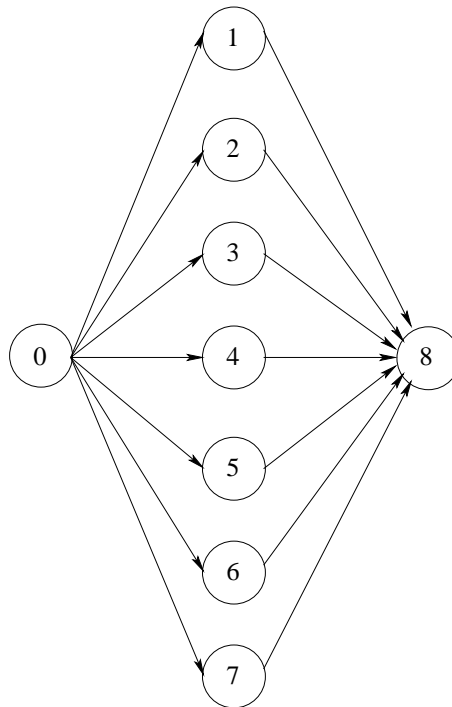


Figura 6.10: Grafo do Exemplo 6.4

A tabela 6.14 apresenta os valores das quantidades mínimas (\underline{q}_{ik}) e máximas (\bar{q}_{ik}) admissíveis para cada tipo de recurso e os respectivos tempos máximos (\bar{p}_{ik}) e mínimos (\underline{p}_{ik}) de processamento.

Atividade	R_1	\underline{q}_{i1}	\bar{q}_{i1}	\bar{p}_{i1}	\underline{p}_{i1}	R_2	\underline{q}_{i2}	\bar{q}_{i2}	\bar{p}_{i2}	\underline{p}_{i2}
1	9	1	4	10	2,50	10	2	6	6	2
2	9	2	5	6	2,40	10	4	7	3,50	2
3	9	3	6	4,67	2,33	10	6	8	2	1,50
4	9	4	8	5	2,50	10	3	9	8,33	2,78
5	9	6	9	4,17	2,78	10	5	8	4	2,50
6	9	7	9	4,29	3,33	10	5	7	3	2,14
7	9	4	7	3,75	2,14	10	7	9	2,14	1,67

Tabela 6.14: Dados técnicos do Exemplo 6.4

Os valores ótimos obtidos para as quantidades de recurso utilizadas em cada atividade, aproximando as curvas dos tempos de processamento por múltiplos segmentos, estão apresentados na tabela 6.15, para o recurso R_1 , e na tabela 6.16, para o recurso R_2 . A tabela 6.17, por sua vez, apresenta os resultados ótimos para as datas de início de execução e os tempos de processamento das atividades.

Atividade	Número de segmentos	x_{i1}	$\underline{q}_{i1} + x_{i1}$
0	-	-	-
1	3	1	2
2	3	2	4
3	3	3	6
4	4	1	5
5	3	3	9
6	2	2	9
7	3	3	7
8	-	-	-

Tabela 6.15: Solução do Exemplo 6.4 pela aproximação por múltiplos segmentos de reta para o recurso R_1

Atividade	Número de segmentos	x_{i2}	$\underline{q}_{i2} + x_{i2}$
0	-	-	-
1	4	1	3
2	3	0	4
3	2	0	6
4	6	3	6
5	3	3	8
6	2	0	5
7	2	0	7
8	-	-	-

Tabela 6.16: Solução do Exemplo 6.4 pela aproximação por múltiplos segmentos de reta para o recurso R_2

Atividade	t_i	p_i
0	0	0
1	6,11	5
2	11,11	3,50
3	6,11	2,33
4	10,58	4,17
5	3,33	2,78
6	0	3,33
7	8,44	2,14
8	14,75	0

Tabela 6.17: Solução do Exemplo 6.4 pela aproximação por múltiplos segmentos

A solução ótima também está representada pelo seu diagrama de Gantt na figura 6.11.

