



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2017/04.07.19.13-TDI

**ARTEFATO PARA SIMULAÇÃO DO PROCESSO DE
TIME COST TRADE-OFF USANDO FUNÇÕES
CONTÍNUAS DO TIPO LINEAR E EXPONENCIAL
CONVEXA**

Alexandre Balistrieri

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelo Dr. Milton de Freitas Chagas Junior, aprovada em 20 de fevereiro de 2017.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3NLMTH8>>

INPE
São José dos Campos
2017

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@inpe.br

COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):

Presidente:

Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação (CPG)

Membros:

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Dr. André de Castro Milone - Coordenação de Ciências Espaciais e Atmosféricas (CEA)

Dra. Carina de Barros Melo - Coordenação de Laboratórios Associados (CTE)

Dr. Evandro Marconi Rocco - Coordenação de Engenharia e Tecnologia Espacial (ETE)

Dr. Hermann Johann Heinrich Kux - Coordenação de Observação da Terra (OBT)

Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Silvia Castro Marcelino - Serviço de Informação e Documentação (SID)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2017/04.07.19.13-TDI

**ARTEFATO PARA SIMULAÇÃO DO PROCESSO DE
TIME COST TRADE-OFF USANDO FUNÇÕES
CONTÍNUAS DO TIPO LINEAR E EXPONENCIAL
CONVEXA**

Alexandre Balistrieri

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelo Dr. Milton de Freitas Chagas Junior, aprovada em 20 de fevereiro de 2017.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3NLMTH8>>

INPE
São José dos Campos
2017

Dados Internacionais de Catalogação na Publicação (CIP)

Balistreri, Alexandre.

B198a Artefato para simulação do processo de Time Cost Trade-off usando funções contínuas do tipo linear e exponencial convexa / Alexandre Balistreri. – São José dos Campos : INPE, 2017.

xx + 156 p. ; (sid.inpe.br/mte-m21b/2017/04.07.19.13-TDI)

Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2017.

Orientador : Dr. Milton de Freitas Chagas Junior.

1. Crashing. 2. Time-Cost Trade-off Problem. 3. Monte Carlo.
4. Critical path method. I.Título.

CDU 629.7.018



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).


This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

Aluno (a): **Alexandre Balistrieri**

Título: " ARTEFATO PARA SIMULAÇÃO DO PROCESSO DE TIME COST TRADE-OFF
USANDO FUNÇÕES CONTÍNUAS DO TIPO LINEAR E EXPONENCIAL CONVEXA".

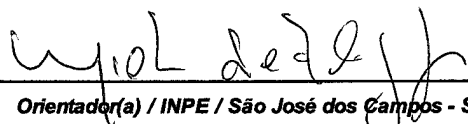
Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Mestre** em
**Engenharia e Tecnologia Espaciais/Eng.
Gerenc. de Sistemas Espaciais**

Dra. Ana Maria Ambrosio



Presidente / INPE / São José dos Campos - SP

Dr. Milton de Freitas Chagas Junior



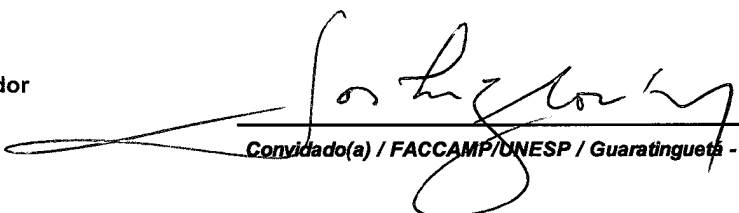
Orientador(a) / INPE / São José dos Campos - SP

Dr. Maurício Gonçalves Vieira Ferreira



Membro da Banca / INPE / SJCampos - SP

Dr. José Luiz Contador



Convidado(a) / FACCAMP/UNESP / Guaratinguetá - SP

Este trabalho foi aprovado por:

maioria simples

unanimidade

São José dos Campos, 20 de Fevereiro de 2017

AGRADECIMENTOS

Por proporcionar os recursos, meio acadêmico e apoio profissional, agradeço ao INPE, docentes e colegas que dele fazem um ambiente de trabalho e estudos riquíssimos.

Pelo sacerdócio, incentivo e pelo cabedal de conhecimentos, sem os quais este trabalho não seria possível, agradeço ao professor Dr. Milton de Freitas Chagas Junior, criador e motivador da ideia desta dissertação.

Pelo conhecimento, incentivo, carinho e incontáveis horas de apoio na confecção desta dissertação, plantando novas ideias e direcionando o trabalho, agradeço ao colega e amigo Dr. Giuliani Paulineli Garbi.

Pela motivação, incentivo, e pelo objetivo nascido em mim na busca pelo stricto sensu em mestrado, agradeço aos mentores e semeadores desta minha nova etapa de estudos acadêmicos, meus caríssimos amigos e colegas Koiti Ozaki e Carlos Shinya Shibata.

Pela cumplicidade, amor fraternal e presença constante, pela ajuda, conselhos e ensinamentos valiosos, por manter a família sempre coesa no seu caloroso abraço e por saber ouvir quando precisei, agradeço a minha irmã Rosalinda Balistriieri Dias. A ela meu maior carinho, consideração e respeito.

Pela paciência com as minhas manias, pelo respeito e admiração por mim, pelo orgulho que cada um a seu modo me faz sentir, me levando às lágrimas sempre por alegria, saudade e reconhecimento de suas realizações, agradeço a meus filhos Sandro, Bruno e Felipe. A eles a minha maior torcida por um futuro próspero e feliz.

Responsáveis pela carreira que escolhi seguir, pela formação pessoal, acadêmica e profissional, pelo exemplo, esforço e dedicação, agradeço postumamente a meus pais Alexandre e Claudete, que estejam onde estiverem, saibam da minha eterna gratidão por tudo que realizei até hoje. A eles o maior galardão e o meu maior tributo.

Pelo amor incondicional e compreensão, pelo carinho e dedicação, por não me deixar esmorecer diante de qualquer obstáculo, compartilhando comigo os bons e maus momentos, agradeço a minha amada esposa Marisa Pedote, por estar sempre perto, fisicamente ou em pensamento, sempre me ajudando e zelando por mim. A ela, meu amor eterno e profunda gratidão por tudo que realizei e conquistei nos últimos 30 anos.

RESUMO

Esta dissertação apresenta um artefato de *software* que sistematiza em computador o método *Crashing* de otimização de rede de projetos e quantifica os riscos envolvidos com o apoio da simulação Monte Carlo. O artefato baseia-se no método CPM (*Critical Path Method*) de balanceamento entre as estimativas de tempo e custo, conhecido como *Time-Cost Trade-off Problem* (TCTP) e possui duas versões de aproximação da função custo, sendo uma linear e outra exponencial convexa como contribuição. Ao fazer uma simulação e análise das estimativas dos custos e das durações das atividades nos caminhos possíveis do projeto, o artefato demonstrou em dois estudos de caso abordados, a utilidade prática da ferramenta computacional desenvolvida. O artefato corrobora a dificuldade de uma abordagem manual do problema mostrando a interação das variáveis envolvidas e a variabilidade delas ao se aplicar mudanças em suas estimativas. Adicionalmente, foi feita uma quantificação dos riscos resultantes da incerteza presente na variabilidade das estimativas e o resultado mostrou as probabilidades de sucesso ou fracasso diante dos múltiplos cenários aleatórios gerados com o apoio da técnica de simulação Monte Carlo.

Palavras-chave: *Crashing*. *Time-Cost Trade-off Problem*. Monte Carlo. *Critical Path Method*.

ARTIFACT FOR SIMULATION OF THE TIME COST TRADE-OFF PROCESS USING LINEAR AND EXPONENTIAL CONVEX CONTINUOUS FUNCTIONS

ABSTRACT

This dissertation presents a software artifact that systematizes in computer the Crashing method of network optimization of projects and quantifies the risks with the Monte Carlo simulation support. The artifact is an algorithmic approach that solves the Time-Cost Trade-off Problems (TCTP) in Critical Path Method network flows (CPM) and has two cost approximation versions: Linear and Exponential Convex as contribution. In doing a simulation and analysis of the cost estimates and the durations of the activities in the possible project paths, the artifact showed in two case studies addressed, the practical utility of the developed computational tool. The artifact corroborates the difficulty of a manual approach to the problem showing the interaction of the variables involved and their variability when applying changes in their estimates. Additionally, it was made a risk quantification present in the variability of the estimates and the result showed the probabilities of success or failure against the multiple random scenarios generated with the support of the Monte Carlo simulation technique.

Key words: Crashing. Time-Cost Trade-off Problem. Monte Carlo. Critical Path Method.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1.1 – Publicações anuais sobre TCTP nos últimos 20 anos	4
Figura 1.2 – Total de citações anuais referentes ao tema TCTP nos últimos 20 anos	4
Figura 2.1 – (a) Triângulo Tempo/Custo/Desempenho; (b) Desvios de Estimativas no Triângulo Tempo/Custo/Desempenho.	9
Fonte: Adaptado de Kerzner (2009).	9
Figura 2.2 – Grafo com tipo de rede ANN	14
Figura 2.3 – Grafo com tipo de rede ANA	15
Figura 3.1 – Rede de Projeto Hipotético com 4 atividades.....	22
Figura 4.1 – Fluxograma Geral do Algoritmo.....	34
Figura 4.2 – Fluxograma do passo 2: Identificar os caminhos da rede	37
Figura 4.3 - Fluxograma do cálculo das durações e folgas dos caminhos	39
Figura 4.4 – Fluxograma da programação para frente	42
Figura 4.5 – Fluxograma do Passo 7: Escolher Conjunto de Aceleração	45
Figura 4.6 – Combinação com mapeamento binário.....	47
Figura 4.7 – Fluxograma Gerar Combinações por mapeamento binário.....	48
Figura 4.8 – Fluxograma do módulo para filtrar aceleração	50
Figura 4.9 – Evolução dos custos, indireto e marginal para uma dada atividade de um projeto	54
Figura 4.10 – Gráfico de evolução do custo marginal linear e convexo	55
Figura 4.11 – Gráfico de evolução dos custos durante a compressão.....	56
Figura 4.12 – Fluxograma Filtro Custo Mínimo	58
Figura 5.1 – Gráfico comparativo das compressões linear e exponencial convexa (Caso 1)	78
Figura 5.2 - Gráfico comparativo das compressões linear e exponencial convexa (Caso 2)	79
Figura 6.1 – Distribuição de Frequência Acumulada da Simulação do Estudo de Caso 1	82

Figura 6.2 – Distribuição de Frequência Acumulada da Simulação do Estudo de Caso 2.....	83
--	----

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 1.1 – Interesse do tema na comunidade Acadêmico-Científica.....	3
Tabela 2.1 - Cronograma da etapa de AIT do CBERS-3 do INPE	10
Tabela 2.2 – Exemplos de grau de confiança e desvio padrão.....	19
Tabela 3.1 – Detalhes, e características das abordagens de TCTP)	32
Tabela 5.1: Dados de entrada do Algoritmo – Caso 1.....	61
Tabela 5.2 – Relatório Cenário Inicial (Caso 1, Versão Linear)	62
Tabela 5.3 – Resultados calculados antes da compressão para o caso 1 (versão Linear)	63
Tabela 5.4 – Cenário 1ª iteração (Caso 1, Versão Linear)	64
Tabela 5.5 - Resultados calculados no 1º cenário do caso 1 (versão Linear) ..	65
Tabela 5.6 – Cenário 2ª iteração (Caso 1, Versão Linear)	66
Tabela 5.7 – Cenário 3ª iteração (Caso 1, Versão Linear)	66
Tabela 5.8 – Cenário 4ª iteração (Caso 1 - Versão Linear).....	67
Tabela 5.9: Dados de entrada do Algoritmo – Caso 2.....	68
Tabela 5.10 – Primeiro cenário sem compressão (Caso 2, Versão Linear)	69
Tabela 5.11 – Último cenário após o <i>Crashing</i> (Caso 2, Versão Linear).....	71
Tabela 5.12 – Evolução do <i>Crashing</i> (Caso 2, Versão Linear)	72
Tabela 5.13 – Relatório Cenário Inicial – (Caso 1, Versão Exponencial Convexa)	73
Tabela 5.14 – Custos das compressões possíveis no 1º cenário do caso 1 (versão exponencial convexa).....	74
Tabela 5.15 – Cenário 1ª iteração – (Caso 1, Versão Exp. Convexa).....	75
Tabela 5.16 – Cenário 2ª iteração – (Caso 1, Versão Exp. Convexa).....	76
Tabela 5.17 – Cenário 3ª iteração – (Caso 1, Versão Exp. Convexa).....	76
Tabela 5.18 – Cenário 4ª iteração – (Caso 1, Versão Exp. Convexa).....	77
Tabela 7.1 – Quadro comparativo da sistematização do <i>Crashing</i> por aproximação linear e convexa.....	88

LISTA DE SIGLAS E ABREVIATURAS

ACO	<i>Ant Colony Optimization.</i>
ACS	<i>Ant Colony System.</i>
AIT	<i>Assembly, Integration and Testing.</i>
ANA	Atividade no Arco.
ANN	Atividade no Nó.
AOA	<i>Activity on Arrow.</i>
AON	<i>Activity on Node.</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.
CBERS	<i>China Brazil Earth Resources Satellite.</i>
CPM	<i>Critical Path Method.</i>
GA	<i>Genetic Algorithm.</i>
INPE	Instituto Nacional de Pesquisas Espaciais.
PDM	<i>Precedence Diagram Method.</i>
PERT	<i>Program Evaluation and Review Technique.</i>
PSO	<i>Particle Swarm Optimization.</i>
SID	Serviço de Informação e Documentação.
SPG	Serviço de Pós-Graduação.
TCT	<i>Time-Cost Trade-off.</i>
TCTP	<i>Time-Cost Trade-off Problem.</i>

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO.....	1
1.1. Objetivo Geral.....	2
1.2. Objetivos Específicos.....	3
1.3. Análise do interesse do tema na comunidade Acadêmico-Científica.....	3
1.4. Metodologia Científica e de Implementação.....	5
1.5. Estrutura da Dissertação.....	7
2 REVISÃO BIBLIOGRÁFICA.....	9
2.1. Diagrama de Gantt.....	10
2.2. PERT- Program Evaluation and Review Technique.....	10
2.3. CPM – Método do Caminho Crítico (<i>Critical Path Method</i>).....	11
2.4. Redes de Projetos.....	13
2.4.1. Desenhos de Redes tipo ANN e ANA.....	14
2.5. Gerenciamento dos Riscos.....	15
2.5.1. Processos do Gerenciamento dos Riscos.....	16
2.5.2. Técnicas do Processo de Análise Quantitativa dos Riscos.....	17
2.5.3. Simulação Monte Carlo.....	17
2.5.3.1. Ferramentas de Simulação Monte Carlo.....	19
3 FUNDAMENTAÇÃO TEÓRICA.....	21
3.1. TCTP – Balanceamento entre tempo e custo.....	21
3.1.1. Custos envolvidos em Projetos.....	22
3.2. Métodos de resolução do TCTP.....	23
3.3. Métodos Heurísticos.....	25
3.3.1. Método de Aproximação de Siemens.....	25
3.3.2. Método de Aproximação Linear de Contador.....	26
3.4. Métodos Meta-Heurísticos.....	27
3.4.1. Algoritmos Genéticos (<i>Genetic Algorithm - GA</i>).....	27
3.4.2. Otimização por Enxame de Partículas (<i>Particle Swarm Optimization - PSO</i>).....	29