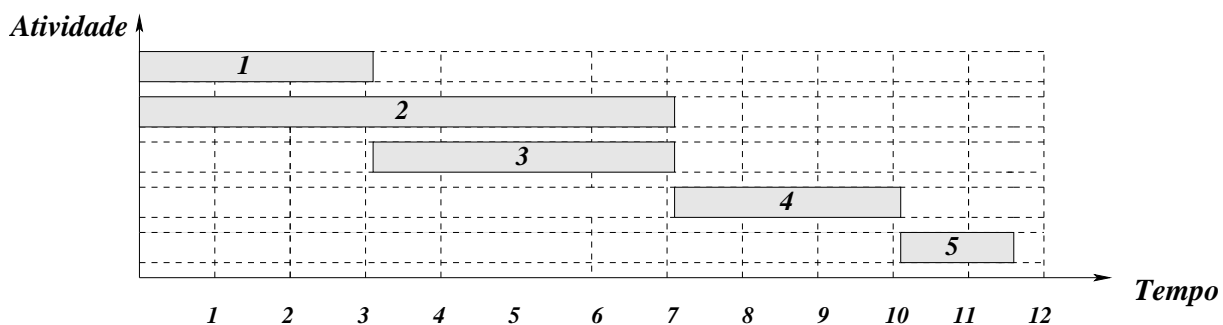


Figura 6.11: Diagrama de Gantt do Exemplo 6.4

6.5 Um problema com tempo de espera

Primeiramente, considere um projeto constituído de dois tipos de recursos a serem compartilhados por um conjunto de seis atividades, como mostra a tabela 6.18. Para efeito de contextualização, considere se tratar da pintura de seis áreas de um prédio comercial. A pintura de cada área corresponde a uma atividade. Os pintores estão divididos em duas categorias e representam os recursos. Considere também que cada área do prédio será pintado apenas por uma camada de tinta, tal que não existe nenhuma precedência pré-estabelecida. O problema é o de sequenciar as atividades e distribuir os pintores de modo a concluir a pintura em tempo mínimo.

Atividade	Recurso 1		Recurso 2	
	Homens-Hora	Limites de trabalhadores	Homens-Hora	Limites de trabalhadores
1	10	$1 \leq \text{trabalhadores} \leq 4$	12	$2 \leq \text{trabalhadores} \leq 6$
2	12	$2 \leq \text{trabalhadores} \leq 5$	14	$4 \leq \text{trabalhadores} \leq 7$
3	14	$3 \leq \text{trabalhadores} \leq 6$	12	$6 \leq \text{trabalhadores} \leq 8$
4	20	$4 \leq \text{trabalhadores} \leq 8$	25	$3 \leq \text{trabalhadores} \leq 9$
5	25	$6 \leq \text{trabalhadores} \leq 9$	20	$5 \leq \text{trabalhadores} \leq 8$
6	30	$7 \leq \text{trabalhadores} \leq 10$	15	$5 \leq \text{trabalhadores} \leq 7$

Tabela 6.18: Dados gerenciais do Exemplo 6.5

O grafo de representação deste problema é ilustrado na figura 6.12. A atividade 0 e a atividade 7 representam, respectivamente, o início e o término do projeto.

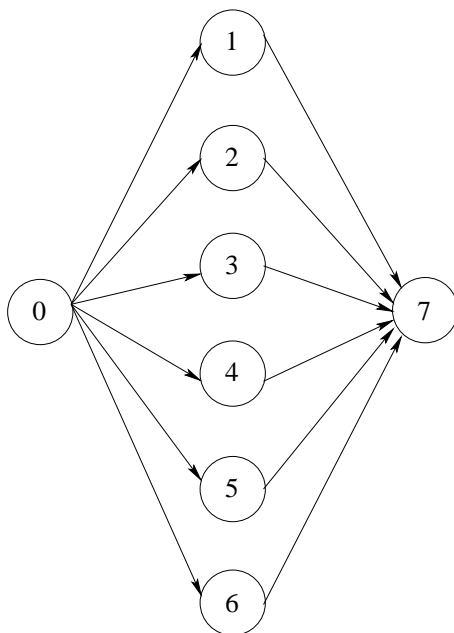


Figura 6.12: Grafo do Exemplo 6.5

A tabela 6.19 apresenta as quantidades dos recursos mínimas (\underline{q}_{ik}) e máximas (\bar{q}_{ik}) admissíveis para a execução de cada atividade do projeto, e dos respectivos tempos de processamento máximos (\bar{p}_{ik}) e mínimos (\underline{p}_{ik}).

Atividade	R_1	\underline{q}_{i1}	\bar{q}_{i1}	\bar{p}_{i1}	\underline{p}_{i1}	R_2	\underline{q}_{i2}	\bar{q}_{i2}	\bar{p}_{i2}	\underline{p}_{i2}
1	9	1	4	10	2,50	10	2	6	6	2
2	9	2	5	6	2,40	10	4	7	3,50	2
3	9	3	6	4,67	2,33	10	6	8	2	1,50
4	9	4	8	5	2,50	10	3	9	8,33	2,78
5	9	6	9	4,17	2,78	10	5	8	4	2,50
6	9	7	9	4,29	3,33	10	5	7	3	2,14

Tabela 6.19: Dados técnicos do Exemplo 6.5

Os valores correspondentes à solução ótima obtida aproximando-se as curvas dos tempos

de processamento por múltiplos segmentos (um segmento para cada unidade de intervalo) estão apresentados nas tabelas 6.20, 6.21 e 6.22.

Atividade	Número de segmentos	x_{i1}	$\underline{q}_{i1} + x_{i1}$
0	-	-	-
1	3	2	3
2	3	2	4
3	3	3	6
4	4	1	5
5	3	3	9
6	2	2	9
7	-	-	-

Tabela 6.20: Alocação ótima do recurso R_1 para o Exemplo 6.5 sem tempo de espera

Atividade	Número de segmentos	x_{i2}	$\underline{q}_{i2} + x_{i2}$
0	-	-	-
1	4	2	4
2	3	0	4
3	2	0	6
4	6	3	6
5	3	3	8
6	2	0	5
7	-	-	-

Tabela 6.21: Alocação ótima do recurso R_2 para o Exemplo 6.5 sem tempo de espera

Atividade	t_i	p_i
0	0	0
1	6,1	3,3
2	9,4	3,5
3	6,1	2,3
4	8,4	4,2
5	3,3	2,8
6	0	3,3
7	12,9	0

Tabela 6.22: Solução ótima do Exemplo 6.5 sem tempo de espera

A solução também está representada pelo seu diagrama de Gantt na figura 6.13.

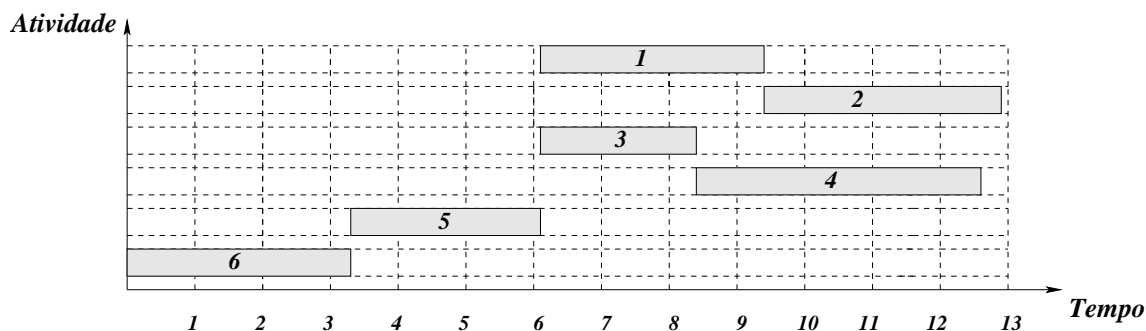


Figura 6.13: Diagrama de Gantt do Exemplo 6.5 sem tempo de espera

Considere agora o problema anterior com a seguinte modificação: suponha que, no lugar de seis áreas, existam somente quatro áreas. Porém, duas das quatro áreas necessitam da aplicação de um selador para posterior aplicação da pintura propriamente dita. Assim, tem-se, como antes, um total de seis atividades. Entretanto, duas correspondem à aplicação do selador e as demais correspondem à aplicação da tinta. A aplicação do selador tem que anteceder a da tinta. Além disso, a tinta só poderá ser aplicada após a secagem completa do selador. O tempo de secagem do selador representa um tempo de espera que, para este exemplo, foi adotado como uma unidade.

O grafo de representação deste problema é mostrado na figura 6.14. Os pares de atividades (1,2) e (6,5) devem ser realizados nas mesmas áreas. As atividades 1 e 6 representam a

aplicação do selador, enquanto as atividades 2 e 5 representam a aplicação da tinta. As atividades 3 e 4 representam a aplicação da tinta em outras áreas sem a necessidade de selador. As atividades 0 e 7 são atividades fictícias.