3.4.3.	Otimização por Colônia de Formigas (<i>Ant Colony Optimization – ACO</i>)	30
3.4.4.	Otimização Híbrida.....	31
4	DESENVOLVIMENTO DO ARTEFATO DE SOFTWARE PARA SIMULAÇÃO DO PROCESSO DE <i>TIME COST TRADE-OFF</i>	33
4.1.	Sistematização do Algoritmo de Contador.....	33
4.1.1.	Passo 1 – Ler dados de entrada e escolher o critério de parada ..	35
4.1.2.	Passo 2 – Identificar os caminhos da rede.....	36
4.1.3.	Passo 3 – Gerar Relatório do Cenário	38
4.1.4.	Passos 4 e 5 - Calcular as durações de cada caminho identificando caminhos críticos e folga de cada caminho.....	38
4.1.5.	Passo 6 – Calcular início e término de cada atividade	40
4.1.5.1.	Programação para Frente e para Trás	40
4.1.6.	Passo 7 – Escolher conjunto candidato a aceleração	43
4.1.6.1.	Gerar um Conjunto para aceleração	46
4.1.6.2.	Filtrar Conjunto Acelerável	49
4.1.6.3.	Filtrar Redundância	51
4.1.6.4.	Filtrar Caminho Crítico	51
4.1.6.5.	Filtrar Custo Mínimo	51
4.1.6.6.	Encontrar o Passo de Compressão.....	55
4.1.6.7.	Encontrar o Custo Marginal na Curva de aproximação.....	56
4.1.7.	Passo 8 - Processar Compressão e Calcular Custos.....	59
4.1.8.	Passo 9 - Verificar critério de parada	59
5	ESTUDOS DE CASO	61
5.1.	Caso 1 – Versão Linear	61
5.2.	Caso 2 – Versão Linear	68
5.2.1.	Relatório do primeiro cenário sem compressão – Caso 2.....	69
5.3.	Caso 1 – Versão Exponencial Convexa.....	72
5.4.	Caso 2 – Versão Exponencial Convexa.....	78
6	QUANTIFICAÇÃO DOS RISCOS.....	81
6.1.	Quantificação do Risco do Estudo de caso 1	82

6.2. Quantificação do Risco do Estudo de Caso 2.....	83
7 CONCLUSÕES.....	85
7.1. Limitações.....	86
7.2. Posicionamento do Artefato em Relação à Bibliografia	87
7.3. Considerações finais.....	88
REFERÊNCIAS BIBLIOGRÁFICAS	91
APÊNDICE A – Artefato de Software Crashing Versão Linear.....	95
APÊNDICE B – Artefato de Software para o Crashing Versão Convexa	105
APÊNDICE C – Relatório Estudo de Caso 1 Versão Linear	115
APÊNDICE D – Relatório Estudo de Caso 2 Versão Linear	119
APÊNDICE E – Relatório Estudo de Caso 1 Versão Convexa.....	133
APÊNDICE F – Relatório Estudo de Caso 2 Versão Convexa.....	137

1 INTRODUÇÃO

Esta dissertação de mestrado é endereçada ao processo de programação da fase de planejamento do ciclo de vida do gerenciamento de projetos conhecido como *Time-Cost Trade-off Problem* (TCTP). É uma abordagem específica utilizada no planejamento de projetos a fim de proporcionar um balanceamento simultâneo entre o gerenciamento do tempo de conclusão do projeto e o seu custo total (CHOI; PARK, 2015; FENG, LIU; BURNS, 2000).

O TCTP tem sido largamente estudado desde o fim da década de 50 do século XX, quando surgiram as técnicas de programação de rede de projetos conhecidas como *Program Evaluation and Review Technique* (PERT) e *Critical Path Method* (CPM) (KELLEY, 1961; FOLDES e SOUMIS, 1993). Estas técnicas são usadas até hoje e abordam o planejamento e programação de fluxo de rede de projetos, sendo que o fluxo de rede relaciona o encadeamento entre as atividades do projeto com as estimativas de tempo e custo de cada uma delas (FULKERSON, 1961).

Estas estimativas podem ser determinísticas ou estocásticas e são dependentes entre si, pois a variação do tempo de duração de uma atividade interfere no seu custo e vice-versa (GARBI et. al., 2015). O objetivo do TCTP vem da necessidade de comprimir partes do projeto para cobrir eventuais atrasos inesperados ou para cumprir um prazo específico ou ainda conseguir um custo menor ao antecipar o seu término (AMINBACKHSH, 2013).

Os modelos de TCTP resolvem a função objetivo ao aplicar a redução na duração das atividades do projeto (CONTADOR, 1988). Assume-se assim que essa redução implica em maior custo para acelerar a conclusão das atividades (FENG et. al., 2000). O objetivo, portanto, é encontrar a combinação mais adequada de duração e custo que levem à conclusão do projeto, cumprindo prazos e orçamentos específicos (CONTADOR, 1998).

As dificuldades surgem ao se aplicar o modelo em projetos com dezenas ou até centenas de atividades com várias opções de conclusão e custos. Isto cria um problema clássico de análise combinatória para se encontrar as alternativas mais adequadas que satisfaçam a função objetivo (SONMEZ e BETTEMIR, 2012).

O trabalho desenvolvido é um artefato de *software* que sistematiza o algoritmo de TCTP apresentado em Contador (1998) para resolver o método CPM de balanceamento entre tempo e custo.

O artefato possui duas versões para resolver o TCTP, sendo que a primeira versão foi desenvolvida pressupondo-se uma função linear para a evolução do custo marginal de aceleração de cada atividade, e uma segunda versão, com modelagem do custo das atividades por meio de uma função exponencial, é apresentada como contribuição original.

Esta dissertação considera que o ciclo de vida de um TCTP é composto pelas seguintes etapas: estimativas do projeto; correlação entre duração e custo das atividades; algoritmo de aceleração (*Crashing*) para a conclusão das atividades, e, finalmente, simulação e quantificação dos riscos relacionados ao tempo de conclusão e custo total do projeto (GARBI, 2015).

Após o planejamento das atividades do projeto e suas estimativas de duração e custo que se assume estarem nas entradas do artefato, o modelo de TCTP é aplicado. Em seguida, os resultados são submetidos a uma simulação Monte Carlo para quantificação dos riscos, em que os resultados gerados e a simulação fornecem um embasamento adicional às equipes de projetos e gestores na tomada de decisão.

1.1. Objetivo Geral

Propor um artefato de software para aceleração de projetos, demonstrando que as atividades do projeto apresentam resultados diferentes para o custo e duração quando são modeladas com função

custo-duração linear e exponencial convexa, de forma a auxiliar as equipes de projetos nos processos de tomadas de decisões.

1.2. Objetivos Específicos

- a) Desenvolver e implementar o algoritmo para o método de aproximação linear entre o tempo e o custo das atividades do projeto;
- b) Desenvolver e implementar o algoritmo para o método de aproximação exponencial convexa entre o tempo e o custo das atividades do projeto;
- c) Implementar a quantificação dos riscos para as duas versões do artefato de software usando o método de simulação Monte-Carlo.

1.3. Análise do interesse do tema na comunidade Acadêmico-Científica

Uma consulta feita no banco de dados científicos *Web of Science* por meio do portal de periódicos da CAPES, usando como palavra chave o termo *Time Cost Trade off Problem*, nos últimos trinta anos retornou um total de 835 artigos, sendo a maior parte publicada nos últimos vinte anos. O tema foi citado 11.096 vezes, dentre as quais aproximadamente apenas 2,5% são autorreferenciados, como mostra a Tabela 1.1.

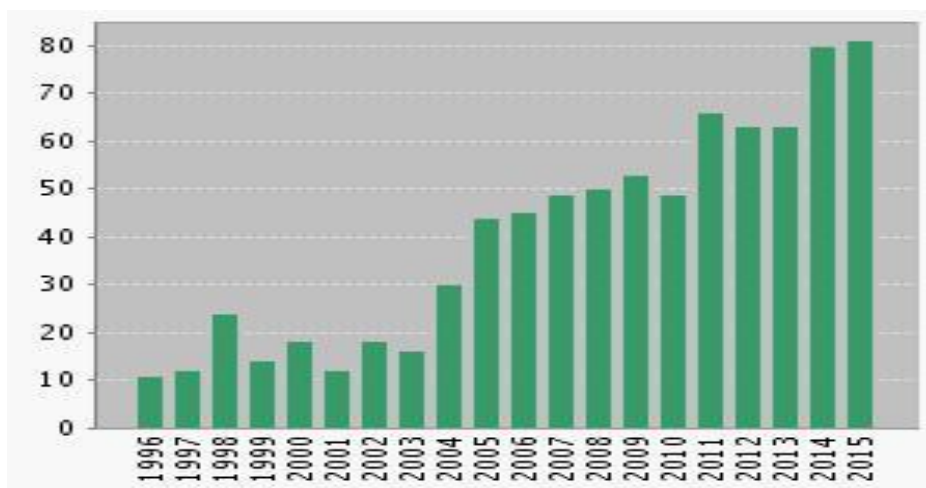
Tabela 1.1 – Interesse do tema na comunidade Acadêmico-Científica

Resultados encontrados	835
Total de citações	11.096
Citações sem auto referência	10.827

Fonte: *Web of Science* (2016).

O gráfico da Figura 1.1 mostra os totais de publicações feitas por ano enquanto que o da Figura 1.2 mostra os totais de citações do assunto por ano.

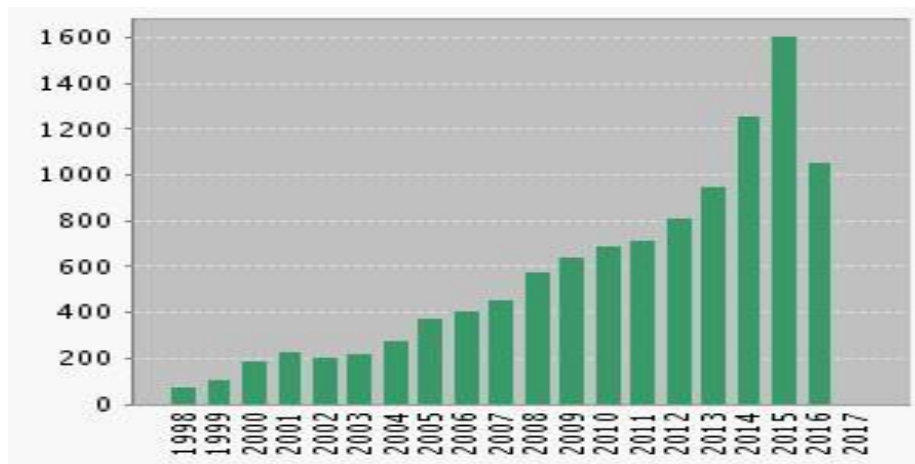
Figura 1.1 – Publicações anuais sobre TCTP nos últimos 20 anos



Fonte: *Web of Science* (2016).

O gráfico da Figura 1.1 mostra que as publicações sobre o assunto TCTP vem crescendo em uma taxa média aproximada de 10% ao ano, pelo menos nos últimos 20 anos.

Figura 1.2 – Total de citações anuais referentes ao tema TCTP nos últimos 20 anos



Fonte: *Web of Science* (2016).

O gráfico da Figura 1.2 mostra que as citações referentes ao assunto TCTP, um pouco diferente do gráfico anterior (Figura 1.1), cresceram em um ritmo quase constante, em que pela figura se vê que começa em aproximadamente 100 citações em 1998 e em 2015, desconsiderando o ano de 2016 em que a consulta foi feita, fica próximo de 1600 citações. Isso dá uma média de crescimento de aproximadamente 17% ao ano, o que corrobora o crescente aumento de interesse na temática.

O interesse da comunidade acadêmica no assunto demonstra que o TCTP é um tema recorrente em que historicamente os pesquisadores têm explorado novas abordagens de programação de projetos com o processo de gerenciamento simultâneo do tempo e do custo das atividades.

1.4. Metodologia Científica e de Implementação

Esta dissertação é pautada em uma pesquisa de abordagem quantitativa que sugere soluções viáveis de compressão de duração de projetos ao menor custo possível.

Esta foi uma pesquisa de natureza aplicada e como resultado são apresentados dois artefatos de software em que são consideradas correlações (i) lineares e (ii) exponenciais convexas para a modelagem do processo de aceleração de um projeto, sendo a última abordagem a contribuição ao tema em estudo.

Os dados referentes à variação do tempo de conclusão dos projetos presentes nos relatórios dos apêndices C ao F, produzidos pelo artefato de software, foram submetidos a uma aplicação externa de planilha eletrônica para quantificar os riscos de sucesso ou fracasso envolvidos na variação das estimativas.

Ressalta-se que esta é uma aplicação prática que pode melhorar os processos de tomada de decisão e revisão ainda na fase de preparação/elaboração/discussão preliminar do projeto.

Como a definição do problema está baseada na pesquisa bibliográfica e documental sobre os TCTP, a dissertação apresenta algumas abordagens que exploraram o problema sobre diversas perspectivas de modelagem e técnicas de solução (ex: heurísticas com aproximações lineares, ou meta-heurísticas com regras independentes da função objetivo tratada), desde os pioneiros avanços na metade do século XX até os dias atuais. A pesquisa mostrou que não existe uma abordagem ótima que seja a melhor em todos os contextos. A abordagem ótima é aquela que melhor se adapta ao escopo, objetivos e configuração do problema.

Finalmente, o modelo desenvolvido foi testado em exemplos encontrados na literatura acadêmica sobre o assunto, o que caracteriza o trabalho desta dissertação como um procedimento experimental.

A sequência básica de implementação foi estudar o método CPM de balanceamento entre tempo e custo que é uma área fundamental da técnica CPM (*Critical Path Method*) e desenvolver um artefato de *software* para aplicar as compressões baseadas em aproximação linear.