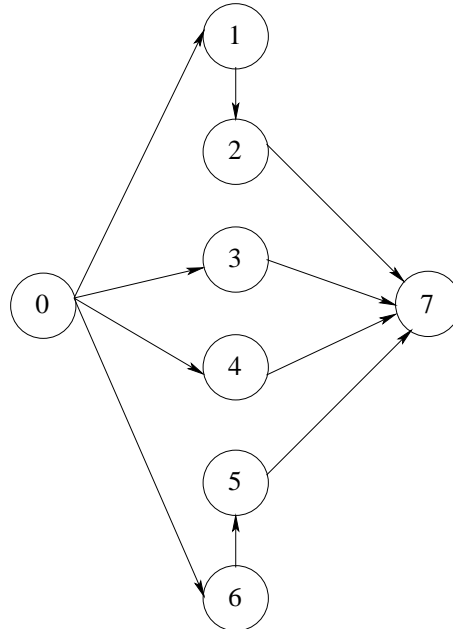


Figura 6.14: Grafo do Exemplo 6.5 com tempo de espera

A solução ótima obtida aproximando-se as curvas dos tempos de processamento por múltiplos segmentos (um segmento para cada unidade de intervalo) está apresentada nas tabelas 6.23, 6.24 e 6.25.

Atividade	Número de segmentos	x_{i1}	$\underline{q}_{i1} + x_{i1}$
0	-	-	-
1	3	3	4
2	3	1	3
3	3	2	5
4	4	4	8
5	3	0	6
6	2	2	9
7	-	-	-

Tabela 6.23: Alocação ótima do recurso R_1 para o Exemplo 6.5 com tempo de espera

Atividade	Número de segmentos	x_{i2}	$\underline{q}_{i2} + x_{i2}$
0	-	-	-
1	4	2	4
2	3	0	4
3	2	0	6
4	6	6	9
5	3	1	6
6	2	0	5
7	-	-	-

Tabela 6.24: Alocação ótima do recurso R_2 para o Exemplo 6.5 com tempo de espera

Atividade	t_i	p_i
0	0	0
1	3,3	3
2	8,9	4
3	3,3	2,7
4	6,3	2,6
5	8,9	4
6	0	3,3
7	12,9	0

Tabela 6.25: Solução ótima do Exemplo 6.5 com tempo de espera

A solução ótima também está representada pelo seu diagrama de Gantt na figura 6.15.

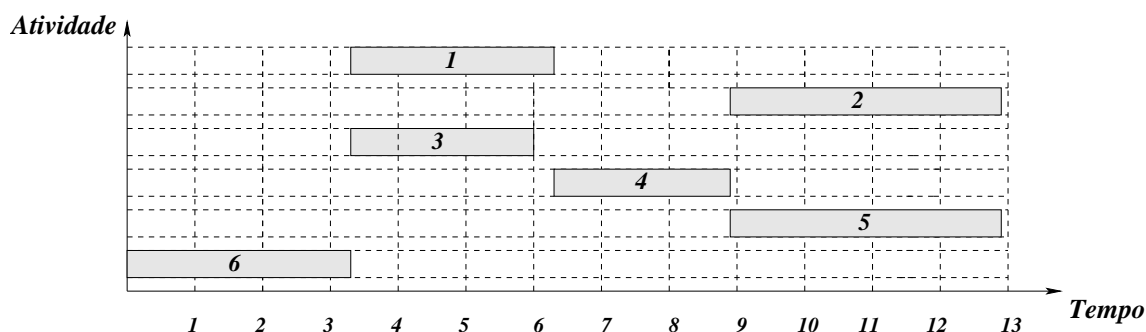


Figura 6.15: Diagrama de Gantt do Exemplo 6.5 com tempo de espera

Note que, na primeira situação, com seis atividades sem tempo de espera, as atividades 1 e 2 e as atividades 6 e 5 foram sequenciadas a posteriori. Já na segunda situação, tendo duas precedências estabelecidas a priori, por causa do tempo de espera, a sequência de execução das atividades foi modificada. Apesar disso, o tempo total de duração do projeto permaneceu igual nas duas situações (12,9). Note ainda que a solução ótima para a segunda situação é também ótima para a primeira. A recíproca, porém, não é verdadeira.

6.6 Comentários

1. Dentro das especificidades de cada um, os exemplos resolvidos apresentaram resultados coerentes e sem qualquer inconsistência que sugerisse alterações na modelagem realizada.
2. Foram realizados testes com o Exemplo 6.3.1 para especular a influência do valor do parâmetro M_3 na obtenção de soluções computacionais. Para isso, introduziu-se a variável artificial v_a na formulação do problema, apesar de não se tratar de uma aplicação do Método de Benders. A tabela 6.26 a seguir apresenta os valores obtidos para a função objetivo e os tempos computacionais considerando diversos valores de M_3 . Todos os valores do parâmetro M_3 adotados foram suficientemente grandes para obter-se soluções ótimas não alteradas (com valores ótimos de v_a nulos).

Problema	Solução	Valor de M_3	Tempo computacional (seg)
Exemplo 6.3.1	11,5	-	3451,25
Exemplo 6.3.1	11,5	99	2999,52
Exemplo 6.3.1	11,5	999	3033,57
Exemplo 6.3.1	11,5	9999	3640,48
Exemplo 6.3.1	11,5	99999	2440,46
Exemplo 6.3.1	11,5	999999	3862,77

Tabela 6.26: Relação do tempo computacional com o valor de M_3

Nota-se que, para todos os valores de M_3 adotados, obteve-se o mesmo valor da função objetivo. Nota-se também que os tempos computacionais alteraram-se muito pouco para todos os valores de M_3 . Em outras palavras, o valor de M_3 , respeitado um limite mínimo, pouco interferiu no tempo computacional.

Capítulo 7

Conclusão

7.1 Comentários finais

Antes de finalizar o trabalho, os seguintes comentários merecem ser considerados:

1. Aproximações por múltiplos segmentos de reta foram utilizadas para representar o tempo de processamento de cada atividade como função linear por partes da quantidade de recurso alocada, com os valores dos tempo de processamento ligeiramente superiores aos tempos exatos. Com auxílio de desigualdades lineares, conforme a seção 4.3.3, estas aproximações tornaram possível reformular o problema original não linear, como um Problema de Programação Linear Inteira Mista. Pela forma de aproximação adotada, as soluções aproximadas obtidas serão sempre soluções viáveis. Isso dá às soluções aproximadas o caráter de soluções subótimas (soluções que atendem a todas as restrições com desempenho um pouco inferior ao desempenho ótimo). Através do número de segmentos adotado, pode-se controlar a diferença máxima entre a curva segmentada (linear por partes) e a curva exata, permitindo aproximar-se da solução ótima exata tanto quanto se queira.
2. Conforme comentado no final do Capítulo 4, a aproximação por segmentos de reta foi proposta para relações entre o tempo de processamento e a quantidade de recurso alocada do tipo hiperbólica ($p_{ik} = \alpha_{ik}/q_{ik}$). No entanto, o procedimento pode ser facilmente estendido a outras formas não lineares, ressalvado que as funções

$p_{ik} = \phi_{ik}(q_{ik})$ sejam contínuas, monotônicas e convexas no domínio $q_{ik} \in [\underline{q}_{ik}, \bar{q}_{ik}]$.

3. A formulação do Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR), expressada no Capítulo 4, incorpora restrições de precedência práticas que não estão presentes na literatura pesquisada. Em primeiro lugar, considera-se a existência de atividades que não têm nenhuma relação de precedência estabelecida a priori, exceto que as mesmas não podem ser realizadas simultaneamente. Isso acontece, por exemplo, em atividades não correlacionadas, mas que devem ser realizadas em um mesmo local, porém com a impossibilidade de duas equipes trabalhando simultaneamente no local. Em segundo lugar, admitiu-se a fixação de intervalos mínimos entre o término de uma atividade e a data de início de uma atividade subsequente (tempo mínimo de espera).
4. A incorporação de tempos de espera na formulação do PSPRR obriga a consideração de dois tipos distintos de precedência. O primeiro é a precedência a priori. Neste caso, trata-se de uma precedência imposta por critérios físicos operacionais, exigindo a consideração do tempo mínimo de espera. O segundo tipo é a precedência a posteriori. Neste caso, a precedência ocorre somente como consequência do processo de otimização, não tendo razões físicas especiais, motivo pelo qual não deve levar em conta o tempo mínimo de espera.
5. A aproximação linear por partes foi idealizada para problemas com recursos contínuos. Entretanto, ela pode ser também aplicável a problemas com recursos discretos, visando a obtenção de boas soluções relaxadas de modo eficiente. Estas soluções relaxadas podem ser manipuladas a posteriori com baixo esforço computacional. Conjectura-se que esta estratégia pode ser interessante para a solução de problemas com recursos discretos, especialmente na aplicação do Método de Benders, uma vez que reduz significativamente o número de variáveis discretas do problema, aliviando o problema mestre.
6. Conforme descrito no Capítulo 4, estabeleceu-se uma estratégia para atribuição automática de valores para os parâmetros M_1 e M_2 . Essas atribuições são baseadas em limitações físicas do problema real, evitando-se atribuições exageradas ou subestimadas destes parâmetros. Considera-se essa ser também uma pequena contribuição do presente estudo. De fato, na literatura (ver Artigues *et al.* [2] e Deblaere *et al.* [21]) não foram encontradas publicações nas quais esses parâmetros aparecessem com critérios explícitos de atribuição de valor.