O método de compressão, também conhecido como *Crashing*, faz uma análise de custo marginal e resolve o problema de um projeto que precisa ser concluído dentro de um prazo específico e a um custo mínimo, em que algumas atividades podem ser comprimidas (aceleradas) mediante maior gasto monetário (Custo Marginal).

O artefato foi desenvolvido em linguagem de programação C e o ambiente de desenvolvimento usado foi o *NetBeans 8.1* para codificação e testes do artefato.

Da literatura acadêmica sobre o assunto, foram usados dois exemplos de compressão de projetos, com a correlação do tipo linear contínua entre o tempo e o custo da atividade.

Os exemplos foram submetidos ao artefato e os resultados gerados foram comparados e verificados, comprovando a equivalência das soluções encontradas com aquelas fornecidas pela literatura.

Posteriormente, para extrair os dados necessários à quantificação dos riscos, o artefato foi executado com critério de parada ajustado em compressão máxima nos dois exemplos com o intuito de conseguir os limites de tempo máximo e mínimo de conclusão dos projetos. Esses limites gerados pelo artefato foram usados no ambiente de planilhas eletrônicas do *Microsoft Excel*® para a construção dos gráficos de quantificação dos riscos com o apoio da técnica de simulação Monte Carlo. O resultado demonstra situações em que as probabilidades do risco envolvido servem de subsídio aos tomadores de decisão do projeto.

Após o resultado satisfatório dos testes, o artefato foi congelado nesta primeira versão e em seguida foi desenvolvida outra versão presumindo um relacionamento exponencial convexo do custo em função do tempo, em que foi usada a primeira derivada para o custo marginal.

Para os testes e simulação da segunda versão, foram usados os mesmos exemplos da versão anterior, e as comparações e verificações foram feitas por analogia.

1.5. Estrutura da Dissertação

A dissertação está organizada em sete capítulos, conforme descrição a seguir.

O capítulo 1 contém o que já foi descrito até aqui e a descrição da estrutura desta dissertação.

O capítulo 2 traz a revisão bibliográfica do assunto, discorrendo principalmente sobre PERT e CPM.

O capítulo 3 discorre sobre a fundamentação teórica das várias metodologias de resolução dos TCTP, desde as pioneiras às mais atuais.

O capítulo 4 apresenta o desenvolvimento do algoritmo sistematizado em duas versões, o que faz cada passo do algoritmo e a descrição de dois estudos de caso.

O capítulo 5 apresenta e descreve detalhadamente dois estudos de caso.

O capítulo 6 apresenta uma quantificação dos riscos após a execução do *Crashing* nos estudos de caso para cenários aleatórios do tempo de conclusão do projeto, gerados com o apoio da técnica de simulação Monte Carlo.

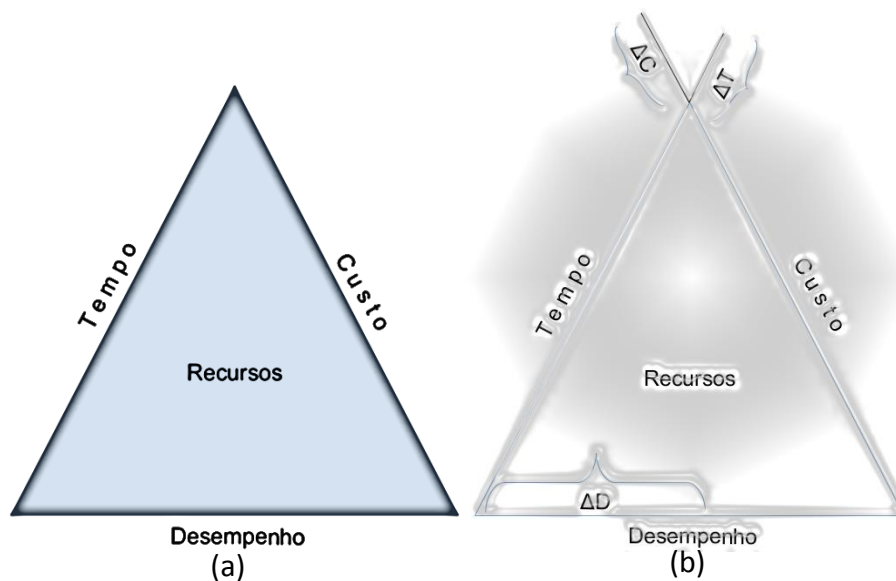
E finalmente o capítulo 7 apresenta as conclusões deste trabalho.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo descreve resumidamente o funcionamento das técnicas PERT e CPM, mostrando as primeiras abordagens de cronograma de rede de projeto e algumas técnicas de otimização de TCTP.

Em gerenciamento de projetos, o que se busca sempre é distribuir da melhor maneira possível os recursos para se atingir o balanceamento otimizado entre tempo, custo e desempenho (CARVALHO e RABECHINI, 2011). Mas, infelizmente, o que é mais comum acontecer são desvios positivos de tempo-custo e desvios negativos de desempenho. Na Figura 2.1, Kerzner (2009) representa ΔT e ΔC respectivamente, como os desvios positivos de tempo e custo, e ΔD como o desvio negativo de desempenho.

Figura 2.1 – (a) Triângulo Tempo/Custo/Desempenho; (b) Desvios de Estimativas no Triângulo Tempo/Custo/Desempenho.



Fonte: Adaptado de Kerzner (2009).

Ao longo dos anos, o gerenciamento de projetos evoluiu, passando a fazer uso de algumas ferramentas de otimização que endereçam principalmente estimativas de tempo e custo, auxiliando os gerentes nas tomadas de decisão para chegarem o mais perto possível de uma solução ótima, minimizando desvios desfavoráveis e maximizando os desvios favoráveis (HILLIER, 2010). Dentre essas ferramentas, as mais famosas

são as técnicas de programação de redes como diagrama de Gantt e PERT/CPM.

2.1. Diagrama de Gantt

O Diagrama de Gantt é uma das técnicas mais antigas e bastante utilizadas na elaboração de cronogramas de projetos. A técnica foi desenvolvida no início do século vinte por Henry Gantt, que elaborou um diagrama mostrando as atividades de projeto dentro de uma escala linear de tempo (KERZNER, 2009).

No Diagrama de Gantt, as atividades são listadas no eixo vertical em que as datas (ou a passagem do tempo), são mostradas no eixo horizontal, e a duração das atividades é mostrada como barras horizontais dispostas conforme as datas de início e fim planejadas (PMI, 2013).

Um exemplo prático do INPE da fase de montagem integração e testes (AIT) do projeto CBERS-3 é mostrado na Tabela 2.1. O cronograma parcial possui 4 fases, não descritas aqui, de montagem, integração e testes dos vários componentes e subsistemas do satélite que foram nominadas como A, B, C e D.

Tabela 2.1 - Cronograma da etapa de AIT do CBERS-3 do INPE

Fase	Tempo de duração em dias => 219	
A	31	
B		54
C		38
D		96

Fonte: Adaptado de INPE (2011).

2.2. PERT- Program Evaluation and Review Technique

Esta é uma técnica usada quando há incertezas nas estimativas individuais de duração das atividades de um projeto. A equipe do projeto fornece três estimativas possíveis e a técnica é usar como estimativa uma média ponderada entre elas (PMI, 2013).

A metodologia PERT foi originalmente desenvolvida entre os anos de 1958 e 1959 pela Marinha dos Estados Unidos, durante a era da “Engenharia de Grandes Projetos” (projeto do míssil Polaris) e veio para preencher lacunas que outras técnicas como Diagrama de Gantt, por exemplo, não conseguiam suprir devido à complexidade e tamanho dos projetos (KERZNER, 2009).

As atividades e os eventos são colocados num grafo que obedecem a sequenciamentos lógicos de precedência e sucessão, em que são deduzidos todos os caminhos possíveis das sequências de atividades que partem do início e vão até o término do projeto (FOLDES e SOUMIS, 1993).

As estimativas são ponderadas em três categorias, que são: pessimista, mais provável e otimista, que caracterizam uma distribuição de probabilidade.

A mais provável tem peso 4 e as outras, peso 1, e a soma ponderada dessas estimativas, dividida por 6, fornece a estimativa a se considerar (KELLEY e WALKER, 1959; CARVALHO e RABECHINI, 2011).

2.3. CPM – Método do Caminho Crítico (*Critical Path Method*)

Este é um método para estimar a menor duração possível do projeto e determinar o grau de flexibilidade dos prazos na sequência lógica dos caminhos da rede, dentro do modelo de programação (PMI, 2013).

Anterior ao PERT, o CPM nasceu por meio da colaboração das equipes de engenharia das empresas norte-americanas *DuPont* e *Remington Rand Univac* focadas em gerenciar projetos de larga escala que envolviam o acompanhamento de atividades como pesquisa, desenvolvimento, desenho, construção, fornecedores, fabricantes e clientes. (KELLEY e WALKER, 1959).

Analogamente, as atividades e os eventos são colocados em um grafo, em que a principal diferença é que no CPM as estimativas são

determinísticas enquanto que no PERT, as três estimativas estão dentro de uma distribuição de probabilidade (CARVALHO e RABECHINI, 2011).

O método do caminho crítico (CPM) leva este nome porque envolve a determinação do caminho mais longo por meio da rede – técnica também incorporada pelo PERT.

O caminho mais longo na rede de eventos e atividades determina o menor tempo de duração para se concluir o projeto, desde que nenhuma mudança seja efetuada nas estimativas de duração das atividades. Desta maneira, o caminho é crítico porque determina o tempo de conclusão do projeto (KELLEY, 1961).

PERT e CPM no início desenvolveram-se separadamente, auxiliando os gerentes a monitorar o progresso de projetos em andamento, mas posteriormente, as melhores características dessas duas técnicas se fundiram no que é conhecido hoje como PERT/CPM (HILLIER, 2010).

2.3.1. Caminho Crítico

O caminho crítico é uma sequência de atividades que representam o caminho de duração mais longo, atravessando o projeto do início ao término, e que determina a menor duração possível do projeto (PMI, 2013).

O desenho da rede contém todas as atividades e as ligações das dependências entre elas, formando todos os caminhos possíveis do início até o término do projeto. O caminho que possui, ou os caminhos que possuírem, a maior duração é o caminho crítico, pois é ele quem determina em unidades de tempo a duração total do projeto (HILLIER e LIEBERMAN, 2010).

A compressão das atividades no caminho crítico reduz a sua duração total, gerando possibilidades de empate com outros caminhos na rede do projeto e, em consequência disto, aumentando o número de caminhos críticos. Como o caminho crítico é o que determina o menor tempo

possível para a conclusão do projeto, é este caminho que deve ser acelerado (CONTADOR, 1988; CARVALHO e RABECHINI, 2011).

Contudo, se existir mais de um caminho crítico, a compressão das atividades nestes caminhos precisa ser feita em paralelo, mas, se apenas um caminho crítico é comprimido, os outros caminhos críticos continuarão determinando o prazo final do projeto, invalidando a compressão única (CONTADOR, 1998), sendo assim, nesse contexto, o problema é de natureza combinatória (DEMEULEMEESTER et. al., 1996).

Há problemas de otimização semelhantes, e um exemplo clássico é o problema do “Caixeiro Viajante”: dadas as localizações geográficas de um número específico de cidades, deve-se encontrar o menor percurso possível para o vendedor, passando por todas as cidades. Estudos recentes têm usado algoritmos baseados em meta-heurísticas para resolver esse problema, sendo que um deles é o *Ant Colony Optimization* (COLORNI et. al., 1991) abordado mais à frente no capítulo 3.

2.4. Redes de Projetos

As redes PERT/CPM (grafos), fornecem uma visualização das relações de dependência entre as atividades e do fluxo de informação entre elas. São desenhos formados por uma série de nós e uma série de arcos que conectam esses nós (CARVALHO e RABECHINI, 2011; PMI, 2013).

Nesse tipo de desenho de rede são necessários três tipos de informação para descrever um projeto, que são:

1. Informação de Atividades: em que o termo *atividade* significa uma porção distinta de trabalho executado durante o andamento de um projeto.
2. Relações de Precedência ou também *Logical Relationship* ou *Precedence Relationship*: dependência entre duas ou mais atividades.
3. Informação de tempo (Duração): número total em unidades de tempo (sem incluir feriados, recessos ou outros períodos não

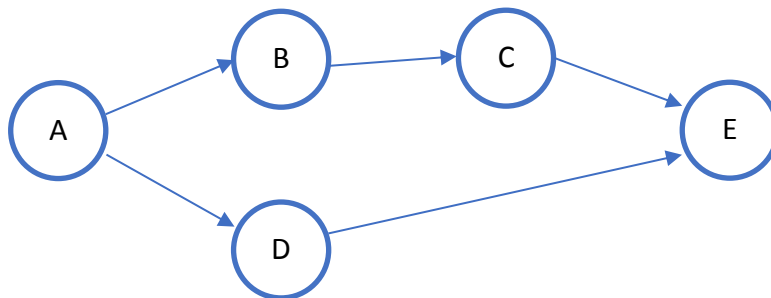
trabalhados) para completar uma atividade ou trabalho planejado.

2.4.1. Desenhos de Redes tipo ANN e ANA

Para passar todas essas informações nos grafos, existem dois tipos de desenho de redes de projeto (PMI, 2013) que são: atividade no nó (ANN) e atividade no arco (ANA).

No tipo de rede ANN como pode ser visto na Figura 2.2, as atividades são colocadas no nó, que no grafo é representado como um círculo. As relações lógicas de dependência entre as atividades descrevendo a sequência em que são executadas são mostradas por arcos de seta. Este termo também é conhecido como PDM (*Precedence Diagram Method*) ou AON (*Activity On Node*).

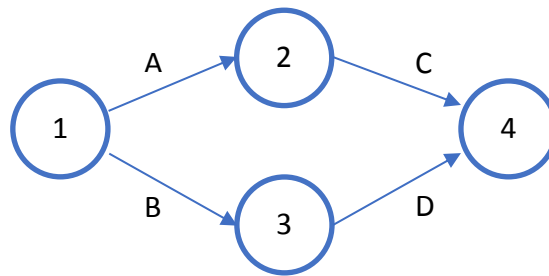
Figura 2.2 – Grafo com tipo de rede ANN



Fonte: Hillier e Lieberman (2010).

O segundo tipo de rede, conhecido como ANA ou AOA (*Activity on Arrow*), descreve a atividade no arco de seta e os nós são usados como eventos de início e fim de um conjunto de atividades. A Figura 2.3 mostra um grafo simples com desenho de rede tipo ANA com quatro atividades e quatro eventos.

Figura 2.3 – Grafo com tipo de rede ANA



Fonte: Elaborado pelo autor (2017).