7.2 Conclusões

Com relação ao presente estudo, pode-se extrair as seguintes conclusões:

1. Apresentou-se uma nova formulação do Problema de Sequenciamento em Projeto com Restrição de Recursos (PSPRR) que incorpora relações não lineares entre o tempo de processamento e a quantidade de recurso alocada e dois tipos adicionais de restrições de precedência.
2. A formulação do PSPRR proposta considera tanto a quantidade de recurso alocada quanto o tempo de processamento de cada atividade como variáveis contínuas a serem otimizadas.
3. Através de aproximações lineares com número arbitrário de segmentos de reta, aplicadas às relações que representam o tempo de execução das atividades como função da quantidade de recurso alocada, foi possível resgatar a natureza linear do PSPRR.
4. Com artifícios matemáticos simples foi possível incorporar na formulação do PSPRR a possibilidade de restrições para atividades que não têm nenhuma exigência de precedência a priori, exceto a impossibilidade de serem executadas simultaneamente.
5. Artifícios matemáticos simples também tornaram possível incorporar ao PSPRR o tempo mínimo de espera de atividades, sem que a natureza linear do problema seja quebrada.
6. A consideração do tempo de espera divide as relações de precedência em dois tipos: precedência a priori e precedência a posteriori. Somente o primeiro tipo deve levar em conta o tempo mínimo de espera.
7. Estabeleceu-se uma forma fechada de atribuição de valores válidos (suficientes) e não exagerados para os parâmetros M_1 e M_2 .
8. Diferentemente, não se apresentou uma forma fechada para atribuição de um valor para o parâmetro M_3 . Entretanto, testes realizados sugerem que o valor de M_3 , desde que superior ao valor mínimo necessário, ofereça pouca influência no tempo computacional para se obter a solução ótima.
9. A técnica de aproximação desenvolvida para recursos contínuos pode ser adotada como uma estratégia numérica que pode facilitar a obtenção de soluções para problemas com recursos discretos.

10. Problemas com recursos discretos, quando tratados numericamente como recursos contínuos, devem receber um tratamento a posteriori para adaptar a solução obtida. O tratamento a posteriori é simples de ser realizado.
11. É importante a realização de estudos para melhorar o desempenho da aplicação do Método de Decomposição de Benders na solução do PSPRR. De fato, não obstante a elegância matemática do Método de Benders, sua aplicação apresentou desempenho inferior quando comparada com outros métodos de resolução, especialmente ao CPLEX.

7.3 Perspectivas de continuidade do estudo

Em continuidade ao presente estudo, pode-se sugerir:

1. Aprimoramento do modelo apresentado, através da inclusão de novas restrições ao problema, tornando-o cada vez mais próximo da realidade gerencial. Um exemplo seria considerar no mesmo projeto a utilização de recursos renováveis e não renováveis. Outro exemplo seria considerar relações entre atividades que devem, necessariamente, ser executadas uma imediatamente após a outra. Ou seja, entre duas atividades i e j , a data de início da atividade j seria igual a data de início da atividade i mais o seu tempo de processamento.
2. Soluções de problemas de grandes dimensões através da implementação de técnicas matemáticas e computacionais aprimoradas, em particular, versões melhoradas do Método de Decomposição de Benders.
3. Estudo de técnicas que viabilizem a introdução de cortes que agilizem o procedimento de otimização, importantes para uma eficiente implementação computacional.

Referências Bibliográficas

- [1] R. N. Anthony and V. Govindarajan. *Management control systems*. Irwin McGraw-Hill, 1998.
- [2] C. Artigues, P. Michelon, and S. Reusser. Insertion techniques for static and dynamic resource constrained project scheduling. *European Journal of Operational Research*, 149:249–267, 2003.
- [3] E. Balas. Project scheduling with resource constraints. *Operational Research*, 15:915–957, 1967.
- [4] P. P. Belfiore, O. L. V. Costa, and L. P. L. Fávero. Decomposição de Benders para programação mista e aplicações ao problema de estoque e roterização. *Revista Gestão Industrial*, 01:123–136, 2005.
- [5] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 04:238–252, 1962.
- [6] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Scheduling computer and manufacturing processes*. Springer, 1996.
- [7] P. O. Boaventura. *Grafos-teoria, modelos, algoritmos*. Ed. Edgard Blücher Ltda, 2003.
- [8] F. F. Boctor. Some efficient multi-heuristic procedures for resource-constrained project scheduling. *European Journal of Operational Research*, 49:3–13, 1990.
- [9] P. Brucker and S. Knust. A linear programming and constraint propagation-based lower bound for the resource constrained project scheduling problem. *European Journal of Operational Research*, 127:355–362, 2000.