Quando surgiram, as técnicas PERT e CPM usavam modelos de rede tipo ANA, mas atualmente, o padrão mais comum encontrado é o modelo tipo ANN, sobretudo por possuir algumas vantagens importantes para transmitir a informação da rede, que são: consideravelmente mais fáceis de construir; melhor compreensão para usuários inexperientes, inclusive gerentes, e facilidade na revisão quando há mudanças no projeto (HILLIER e LIEBERMAN, 2010).

2.5. Gerenciamento dos Riscos

A quantificação do risco vem da motivação de que a simulação em computador do algoritmo de *Crashing* pode fornecer resultados que, mesmo podendo ser inferidos por meio de suas premissas, seriam difíceis de reproduzir manualmente (SIMON, 1981).

Portanto, o artefato produzido, ao sistematizar o *Crashing* simulando suas regras em um programa de computador, fornece resultados que podem ser explorados, por exemplo, em um ambiente de múltiplos cenários aleatórios para quantificação de riscos.

Assim, esta subseção não tem como objetivo esgotar a literatura sobre o gerenciamento dos riscos de projetos, mas sim apresentar os principais processos e conceitos bem como as ferramentas aplicadas.

Frequentemente tem-se que tomar decisões baseadas nas informações de que se dispõem, e mesmo quando se tem uma amostra de tamanho

considerável para o apoio na tomada de decisões, a incerteza é uma companheira constante (BERSTEIN, 1997).

2.5.1. Processos do Gerenciamento dos Riscos

Hillson (2004) define duas categorias de incerteza, a epistêmica e a aleatória. A incerteza epistêmica é aquela sobre a qual não se tem conhecimento prévio e não pode ser mensurada, portanto é um evento desconhecido de um conjunto desconhecido de possíveis resultados.

Já a incerteza aleatória é aquela sobre a qual se tem algum conhecimento prévio e se pode mensurar a probabilidade de que aconteça, portanto é um evento anormal extraído a partir de um conjunto conhecido de possíveis resultados.

Basicamente, o conjunto de possíveis resultados é na verdade uma aproximação do evento desconhecido, o que possibilita o seu tratamento por meio das ferramentas probabilísticas (KERZNER, 2009).

De acordo com o PMI (2013), o risco do projeto é um evento ou condição incerta, que se ocorrer, provocará um efeito positivo ou negativo em um ou mais objetivos do projeto e o seu gerenciamento envolve seis processos, descritos a seguir:

- a) Planejar o gerenciamento dos riscos: definição de como conduzir as atividades de gerenciamento dos riscos de forma alinhada com o plano de gerenciamento do projeto;
- b) Identificar os riscos: determinação dos riscos que podem afetar os objetivos do projeto (tempo, custo e escopo, entre outros);
- c) Realizar análise qualitativa dos riscos: priorização dos riscos para análise ou ação adicional por meio da avaliação e combinação de sua probabilidade de ocorrência e impacto;
- d) Realizar análise quantitativa dos riscos: análise numérica do efeito dos riscos identificados, por meio da probabilidade de ocorrência,

nos objetivos do projeto. Frequentemente, a quantificação é realizada utilizando as ferramentas de simulações computacionais;

- e) Planejar as respostas aos riscos: desenvolvimento de opções e ações para aumentar as oportunidades e reduzir as ameaças nos objetivos do projeto.
- f) Controlar os riscos: implementação dos planos de respostas aos riscos, acompanhamento dos riscos identificados, monitoramento dos riscos residuais, identificação de novos riscos e avaliação da eficácia dos processos de gerenciamento dos riscos durante todo o projeto.

2.5.2. Técnicas do Processo de Análise Quantitativa dos Riscos

As análises quantitativas são menos utilizadas pelas empresas, mas quando o são, podem contribuir com resultados que estejam mais próximos da realidade. As ferramentas mais utilizadas são: análise de sensibilidade, análise do valor monetário esperado (EVM)- que é construído sobre árvore de decisão e modelagem/simulação (CARVALHO e RABECHINI, 2011).

Como o artefato de *software* desenvolvido fornece os resultados que são submetidos à quantificação dos riscos, o foco desta próxima subseção é a ferramenta de modelagem/simulação.

2.5.3. Simulação Monte Carlo

O método de simulação Monte Carlo é conhecido desde o século XIX e ficou popular no século XX, por meio do trabalho de John Von Neumann e Stanislaw Ulam, publicado em 1949 com a aplicação do método no projeto da bomba atômica dos Estados Unidos da América (DEVROYE, 1986).

A simulação Monte Carlo é uma ferramenta estatística com aplicações em diversas áreas da ciência e engenharia, cujo método consiste em gerar

uma sequência de números aleatórios para simular a representação de processos estocásticos. (BUSTOS e FRERY, 1992).

Nos métodos convencionais, o processo físico é descrito por meio de equações diferenciais que representam o comportamento de um determinado sistema; já no método de Monte Carlo, o processo é simulado por meio de um conjunto aleatório de valores numéricos referentes aos parâmetros físicos em estudo (RIPLEY, 1987).

Segundo Carvalho e Rabechini (2011), a simulação consiste na construção de um modelo lógico que represente a realidade do problema envolvido e a partir disto, testar o modelo com diversos cenários aleatórios.

Como essa aleatoriedade é conseguida graças a ferramentas baseadas em geradores determinísticos e iterativos, as incertezas do tipo aleatórias na verdade são incertezas pseudoaleatórias, pois os conjuntos de possíveis resultados são aproximações (DEVROYE, 1986).

No entanto, esta dissertação considera que tal pseudoaleatoriedade possui as propriedades necessárias de um número verdadeiramente aleatório, que são uniformidade no intervalo unitário e independência entre as sequências dos valores gerados.

Portanto, o cronograma e o orçamento do projeto podem ser testados em cenários aleatórios para se avaliar a probabilidade de ocorrência dos riscos envolvidos e decidir o que fazer mediante o grau de tolerância da equipe do projeto ao risco (PMI, 2013).

O método de Monte Carlo assume que, quanto maior for o número da amostra da variável aleatória contínua, maior será a precisão da estimativa da probabilidade de ocorrência do risco simulado (DEVROYE,1986).

A estimativa do número mínimo necessário de amostras é dado por:

$$N = \left(\frac{Z \cdot \sigma}{E}\right)^2 \quad (2.1)$$

em que N é o número de amostras, E é a margem de erro da estimativa ou erro máximo, σ é o desvio padrão da amostra, e finalmente Z é o grau de confiança desejado (FONSECA e MARTINS, 1982).

A Tabela 2.2 mostra alguns valores associados ao grau de confiança na amostra.

Tabela 2.2 – Exemplos de grau de confiança e desvio padrão

Grau de Confiança	α	Valor Crítico de Z
90%	0,10	1,645
95%	0,05	1,96
99%	0,01	2,575

Fonte: Elaborado pelo autor (2017).

2.5.3.1. Ferramentas de Simulação Monte Carlo

Existem várias ferramentas comerciais para simular o método de Monte Carlo, sendo que, principais são:

- a) *Operational Risk* da *Numerical Technologies*;
- b) *@Risk* da *Palisade*;
- c) *Crystal Ball* da *Oracle*;
- d) *ModelRisk* da *Vose*;
- e) *Excel* da *Microsoft*;
- f) Módulo *NUMPY* do ambiente *Python*.

3 FUNDAMENTAÇÃO TEÓRICA

3.1. TCTP – Balanceamento entre tempo e custo

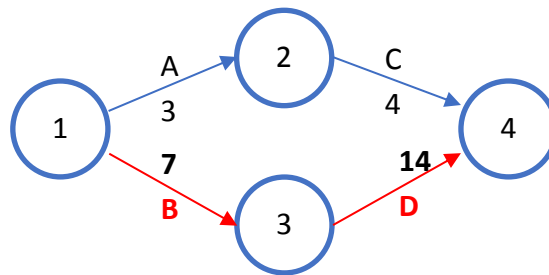
O objetivo principal de todos os envolvidos em um projeto é que ele termine dentro do prazo e do orçamento (AMINBAKHSH, 2013). Em uma rede de projeto, há vários caminhos que partem do início e vão até o término do projeto, sendo que nesses caminhos, cada atividade possui uma data de início e uma data de término. Em cada caminho, uma atividade só pode começar após o término das atividades anteriores (KELLEY e WALKER, 1959).

Portanto podem existir folgas entre as atividades que terminam e as que começam em determinados caminhos. Os caminhos críticos serão aqueles formados pelas atividades que possuírem as menores folgas (iguais a zero se a data programada de término do projeto coincidir com a data da sua última data de término - *due date*) (SIEMENS, 1975).

São as folgas nos caminhos não críticos que podem ser aproveitadas na compressão de duração de projetos. Se houver disponibilidade de aceleração nas durações de algumas atividades dos caminhos críticos, então elas podem ser comprimidas até que as novas folgas nos caminhos não críticos, em relação aos críticos, atinjam o menor valor possível. Caso algumas dessas folgas cheguem a zero, então haverá novos caminhos críticos (CONTADOR, 1988).

A Figura 3.1 mostra o exemplo de um pequeno projeto em desenho de rede tipo ANA que possui 4 eventos representados pelos nós. As atividades e suas durações são colocadas nos arcos de seta.

Figura 3.1 – Rede de Projeto Hipotético com 4 atividades



Fonte: Elaborado pelo autor (2017).

Nesses diagramas, o nó é usado como evento que marca o fim das atividades que convergem para ele e o início das atividades que partem dele. Portanto, o nó inicial possui apenas atividades partindo dele, enquanto que o nó final possui apenas atividades chegando nele. Os números que acompanham as atividades são as durações e o caminho crítico está em vermelho (caminho BD).

Assumindo as durações como unidades de tempo, o caminho AC consome sete unidades de tempo e o caminho BD consome 21 unidades de tempo. Portanto, o caminho AC tem uma folga de 14 unidades de tempo. Assim, após o término da atividade C, o evento 4 - que é o término do projeto - só será atingido ao término da atividade D, consumindo, assim, as 21 unidades de tempo do caminho crítico BD.

Se esse caminho BD puder ser comprimido até a menor folga possível, aproximando-se da duração do caminho AC, o projeto levará menos tempo para ser concluído, podendo compensar o custo adicional da compressão.

3.1.1. Custos envolvidos em Projetos

O custo total de um projeto é composto basicamente por custos diretos e fixos. Os custos fixos são invariantes por unidade de tempo. Aluguéis e seguros são os exemplos mais citados. Já os custos diretos são os que variam com a passagem do tempo, por exemplo, mão de obra e materiais utilizados na execução das atividades (AMINBAKHS, 2013).

O custo indireto do projeto é uma constante resultante do produto entre o custo fixo e a duração total do projeto em unidades de tempo. O custo fixo é estimado pela equipe do projeto e a duração total em unidades de tempo é o valor calculado da duração do caminho crítico (CARVALHO e RABECHINI, 2011).

O custo direto do projeto é a somatória dos custos normais das atividades mais a somatória dos custos marginais das acelerações, se houver.

Custo normal de uma atividade corresponde ao seu mínimo custo direto e está associado ao seu tempo normal. A realização da atividade com tempo superior ao normal não leva a uma redução significativa do seu custo direto.

Define-se também custo acelerado para cada atividade, o qual está relacionado com seu tempo acelerado que é o mínimo tempo em que é exequível realizar a atividade. A realização da atividade com tempo inferior ao acelerado seria impossível ou acarretaria custos adicionais proibitivos.

Já o custo total do projeto é calculado somando-se os custos diretos de cada atividade, mais o custo indireto (CONTADOR, 1998).

3.2. Métodos de resolução do TCTP

Os problemas de TCT envolvem principalmente métodos de otimização exatos, heurísticos e meta-heurísticos. Não existe um método que funcione da melhor maneira para todas as situações, cada método possui vantagens e desvantagens que devem ser analisadas (AMINBAKSHI, 2013).

As primeiras abordagens de TCT partiam do pressuposto de um relacionamento linear entre as grandezas custo e tempo (FORD e FULKERSON, 1956; FULKERSON, 1961), e, posteriormente, começaram a ser considerados relacionamentos não lineares, como por exemplo a curva convexa adotada por Feng et al (1997) e outros.

O pioneiro a abordar o problema heurísticamente foi Siemens (1971) com o seu método de aproximação conhecido como *Siemens Approximation Method* que faz uma aproximação linear por partes.

Os métodos exatos garantem a solução ótima, mas, em contrapartida, consomem mais recursos computacionais conforme aumenta a complexidade da rede e o número de atividades. Alguns exemplos mais populares são a programação linear e a programação dinâmica (FULKERSON, 1961; DEMEULEMEESTER et. al., 1993).

Os métodos heurísticos envolvem bem menos esforço computacional. Eles encontram uma solução próxima da ótima ao usar métodos heurísticos, que geralmente possuem regras simples de abordagens para problemas de otimização complexos (SIEMENS, 1971; FENG et. al., 2000).

Os métodos meta-heurísticos diferem dos heurísticos por não serem dependentes da natureza do problema abordado, pois utilizam técnicas de busca aleatória ao simular comportamento e inteligência de organismos naturais.

Esta busca pode propiciar a fuga de uma região de convergência ótima local para outras regiões de convergência a serem exploradas com soluções viáveis que poderiam não ser encontradas com a heurística simples.

O método meta-heurístico, igualmente ao heurístico, também não garante uma solução ótima, mas chega a uma solução viável próxima mais rápido que as abordagens exatas. Os principais estudos atuais que consideram esses métodos são a computação evolutiva e inteligência de enxame. Alguns exemplos são os Algoritmos Genéticos (GA), Colônia de Formigas (ACO), Enxame de Partículas (PSO), entre outros (FENG et. al., 2000; AMINBAKHSH, 2013).

3.3. Métodos Heurísticos

3.3.1. Método de Aproximação de Siemens

Siemens (1971) desenvolveu um algoritmo que leva o seu nome (*Siemens Approximation Method – SAM*), cujo método faz aproximações não lineares usando o coeficiente angular (inclinação ou custo marginal), calculado a cada iteração (AMINBAKSH, 2013). Este método baseia-se nos seguintes passos:

1. Construir da rede inicial do projeto;
2. Identificar os caminhos da rede;
3. Calcular os custos marginais das atividades da rede a cada iteração;
4. Calcular a duração de cada caminho;
5. Escolher o caminho mais longo como caminho crítico. Havendo empate, decidir pelo caminho com menor somatória das inclinações dos custos;
6. Escolher a atividade com menor inclinação no caminho crítico selecionado. Havendo empate, decidir pela atividade comum ao maior número de caminhos. Se ainda houver empate, usar a atividade que permita maior compressão;
7. Acelerar a atividade com a compressão de tempo disponível;
8. Executar os passos 3 ao 7 até comprimir todas as atividades do caminho crítico selecionado.