- [10] P. Brucker, S. Knust, A. Schoo, and O. Thiele. A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 107:272–288, 1998.
- [11] J. Carlier and E. Néron. On linear lower bounds for the resource constrained project scheduling problem. *European Journal of Operational Research*, 149:314–324, 2003.
- [12] J. Chen and R. G. Askin. Project selection, scheduling and resource allocation with time dependent returns. *European Journal of Operational Research*, 193:23–34, 2009.
- [13] Y. Chen, D. Rinks, and K. Tang. Critical path in an activity network with time constraints. *European Journal of Operational Research*, 100:122–133, 1997.
- [14] N. Christofides, R. Alvarez-Valdes, and J. M. Tamarit. Project scheduling with resource constraints: a branch and bound approach. *European Journal of Operational Research*, 29:262–273, 1987.
- [15] A. S. da Cunha. Um estudo sobre a otimização dos parâmetros de controle de um algoritmo genético. Master’s thesis, UFMG, 2002.
- [16] S. Dauzère-Pèrés, W. Roux, and J. B. Lasserre. Multi-resource shop scheduling with resource flexibility. *European Journal of Operational Research*, 107:289–305, 1998.
- [17] E. W. Davis and J. H. Patterson. A comparison of heuristic and optimum solutions in resource constrained project scheduling. *Management Science*, 21:944–955, 1975.
- [18] W. E. Davis. Project scheduling under resources constraints-historical review and categorization of procedures. *AIIE Transactions*, 5:147–163, 1973.
- [19] R. S. de Camargo, G. Miranda Jr., R. P. M. Ferreira, and H. P. Luna. Multiple allocation hub-and-spoke network design under hub congestion. *Computers and Operations Research*, 36:3097–3106, 2009.
- [20] C. R. V. de Carvalho. Une proposition d’intégration de la planification et l’ordonnancement de production: application de la méthode de Benders. Master’s thesis, Blaise Pascal, 1998.
- [21] F. Deblaere, E. Demeulemeester, W. Herroelen, and S. V. Vonder. Robust resource allocation decisions in resource constrained projects. *Decision Sciences*, 38:5–37, 2007.

- [22] E. L. Demeulemeester and W. S. Herroelen. An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem. *European Journal of Operational Research*, 90:334–348, 1996.
- [23] L. G. Fendly. Toward the development of a complete multiproject scheduling system. *Journal of Ind Engineering*, 19:505–515, 1968.
- [24] M. Fisher. Optimal solution of scheduling problems using Lagrange multipliers. part i. *Operational Research*, 21:1114–1127, 1973.
- [25] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [26] D. D. Gemmill and M. L. Edwards. Improving resource constrained project scheduling with look ahead techniques. *Project Management Journal*, 30:44–55, 1999.
- [27] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.
- [28] M. C. Goldbarg and H. P. L. Luna. *Otimização Combinatória e Programação Linear - Modelos e Algoritmos*. Ed. Campus, 2005.
- [29] M. Gondran and M. Minoux. *Graphs et algorithmes*. Eyrolles, 1995.
- [30] S. Gümüsoglu and H. Tütek. An analysis method in project management using primal-dual relationships. *International Journal of Project Management*, 16:321–327, 1998.
- [31] A. C. Hax and D. Candea. *Production and inventory management*. Prentice-Hall, 1984.
- [32] R. Heilmann. A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research*, 144:348–365, 2003.
- [33] F. S. Hiller and G. J. Lieberman. *Introdução à pesquisa operacional*. Editora Campus LTDA. Editora da Universidade de São Paulo, 1988.
- [34] O. Icmeli and S. S. Erenguc. A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows. *Computers Operations Research*, 21:841–853, 1994.