Siemens (1971) afirma que os estudos de algoritmos heurísticos, análogo a procedimentos exatos, têm dificuldade em lidar com problemas de grande escala. A maioria das heurísticas presumem relações tempo-custo meramente lineares e elas ficam aquém de entregar o conjunto de possíveis soluções (FENG et al., 1997; ZHENG et al., 2005).

O método SAM converge para a função objetivo ao fazer várias aproximações não lineares (menor custo marginal calculado a cada iteração), mas como considera apenas o menor custo sem verificar as restrições dos caminhos aos quais as atividades pertencem, pode comprimir além do tempo desejado (AMINBAKSH, 2013).

3.3.2. Método de Aproximação Linear de Contador

Contador (1998) propõe um método de convergência em um algoritmo que se baseia em relacionamento linear do custo em função do tempo. O autor argumenta que embora o método não possa garantir a solução ótima por não cobrir todas as possibilidades de compressão nos conjuntos escolhidos, ele é mais simples que os outros métodos de resolução dos problemas de TCT, pois o enfoque é voltado para o gerenciamento de projetos.

Uma das características que tornam o problema mais simples é a aproximação linear sobre a evolução dos custos diretos das atividades do projeto quando o tempo de duração destas atividades é diminuído. Na aproximação linear o custo marginal é calculado como a razão da variação do custo sobre a variação do tempo (DEMEULEMEESTER, 2002).

Isto nada mais é do que o coeficiente angular da reta compreendida entre os pontos inicial e final do custo e do tempo. Por essa razão, alguns autores às vezes chamam por inclinação este custo marginal (SIEMENS, 1971).

O algoritmo baseia-se em sete passos:

1. Identificar os caminhos críticos;
2. Identificar o conjunto de atividades a serem aceleradas;
3. Escolher o conjunto de aceleração com menor custo marginal;
4. Determinar o tamanho do passo de aceleração;

5. Aplicar a redução na rede de atividades;
6. Calcular os custos, e
7. Repetir o processo enquanto o custo marginal for menor que o custo fixo do projeto ou o prazo final pretendido não for alcançado.

O algoritmo converge para a função objetivo ao executar compressões nas durações dos caminhos críticos. As compressões se baseiam nas folgas dos caminhos que não são críticos, na disponibilidade de compressão das atividades dos caminhos críticos e na análise do custo marginal envolvido na compressão de cada atividade (CONTADOR, 1988 e 1998).

Para garantir a solução ótima, precisariam ser verificadas todas as possibilidades de compressão em todos os caminhos críticos. Cada caminho crítico poderia ter algumas atividades aceleradas e outras desaceleradas, cumprindo todas as possibilidades de compressão para garantir a verificação da solução ótima. Contador (1998) argumenta que isto geralmente não acontece no contexto do gerenciamento de projetos e que, portanto, partindo dessa premissa, o algoritmo sempre encontrará a solução ótima.

Como o algoritmo foi sistematizado, ele é visto com mais detalhes no capítulo 4.

3.4. Métodos Meta-Heurísticos

3.4.1. Algoritmos Genéticos (*Genetic Algorithm - GA*)

A metáfora por trás dos algoritmos genéticos refere-se à Seleção Natural da teoria da Evolução das Espécies do naturalista britânico Charles Darwin (FENG et. al., 2000). Essas abordagens simulam a evolução genética por meio do comportamento cromossômico.

Os estudos mostram um problema conhecido neste meio como *Genetic Drift*, que ocasiona redução da variação genética. Como é um efeito

indesejado na evolução genética e, portanto, indesejado também no ambiente simulado com essa meta-heurística, as abordagens contornam o problema, cada uma a seu modo (AMINBAKHSH, 2013).

Feng et al. (2000) propuseram essa abordagem baseada no Algoritmo Genético de Holland (1975) para tentar vencer a dificuldade computacional que envolve os problemas de TCT de grande escala. São preparados dois cromossomos, sendo um a sequência normal (todas as atividades com o par tempo-custo normal) e o outro, a sequência acelerada (todas as atividades com o par tempo-custo acelerado). Em seguida, são selecionados os melhores valores de solução baseados nas menores distâncias deles em relação à curva convexa. Em seguida, são executados cruzamentos e mutações de forma iterativa a fim de gerar novas soluções mantendo os elementos anteriores para evitar a redução da variação genética (AMINBAKHSH, 2013).

Feng et al. (2000) argumentam que o modelo foi capaz de descobrir mais de 95% das soluções ótimas em problemas de TCT discretos com 18 atividades.

Zheng et al. (2005) propuseram um modelo que reduz a chance de ficar preso a um ótimo local, não ficando, assim, refém do problema de redução de variação genética. Os autores usam a MAWA (*modified adaptive weight approach*) ajustando a prioridade dos objetivos com relação à qualidade da geração anterior.

O modelo MAWA sinaliza e classifica todas as soluções identificadas em meio a cada geração e marca as soluções mais propensas a sobreviver nas futuras gerações. Desta maneira, à medida que as gerações evoluem, o modelo MAWA diminui a taxa de mutação para evitar a convergência prematura.

O modelo incorpora a Fronteira de Pareto como critério de seleção e formação de nichos como diversificador de população para garantir uma amostragem uniforme. Ressalta-se que foram usadas as mesmas

instâncias de atividades utilizadas por Feng et al. (1997) e o modelo, segundo os autores, mostrou-se robusto.

3.4.2. Otimização por Enxame de Partículas (*Particle Swarm Optimization - PSO*)

A otimização por enxame de partículas utiliza o conceito de inteligência de um enxame de organismos vivos quando buscam um determinado objetivo. Os organismos podem ser por exemplo, um bando de pássaros ou um enxame de abelhas. Nesta meta-heurística o coletivo sempre é tratado como enxame (YANG, 2007).

No enxame há a inteligência individual de cada elemento que no modelo é chamada de *pbest*, e há também a inteligência coletiva do enxame que é chamada de *gbest*. O comportamento do enxame será resultado do balanceamento entre as dominâncias dessas duas inteligências, a individual e a coletiva (AMINBAKSHI, 2013).

Yang (2007) utilizou este modelo de otimização em cuja abordagem são geradas e arquivadas Fronteiras de Pareto, das soluções mais viáveis para análise posterior. São sinalizadas e selecionadas as melhores soluções locais *pbest* quando estas surgem com forte dominância, e as melhores soluções globais *gbest*.

Como o algoritmo não utiliza métricas, é capaz de lidar com vários tipos de funções objetivo, independentemente do tipo de relacionamento custo-tempo. As soluções (partículas) dominantes são armazenadas de forma iterativa em uma biblioteca separada enquanto as partículas dominadas são excluídas e os membros desta biblioteca ajudarão em posteriores explorações no espaço do problema. Os custos indiretos não são considerados ao longo do processo de otimização e são fornecidos externamente apenas após a Fronteira de Pareto final.

Uma abordagem por Enxame de Partículas (*PSO*) chamada *Fuzzy Multi Objective PSO* (FMOPSO) por ser baseada em variáveis nebulosas foi proposta por Zhang e Xing (2010) para resolver o balanceamento entre

tempo, custo e qualidade com dados de entrada determinísticos. Foi incorporada a técnica de múltiplos atributos nebulosos às operações da aritmética nebulosa para melhorar o algoritmo *PSO*, explorando, desta forma, soluções com máxima qualidade e também tempo e custo mínimos.

3.4.3. Otimização por Colônia de Formigas (*Ant Colony Optimization* – *ACO*)

Esta abordagem baseia-se no comportamento das formigas quando elas saem da colônia trilhando caminhos aleatórios em busca de alimento. Quando uma formiga encontra alimento, ela retorna à colônia para transmitir a informação e faz isso marcando o caminho com feromônios. Quando outras formigas da colônia passam por esse caminho marcado, começam a segui-lo e a reforçar a marcação com feromônio (COLORNI et al., 1991).

Ng e Zhang (2008) propuseram um Sistema de Colônia de Formigas para o problema TCT multiobjetivo adotando paralelamente a abordagem de peso adaptativo modificado (MAWA), que verifica a viabilidade das soluções encontradas. Utilizando as mesmas instâncias de 18 atividades de Feng et al. (1997), a abordagem proposta mostrou-se eficiente.

Também para TCTP discreto multiobjetivo, Afshar et al. (2009) propuseram múltiplas colônias de formigas para múltiplos objetivos.

As soluções encontradas em cada colônia são transferidas iterativamente para a próxima colônia a ser avaliada de acordo com o objetivo concorrente e novas soluções são então geradas em relação à trilha atualizada de feromônio.

A cada iteração do algoritmo arquivam-se as soluções dominantes. Ressalta-se que o desempenho do modelo se mostrou melhor que outros algoritmos principalmente quando os custos indiretos são menores (AMINBAKSH, 2013).

3.4.4. Otimização Híbrida

Ashuri e Tavakolan (2012) propuseram um algoritmo de modelo híbrido baseado em Enxame de Partículas (PSO) e Algoritmos Genéticos (GA) para otimização de problemas voltados à construção civil. Utiliza-se também a teoria dos conjuntos nebulosos para analisar problemas de balanceamento entre tempo, custo e recurso, em que para essas variáveis, pressupõem-se valores triangulares não determinísticos.

Os indivíduos de uma nova geração são criados não apenas por simulação de cruzamento e mutação como nos algoritmos genéticos (GA), mas também por enxame de partículas (PSO).

A metáfora da abordagem adotada é o conceito da estratégia de elites. De maneira iterativa, a população é separada em duas metades e os indivíduos são analisados. Na primeira metade, chamada de Elite, ficam os indivíduos mais próximos da função objetivo. Os indivíduos restantes ficam na segunda metade. Para melhorar a população em uma próxima geração, é aplicado o PSO na primeira metade, o GA na segunda metade e o processo se repete.

Os autores argumentam que o híbrido proposto (GA-PSO), mostrou-se eficiente gastando poucos recursos computacionais se comparado a outros modelos.

A Tabela 3.1 mostra resumidamente alguns detalhes destas abordagens, com exceção do algoritmo de Contador (1998) abordado no capítulo 4. As colunas preenchidas com (Ni) significam que os conteúdos não foram informados.

Tabela 3.1 – Detalhes, e características das abordagens de TCTP)

Ano-Autor	Método	Problema	Nº de Atividades	Tempo Segundos	Características	Deficiências
1971-Siemens	Heurístico SAM	Tempo Custo	5	Ni	Considera regras para acelerar atividades com menor custo adicional	Pode comprimir além do necessário por considerar apenas o menor custo
1997-Feng, Liu e Burns	Algoritmos Genéticos	Tempo Custo	18	Ni	Calcula valores elegíveis explorando mínimas distâncias à borda convexa e mantendo cada conjunto na próxima geração para evitar a redução da variação genética	Manipula apenas relacionamentos Finish-to-Start e não considera restrição de recursos
2005-Zheng, Ng and Kumaraswamy	Algoritmos Genéticos	Tempo Custo	18	Ni	Manipula Abordagem Ponderada Modificada (MAWA), Ranking de Pareto e Formação de Nichos para evitar a redução da variação genética, seleciona modelos e usa diversificadores	Ni
2007-Yang	Enxame de Partículas	Tempo Custo	8 e 28	48 e 600	É capaz de lidar com qualquer tipo de função que requeira cálculo manual para análise posterior de cenários What-if	Não leva em consideração os custos indiretos
2008-NG and Zhang	Colônia de Formigas	Tempo Custo	18	Ni	Sistema de ponderação MAWA integrado num Sistema de Otimização de Colônia de Formigas	Probabilidade de convergência prematura com alto número de iterações e muito sensível à seleção de parâmetros
2009-Afshar, Ziaraty, Kaveh and Sharifi	Colônia de Formigas	Tempo Custo	18	Ni	Múltiplas Colônias sem dominância (NA-ACO) são associadas para diferentes objetivos avaliando os resultados de acordo com a concorrência legada à próxima colônia	Ni
2010-Zhang and Xing	Enxame de Partículas	Tempo Custo e Qualidade	13	Ni	Baseado em variáveis nebulosas. Faz análise de qualidade e usa utilitário de atribuição nebulosa para geração de valores	Gera apenas uma solução otimizada ao invés de uma Fronteira de Pareto
2012-Ashuri and Tavakolan	Híbrido – Algoritmo Genético e Enxame de Partículas	Tempo Custo	7 e 14	348 e 1140	Modelo híbrido com tratamento de recursos. Divide a população ao meio e emprega GA na região inferior e PSO na região superior	Usa apenas funções contínuas

Fonte: – Adaptado de AMINBAKHSH (2013).

4 DESENVOLVIMENTO DO ARTEFATO DE SOFTWARE PARA SIMULAÇÃO DO PROCESSO DE *TIME COST TRADE-OFF*

A proposta do trabalho é fornecer uma ferramenta de simulação que auxilie os gerentes de projeto na tomada de decisão mediante relatórios com diferentes representações para levar o projeto ao seu final acompanhando a relação custo-benefício das otimizações.

O artefato de software é uma ferramenta computacional que executa o trabalho de combinar todas as possibilidades de compressões das atividades dos caminhos críticos, dentro de prazos estipulados e a um custo total mínimo. Posteriormente, os dados gerados pelo artefato são compilados e submetidos a um ambiente de simulação Monte Carlo para quantificar os riscos e analisar as probabilidades de conclusão do projeto em múltiplos cenários aleatórios.