- [35] P. A. Jensen. *Operations research models and methods*. Wiley, 2003.
- [36] J. E. Kelley and M. R. Walker. *Scheduling activities to satisfy resource constraints*. Prentice-Hall, 1963.
- [37] H. Kerzner. *Project management for executives*. Van Nostrand Reinhold Company, 1982.
- [38] R. Klein and A. Scholl. Computing lower bounds by destructive improvement: An application to resource-constrained project scheduling. *European Journal of Operational Research*, 112:322–346, 1999.
- [39] R. Kolisch. Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, 14:179–192, 1996.
- [40] R. Kolisch and A. Sprecher. Psplib - a project scheduling problem library. *European Journal of Operational Research*, 96:205–216, 1996.
- [41] R. Kolisch, A. Sprecher, and A. Drexl. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41:1693–1703, 1995.
- [42] P. D. Konstantinidis. A model to optimized project resource allocation by construction of a balanced histogram. *European Journal of Operational Research*, 104:559–571, 1998.
- [43] L. S. Lasdon. *Optimization Theory for Large Systems*. MacMillan, 1970.
- [44] D. G. Luenberger. *Linear and nonlinear programming*. Addison-Wesley Publishing Company, 1989.
- [45] A. Mingozzi, V. Maniezzo, S. Ricciardelli, and L. Bianco. An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Management Science*, 44:714–729, 1998.
- [46] M. Minoux. *Mathematical programming theory and algorithms*. Wiley-Interscience, 1986.
- [47] K. Moumene and J. A. Ferland. Activity list representation for a generalization of the resource constrained project scheduling problem. *European Journal of Operational Research*, 199:46–54, 2009.

- [48] K. Neumann, C. Schwindt, and J. Zimmermann. *Project scheduling with time windows and scarce resource*. Springer Verlag, 2002.
- [49] O. Oguz and H. Bala. A comparative study of computational procedures for the resource constrained project scheduling problem. *European Journal of Operational Research*, 72:406–416, 1994.
- [50] J. H. Patterson, F. B. Talbot, R. Slowinski, and J. Weglarz. Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems. *European Journal of Operational Research*, 49:68–79, 1990.
- [51] V. V. Peteghem and M. Vanhoucke. A genetic algorithm for the preemptive and non-preemptive multi-mode resource constrained project scheduling problem. *European Journal of Operational Research*, 201:409–418, 2010.
- [52] R. C. Pfaffenberger and D. A. Walker. *Mathematical programming for economics and business*. Iowa state University Press, 1976.
- [53] M. Pich, C. Loch, and A. Meyer. On uncertainty, ambiguity, and complexity in project management. *Management Science*, 48:1008–1023, 2002.
- [54] M. L. Pinedo. *Planning and scheduling in manufacturing and services*. Springer, 2006.
- [55] E. Poder, N. Beldiceanu, and E. Sanlaville. 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, 2004.
- [56] A. B. Pritsker, L. J. Watters, and P. M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16:93–108, 1969.
- [57] B. Roy. Contribution de la théorie des graphes a l'étude des problème d'ordonnancement. In *Les Problème d'Ordonnancement, Applications et Méthodes*, pages 109–125. Dunod, 1964.
- [58] L. Schrage. Solving resource-constrained network problems by implicit enumeration-nonpreemptive case. *Operational Research*, 18:225–235, 1971.
- [59] R. Slowinski. Multiobjective network scheduling with efficient use of renewable and non-renewable resources. *European Journal of Operational Research*, 7:265–273, 1981.

- [60] G. A. Steiner. *Top management planning*. Macmillan, 1969.
- [61] V. Valls, S. Quintanilla, and F. Ballestín. Resource-constrained project scheduling: a critical activity reordering heuristic. *European Journal of Operational Research*, 149:282–301, 2003.
- [62] C. S. Vieira. Modelos para o sequenciamento de atividades em projetos com restrições de recursos. Master's thesis, UFMG, 2004.
- [63] J. Weglarz. Project scheduling with continuously divisible, doubly constrained resources. *Management Science*, 27:1040–1053, 1981.
- [64] J. D. Wiest. The scheduling of large projects with limited resources. Master's thesis, Carnegie Institute of Technology, 1963.
- [65] J. D. Wiest. A heuristic model for scheduling large projects with limited resources. *Management Science*, 13:B359–B377, 1967.
- [66] T. Williams. A classified bibliography of recent research relating to project risk management. *European Journal of Operational Research*, 85:18–38, 1995.
- [67] R. Willis. Critical path analysis and resource constrained project scheduling - theory and practice. *European Journal of Operational Research*, 21:149–155, 1985.
- [68] M. Wisniewski and J. H. Klein. *Linear programming Critical path analysis*. Palgrave, 2001.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)