4.1. Sistematização do Algoritmo de Contador

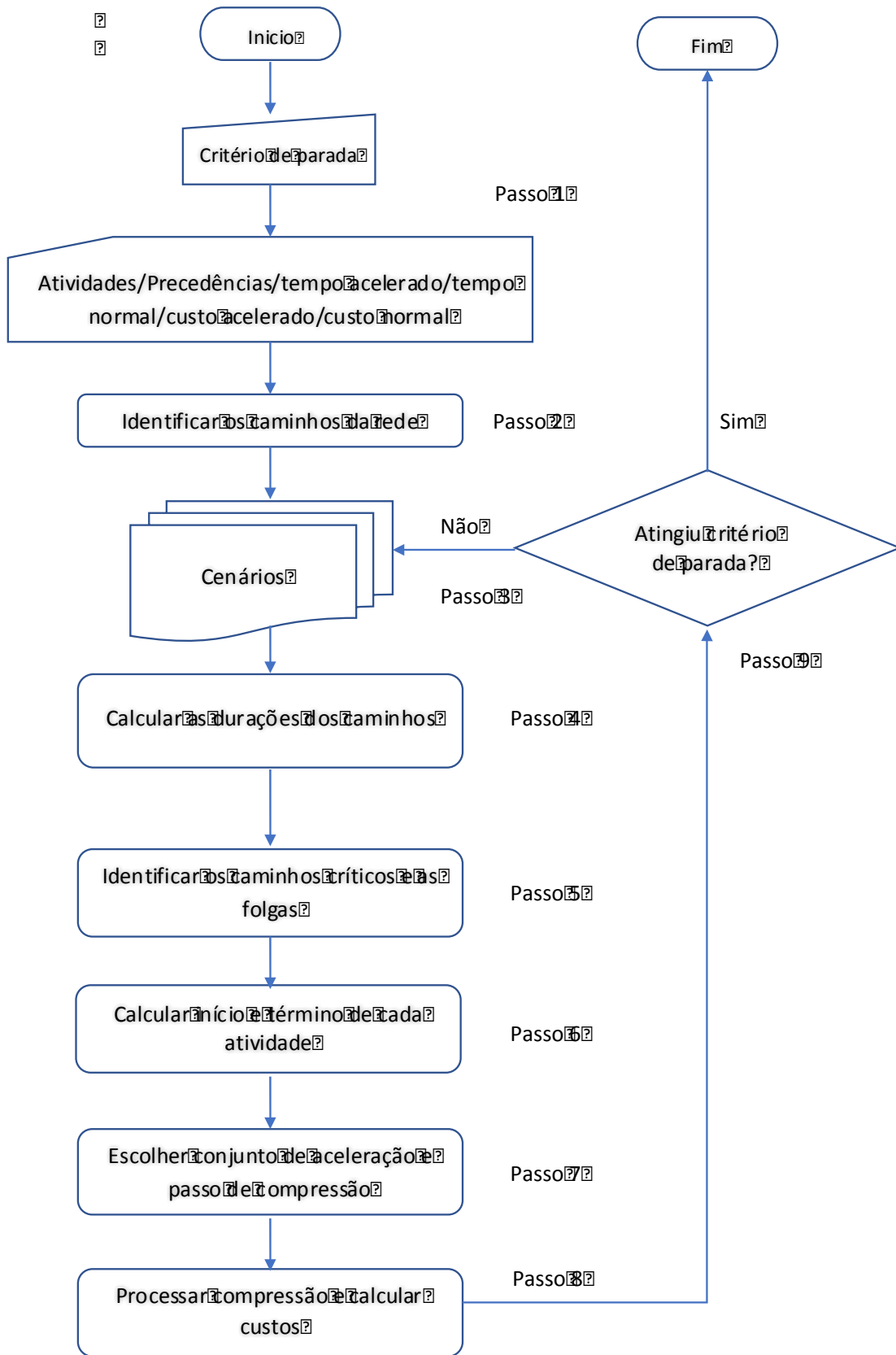
O artefato desenvolvido nesta dissertação possui duas versões de relacionamento entre as estimativas de tempo e custo, sendo: uma versão linear e outra exponencial convexa.

A versão linear é uma sistematização do algoritmo de *Crashing* de Contador (1998) e como contribuição original foi usada uma função exponencial convexa para a aproximação do custo em função do tempo quando este último é diminuído.

A sistematização se inicia com o artefato sendo executado com o critério de parada e o custo fixo como parâmetros de chamada. Em seguida o artefato lê a tabela de entrada com as atividades, relações de precedência e os valores das estimativas de durações e custos.

O artefato infere todos os caminhos possíveis na rede do projeto, varrendo uma lista de atividades e precedências montada através de uma tabela de entrada e em seguida mostra os cenários corrente e posteriores a cada compressão até atingir o critério de parada. O fluxograma da Figura 4.1 mostra os passos do algoritmo.

Figura 4.1 – Fluxograma Geral do Algoritmo



Fonte: Elaborado pelo autor (2017)

4.1.1. Passo 1 – Ler dados de entrada e escolher o critério de parada

O artefato é executado com dois ou três parâmetros de chamada, cuja variabilidade depende da escolha do critério, que por sua vez pode ser *custo* ou *prazo*. Quando a escolha é *custo* então assume-se apenas mais um parâmetro subsequente que é o valor do *custo fixo* do projeto por unidade de tempo. Se o critério escolhido é *prazo* então assume-se mais dois parâmetros subsequentes, que são o *prazo final* predeterminado para o término do projeto e o valor do *custo fixo* por unidade de tempo.

Com o critério de parada *custo*, o artefato comprime iterativamente a duração do projeto enquanto conseguir reduzir seu custo total. À partir do momento em que a redução de custos não é mais possível, a compressão atingiu o ponto de custo mínimo e o critério de parada é satisfeito.

Em contrapartida, o critério de parada *prazo* faz o artefato comprimir iterativamente a duração do projeto, sempre ao menor custo marginal, até atingir o momento exato ou o mais próximo possível do prazo final pretendido que foi passado na chamada de execução. Nesse caso o *prazo final* igual a zero resultará em máxima compressão no tempo de duração do projeto.

Após a verificação dos parâmetros de chamada, o artefato assume que os dados de entrada do projeto são provenientes de uma tabela compilada com as atividades, suas respectivas precedências e estimativas de tempo e custo.

O primeiro registro de entrada que é a primeira linha da tabela, deverá conter o total de registros da própria tabela exceto o primeiro. Os demais registros conterão a cada linha, a atividade, suas predecessoras, duração normal, duração acelerada, custo normal e custo acelerado.

O artefato então lê a tabela até o final, armazenando os dados em memória. A entrada de dados pode ser verificada na função (EntrarDados) do artefato nos Apêndices A e B.

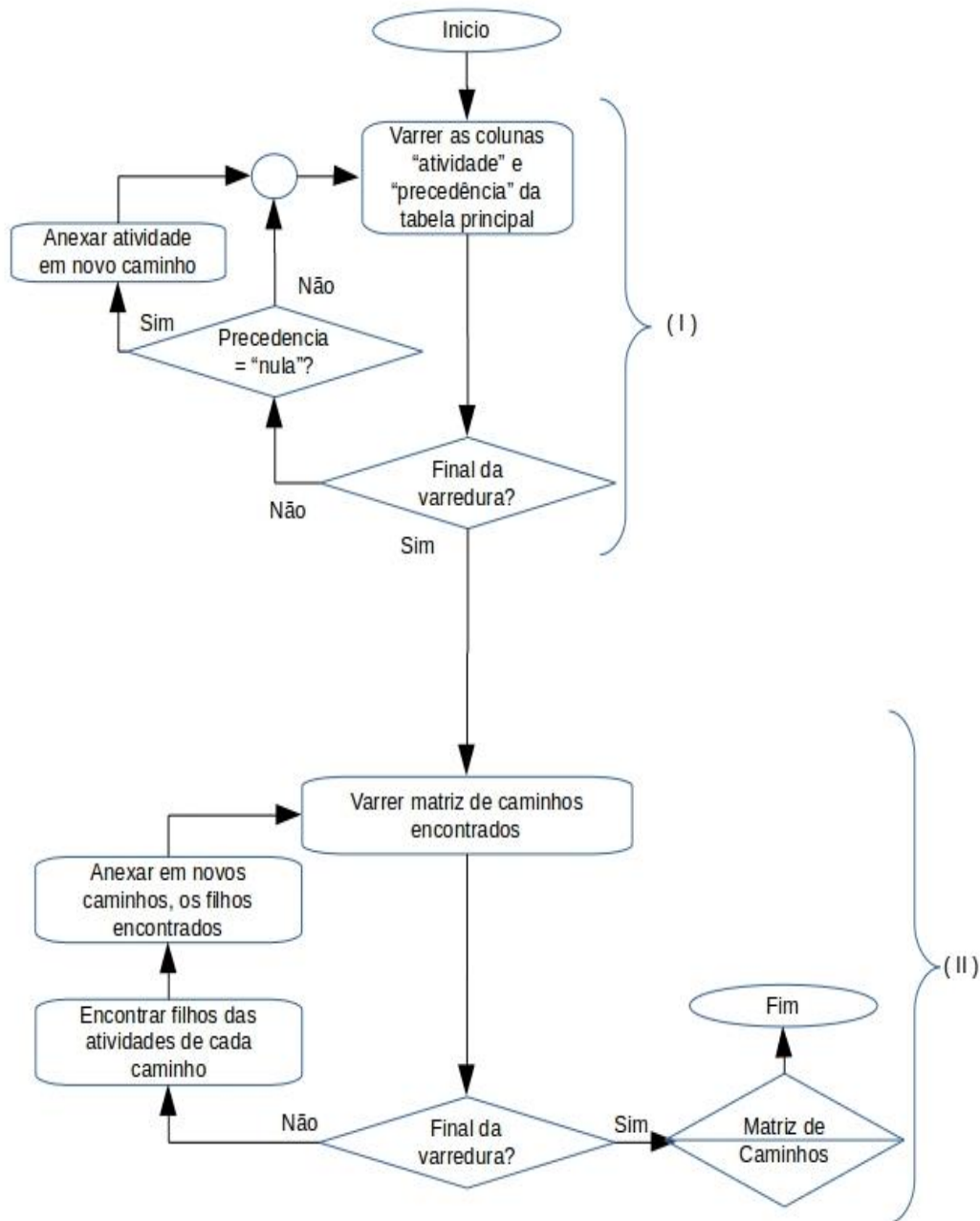
4.1.2. Passo 2 – Identificar os caminhos da rede

Neste modelo de representação, o artefato pressupõe que a cada registro na tabela, qualquer atividade predecessora ocorrerá antes de qualquer atividade sucessora. Ou seja, se por exemplo uma atividade índice (j) depende de uma atividade índice (i), então (i) é menor que (j) na varredura da tabela.

Esse passo identifica todos os caminhos possíveis da rede do projeto, partindo do início, em que se pressupõe a existência de atividades com precedência nula, indo até o final, em que se pressupõe a existência de atividades sem sucessoras.

Ressalta-se que os caminhos encontrados são armazenados em uma matriz de caminhos possíveis do projeto. A Figura 4.2 mostra o fluxograma do passo 2 e o módulo de inferência dos caminhos possíveis pode ser verificado nas funções (EncontrarCaminhos) e (EncontrarFilhos) do artefato nos Apêndices A e B.

Figura 4.2 – Fluxograma do passo 2: Identificar os caminhos da rede



Fonte: Elaborado pelo autor (2017).

Conforme mostra a figura, o bloco iterativo (I) executa uma varredura nas colunas de atividade e precedência nas ocorrências da tabela principal de entrada.

As atividades que possuírem o campo de precedência preenchido como nula, serão origens de novos caminhos e o final deste bloco, haverá uma matriz inicial com os primeiros caminhos possíveis.

Na sequência do fluxo, o bloco iterativo (II) lê a matriz inicial de caminhos possíveis gerados no bloco anterior e faz uma varredura em cada vetor de caminho.

Para cada vetor de caminho, este bloco lê a última posição deste vetor, que é uma atividade e procura por atividades sucessoras, chamadas aqui de filhos. Essa última posição contém a atividade que será, ou a atividade final de um caminho ou a origem de novos caminhos se ela possuir filhos como atividades sucessoras.

No caso de serem encontrados filhos, eles estarão no vetor chamado filhos e serão anexados em novos caminhos possíveis.

Na sequência do fluxo, antes do fim do bloco iterativo que marca o fim do fluxograma foram encontrados todos os caminhos possíveis que partem do início e vão até o fim do projeto. Embora seja um procedimento desnecessário, para facilitar a leitura posterior nos relatórios, a matriz com todos os caminhos é classificada em ordem crescente de caminhos.

4.1.3. Passo 3 – Gerar Relatório do Cenário

Neste passo gera-se um relatório em texto do último cenário registrado em memória com as atualizações de tempo e custo das atividades, suas datas de início e término e os caminhos do projeto com as durações e folgas atualizados. Como este passo não possui lógica intrínseca ao TCTP, o fluxograma não é apresentado mas o módulo gerador do relatório de saída é descrito na função (GravarSaida) do artefato nos apêndices A e B.

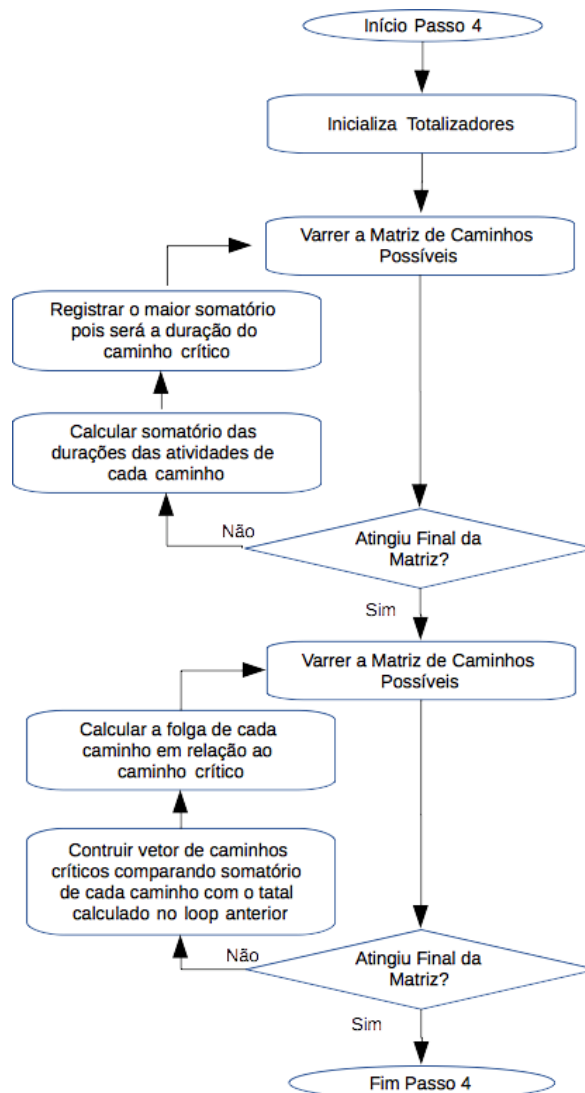
4.1.4. Passos 4 e 5 - Calcular as durações de cada caminho identificando caminhos críticos e folga de cada caminho

O *Crashing* é a técnica usada para comprimir a duração de um cronograma com o menor custo incremental possível ao se adicionar recursos (PMI, 2013). Como os caminhos possíveis já foram identificados, agora é necessário calcular as durações de todos os caminhos, que é a

soma das durações das atividades de cada caminho. Feito isso, a maior duração determinará o caminho crítico, que pode ser mais de um.

Este passo lê a matriz dos caminhos possíveis que foram descobertos no passo 2 e totaliza as durações das atividades. O caminho que atravessa o projeto do início ao fim e que for maior (pode ser mais de um) será o caminho crítico. E em seguida, a matriz é lida novamente para se construir um índice dos caminhos críticos e calcular as folgas de cada caminho. A Figura 4.3 mostra esses passos e o módulo pode ser verificado na função (Duracoes_E_Folgas) do artefato nos apêndices A e B.

Figura 4.3 - Fluxograma do cálculo das durações e folgas dos caminhos



Fonte: Elaborado pelo autor (2017).

4.1.5. Passo 6 – Calcular início e término de cada atividade

Calcular o início e término de cada atividade significa encontrar a primeira data de início e a última data de término, conhecidas como *data cedo* e *data tarde*.

A *data cedo* é a data mais cedo possível (primeira data de início), em que porções incompletas de uma atividade do cronograma podem ser iniciadas baseadas na lógica do cronograma e nas informações e restrições de prazo (PMI, 2013).

A *data tarde* é a data mais tarde possível (última data de término), em que porções incompletas de uma atividade do cronograma podem ser terminadas baseadas na lógica do cronograma, restrições e prazo final do projeto (PMI, 2013).

Essas variáveis não interferem diretamente no *Crashing* do artefato, pois ele trabalha de maneira independente com a folga de cada caminho em relação à duração do caminho crítico. Todavia, é uma informação relevante que mostra ao gerente do projeto as datas de início e fim das atividades.

Para determinar a *data cedo* e a *data tarde* de cada evento na rede, o método CPM utiliza a lógica de programação para frente e para trás ao longo do desenho da rede do projeto. (CARVALHO, 2011; PMI, 2013).

4.1.5.1. Programação para Frente e para Trás

Na programação para frente, parte-se do evento origem e percorre-se o desenho da rede do projeto até o término. Nesse processo, calcula-se a primeira data de início de cada evento (*data cedo*) como a soma parcial do caminho de maior duração dentre os caminhos que convergem para esse evento.

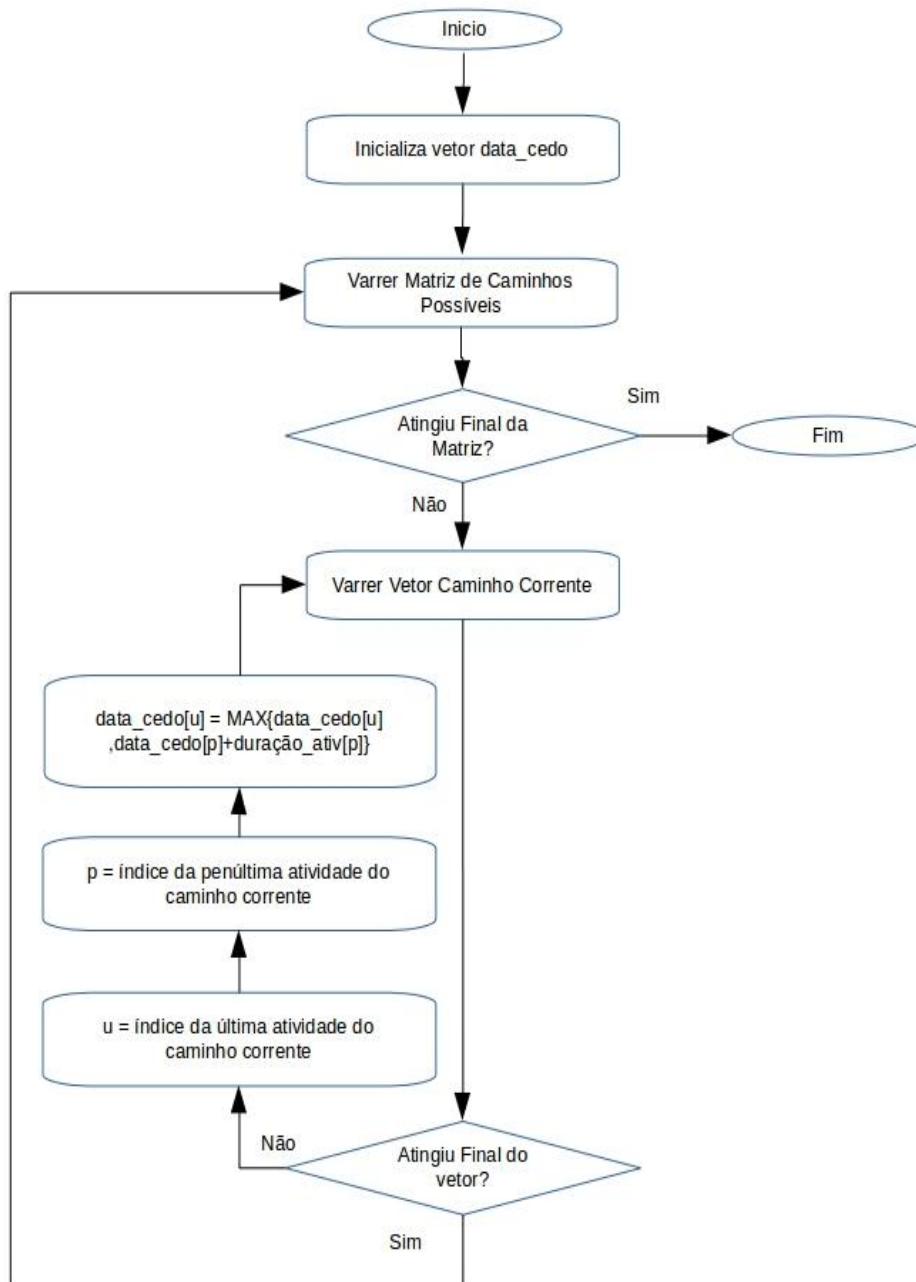
Apenas a completude do evento garante o início das atividades que partem dele, sendo que esta completude só é alcançada totalmente quando todos os caminhos que convergem para o evento tiverem

terminado. Portanto, o caminho de maior duração garante que não há atividades em andamento impedindo a completude do evento (CONTADOR, 1998).

No fluxograma da Figura 4.4, o vetor chamado *data cedo* possui uma ocorrência para cada atividade do projeto. Como a *data cedo* é o valor do caminho parcial de maior duração que converge para a atividade, então basta somar a duração das atividades predecessoras de uma atividade em um caminho e pegar o valor máximo entre esta soma parcial mais o valor que está no vetor *data cedo* desta atividade.

Ao se fazer isto em todos os caminhos, tem-se a primeira data de início de todas as atividades no vetor *data cedo*. A Figura 4.4 mostra o fluxograma. Este módulo é descrito na função (DataCedo) do artefato nos Apêndices A e B.

Figura 4.4 – Fluxograma da programação para frente



Fonte: Elaborado pelo autor (2017).

Na programação para trás, percorre-se o desenho da rede do projeto no sentido inverso, partindo do evento de término até o evento de origem. Neste processo, calcula-se a última data de término de cada evento (*data tarde*) como a somatória parcial do caminho de menor duração dentre os caminhos que convergem no sentido inverso para esse evento.

Sendo assim, nesta programação para frente e para trás, além das datas, precisa-se identificar também as folgas nos caminhos da rede do projeto. Portanto, o caminho de menor duração garantirá a máxima flexibilidade de prazo possível (data tarde), dentre as folgas dos caminhos que retroagem de um evento sucessor para um predecessor (CONTADOR, 1998).

Este módulo é descrito na função (DataTarde) do artefato nos apêndices A e B.

Como esse procedimento de programação para frente e para trás não é um problema intrínseco ao *Crashing*, foi mostrado aqui apenas o fluxograma da programação para frente.

4.1.6. Passo 7 – Escolher conjunto candidato a aceleração

Segundo a lógica do algoritmo, a escolha de um conjunto candidato a aceleração requer a geração de todas as combinações possíveis de atividades aceleráveis, desde que esse conjunto atenda alguns requisitos, que são: ele não pode ter redundância; deve atender todos os caminhos críticos e, cada caminho crítico deve ser atendido com apenas uma atividade. Portanto, antes de gerar as combinações, este passo executa algumas preparações necessárias.

Primeiramente, as atividades dos caminhos críticos deverão ser colocadas todas em um vetor sem redundâncias para atender o primeiro requisito. Por exemplo, se há dois caminhos, em que um é (ABC) e o outro é (ABD), deve-se fazer a união matemática dos conjuntos e o resultado é (ABCD).

Segundo, o vetor será classificado em ordem crescente de atividades. Embora não seja necessária, a classificação facilita a leitura posterior do conjunto e não interfere criticamente no tempo de execução do artefato.

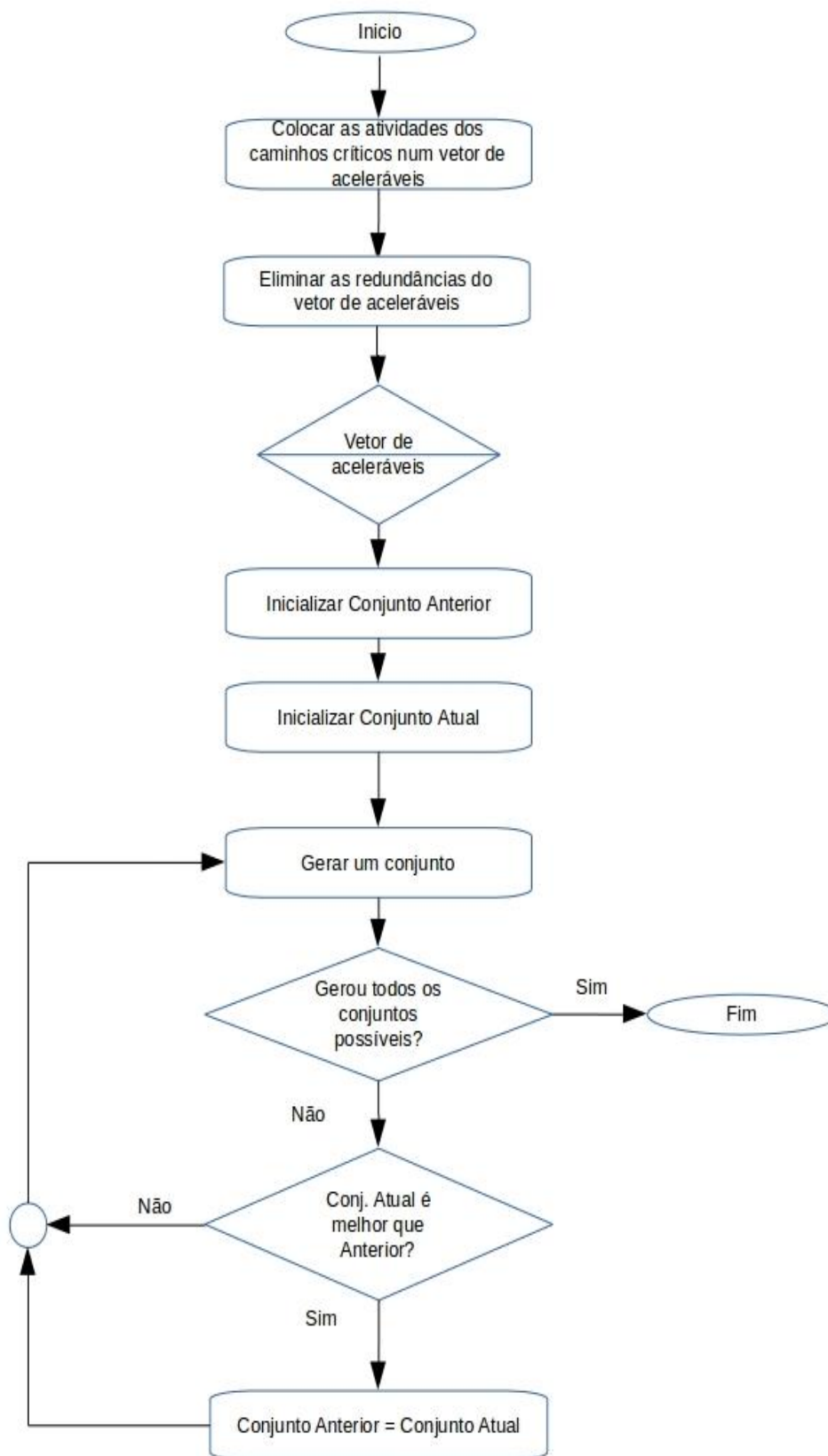
Após estas preparações, o artefato entra em um processo iterativo que gera um conjunto com uma combinação candidata a aceleração e verifica

se o conjunto propicia uma compressão melhor que o conjunto anterior, registrando-o em memória se isso acontecer.

Ao fim deste processo iterativo, estará registrado em memória um conjunto escolhido com a melhor compressão corrente.

A Figura 4.5 mostra o fluxograma do passo 7 em que são feitas as preparações das atividades que são aceleráveis e a escolha de um conjunto para aceleração. Este módulo pode ser verificado nas funções (MontarAtividades) e (ClassificarAtividades) do artefato nos Apêndices A e B.

Figura 4.5 – Fluxograma do Passo 7: Escolher Conjunto de Aceleração



Fonte: Elaborado pelo autor (2017).

4.1.6.1. Gerar um Conjunto para aceleração

O módulo que gera um conjunto é um ponto importante, pois a lógica resultante da abordagem para o processo iterativo de inferir todas as combinações possíveis das atividades dos caminhos críticos pode interferir decisivamente no tempo de execução do artefato.

As combinações são geradas por meio de mapeamento binário. Embora não seja um modelo humanamente intuitivo, a manipulação binária de registros de memória em um computador, quando aplicável, é o recurso mais rápido existente. Porém, há limitações que podem ser vencidas e serão explicadas mais adiante.

Todas as atividades dos caminhos críticos são registradas em memória e sem duplicidade em um vetor de aceleráveis, portanto, se este vetor possui, por exemplo, 'p' atividades passíveis de aceleração, gera-se então todas as combinações possíveis de aceleração, que são:

$C \binom{p}{1}$, $C \binom{p}{2}$, $C \binom{p}{3}$ e assim sucessivamente até $C \binom{p}{p}$.

Esta abordagem usa basicamente uma contagem partindo de 1 até 2^n . Como é preciso usar um *bit* para teste de parada da contagem, é possível ter n *bits* para n atividades e um *bit* para teste de parada, o que dá um campo com $(n + 1)$ *bits*.

Por exemplo, com n igual a 3, pode-se fazer a combinatória de 3 atividades. Este exemplo usa um campo de 4 *bits* preenchidos binariamente como (0111) para o teste de parada feito no *bit* mais significativo. No processo iterativo da contagem, tem-se em numeração binária: (0001), (0010), (0011), (0100), (0101), (0110), (0111) e finalmente (1000).

Cada contagem é verificada com o campo de teste de parada, fazendo-se um (E) lógico *bit a bit*. Esta operação lógica resulta como verdadeira em todas as ocorrências, exceto na contagem (1000).

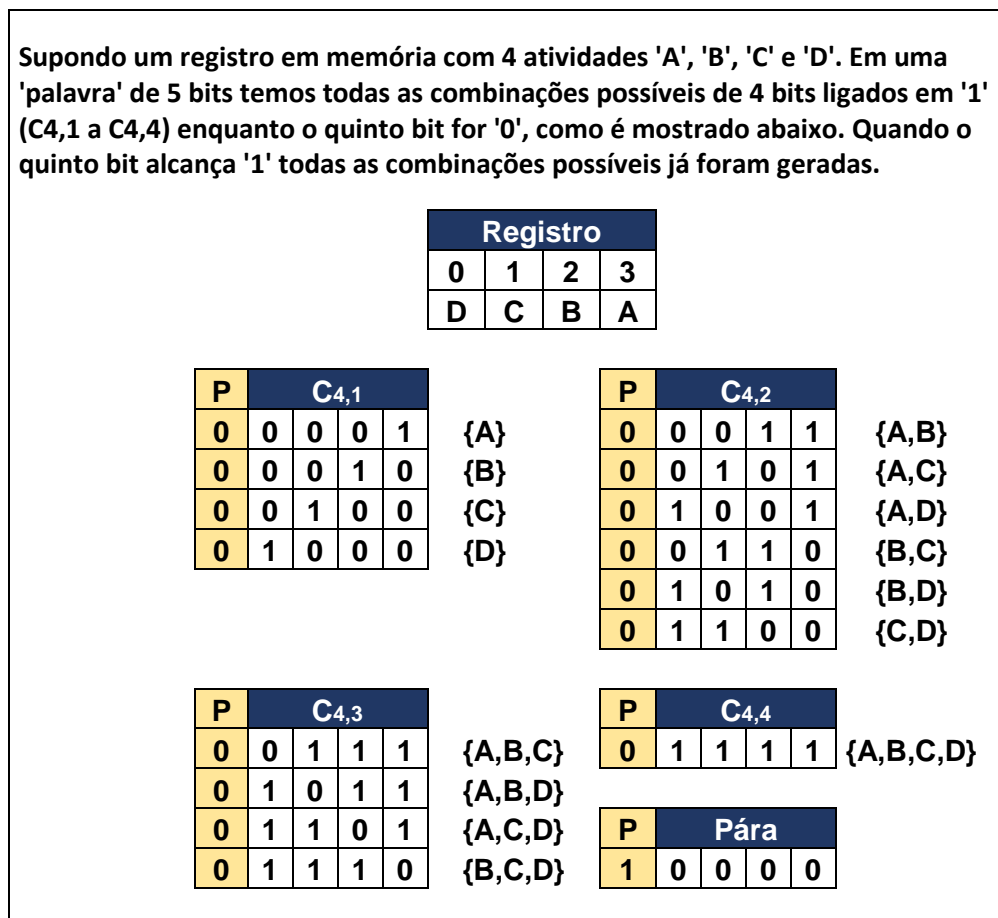
Na contagem (1000) verificada com (0111) *bit a bit*, todos eles são diferentes entre si e o estado resultante da operação lógica é falso, o que satisfaz a condição de parada.

Em todas as outras verificações haverá sempre pelo menos um *bit* ligado em um (1) coincidindo com o *bit* ligado em um (1) do campo de teste de parada. Ou seja, nestas outras verificações, o resultado do (E) lógico *bit a bit*, é o próprio número corrente na contagem.

Em cada ocorrência válida da contagem, os *bits* ligados fornecem os índices das atividades registradas em memória em um campo de mesmo tamanho do campo de contagem.

A Figura 4.6 ilustra um exemplo de como é feito o mapeamento binário em um vetor com quatro atividades.

Figura 4.6 – Combinação com mapeamento binário

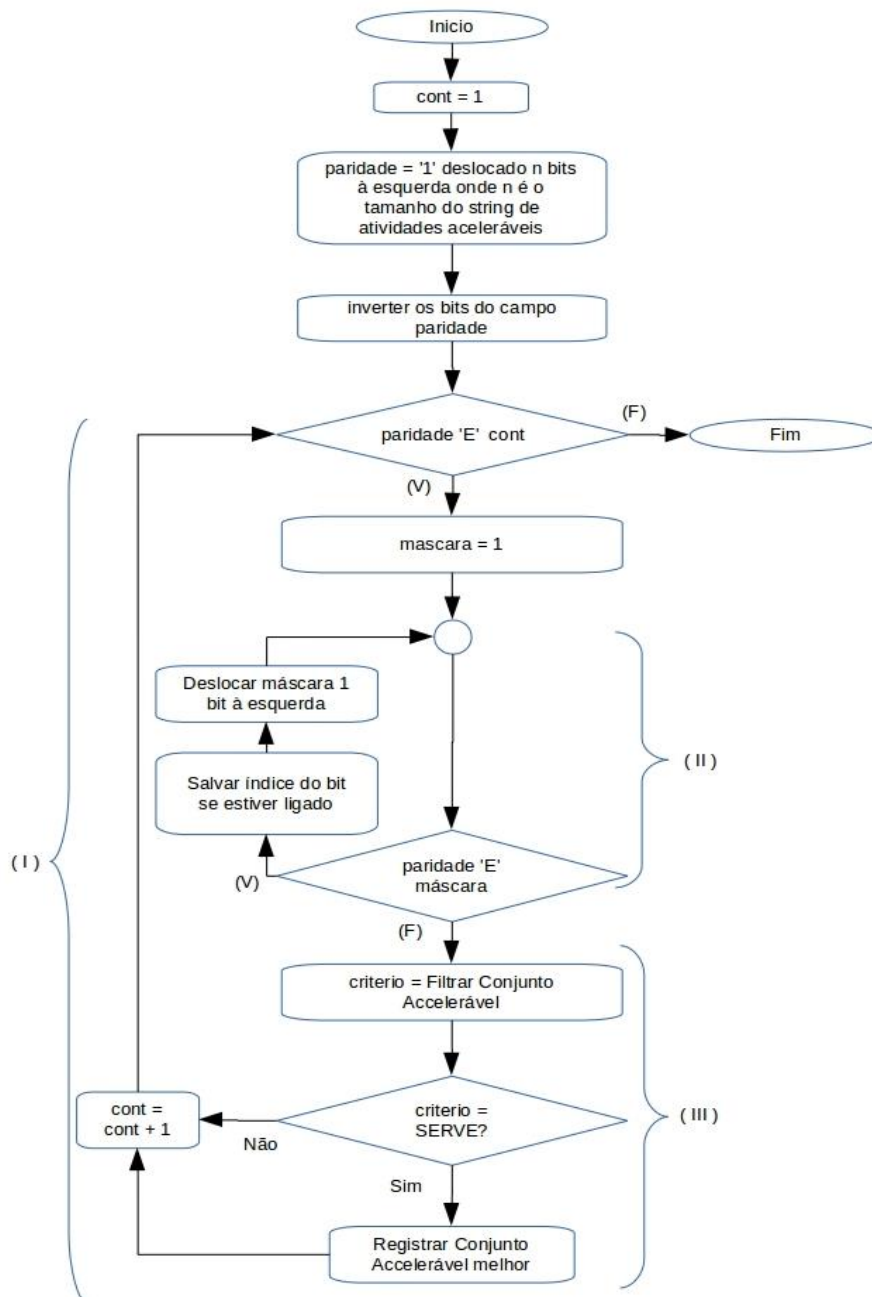


Fonte: Elaborado pelo autor (2017).

Embora a abordagem possua o tamanho da palavra binária como condição limítrofe restritiva dependente do hardware em que o artefato é executado, o limite pode ser vencido usando um vetor de palavras.

A Figura 4.7 mostra o fluxograma para gerar as combinações por mapeamento binário. Esse módulo pode ser verificado na função (GerarConjuntos_Aceleracao) do artefato nos apêndices A e B.

Figura 4.7 – Fluxograma Gerar Combinações por mapeamento binário



Fonte: Elaborado pelo autor (2017).

No bloco iterativo principal (I), a cada iteração o campo (cont) terá uma combinação de *bits* desligados (0) e ligados (1). As posições dos *bits* ligados fornecerão os índices do vetor *string*, que com as atividades aceleráveis, resultam em uma combinação possível. Ao final do *loop*, a memória do computador conterà o registro do conjunto que oferece a melhor compressão.

No processo iterativo do bloco (II), para cada combinação, a máscara começa com um (1) e a cada iteração esse um (1) é deslocado uma posição à esquerda. Quando a posição deste um (1) for a mesma posição do *bit* que está desligado em zero do campo paridade, então será feito o mapeamento da uma combinação corrente.

Quando o fluxo atinge o bloco (III), é porque existe agora um conjunto de aceleração que precisa ser verificado se propicia uma compressão melhor que o conjunto anterior. O fluxo, então, chama um processo de verificação para testar se o conjunto acelerável corrente registrado em memória é melhor que o anterior. Se for, o conjunto anterior é substituído pelo atual e o processo iterativo de contagem continua.

4.1.6.2. Filtrar Conjunto Acelerável

Na varredura por mapeamento binário, cada novo conjunto candidato à compressão é verificado quanto à não redundância e atendimento a todos os caminhos críticos. Se esse novo conjunto produzir uma convergência melhor, torna-se o melhor candidato corrente à compressão em detrimento do conjunto verificado anteriormente. Ao término da varredura binária, o registro em memória conterà o conjunto de aceleração que propiciará o menor incremento no custo de compressão. Portanto, são usados os conjuntos que convergem mais rapidamente para a função objetivo, que no caso sempre será o menor custo final.

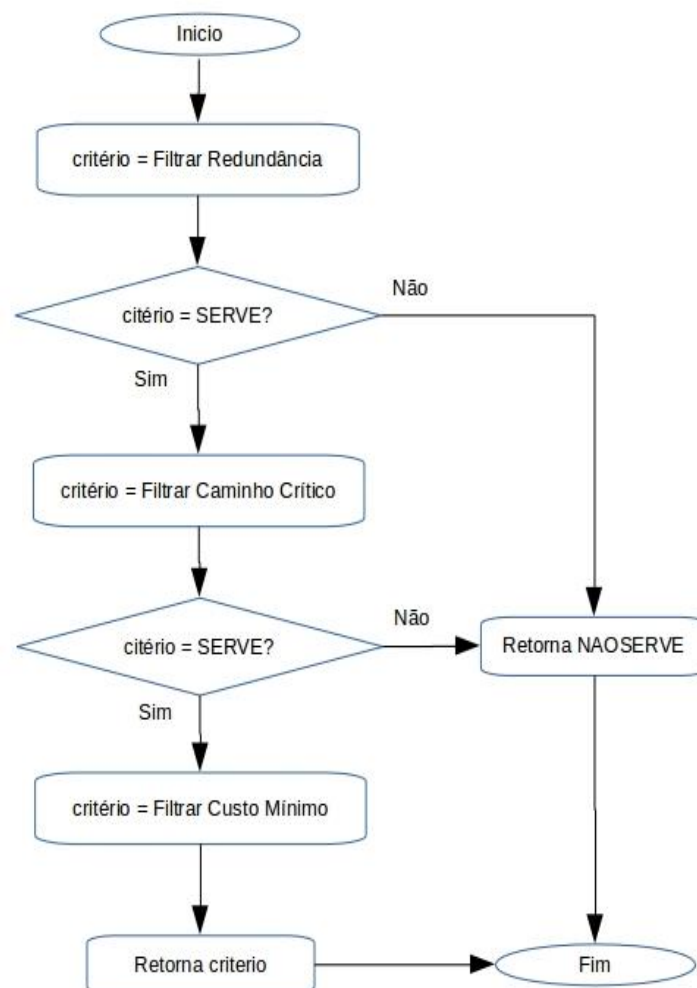
Neste módulo, o artefato deve verificar se o conjunto gerado pelo mapeamento binário propiciará melhor compressão que o conjunto anterior. Ou seja, a cada combinação gerada deve-se ter um conjunto que precisa ser filtrado, pois pode não atender aos critérios de aceleração.

O conjunto passa por três filtros. O primeiro filtro deverá verificar se o conjunto não tem redundância em paralelo, pois cada caminho crítico deverá ser contemplado com apenas uma atividade. Ou seja, se uma atividade atende mais de um caminho, ela deverá aparecer apenas uma vez no conjunto.

Em seguida, é necessário verificar se todos os caminhos críticos foram contemplados, pois todos serão acelerados com a mesma compressão.

O último filtro deve verificar se o conjunto a ser acelerado propicia uma redução na duração do projeto com custo marginal menor que o conjunto anterior registrado na memória. A Figura 4.8 mostra este fluxograma.

Figura 4.8 – Fluxograma do módulo para filtrar aceleração



Fonte: Elaborado pelo autor (2017).

Esse módulo pode ser verificado na função (Filtrar_Aceleracao) do artefato nos apêndices A e B.

Ao passar por cada filtro, uma variável auxiliar interna chamada de *critério* será marcada com um de dois valores simbólicos definidos como (SERVE) ou (NÃO SERVE). O fluxo de verificação para os próximos filtros deverá continuar apenas quando o conjunto receber o aval (SERVE) desta variável auxiliar.

4.1.6.3. Filtrar Redundância

No fluxo, a variável auxiliar *critério* recebe o resultado do primeiro filtro (Filtrar Redundância) que verifica a redundância dos caminhos críticos atendidos. Ou seja, uma determinada atividade não pode aparecer em outro caminho crítico que será atendido por outra atividade do mesmo conjunto. Por exemplo, supondo dois caminhos críticos (ABC) e (CDE), o conjunto (A, C) não seria válido pois a atividade (C) já atende os dois caminhos.

Se cada caminho crítico do conjunto for atendido por uma e, somente uma atividade, o filtro retornará (SERVE) na variável auxiliar. O fluxo em seguida, deverá verificar o estado da variável auxiliar e se nela tiver o aval (SERVE), então o fluxo continuará e chama o próximo filtro: Filtrar Caminho Crítico.

4.1.6.4. Filtrar Caminho Crítico

Nesse ponto do fluxo, a variável auxiliar receberá o resultado do próximo filtro (Filtrar Caminho Crítico), que verificará se todos os caminhos críticos foram atendidos pelo conjunto. Em caso afirmativo, o aval (SERVE) estará na variável auxiliar e o fluxo chamará o próximo filtro: Filtrar Custo Mínimo.

4.1.6.5. Filtrar Custo Mínimo

Se o procedimento chegar até este ponto, ele seguirá para o último e principal módulo do artefato, que é aquele que verifica se o conjunto

gerado propicia uma redução na duração do projeto com custo marginal menor que o conjunto anterior registrado em memória.

Para filtrar o custo mínimo, é necessário encontrar o passo de compressão e o custo marginal. Este é o ponto que diferencia as duas versões do artefato.

Na versão linear basta verificar se o custo marginal do conjunto atual é menor que o anterior registrado em memória, já na versão exponencial convexa, a equação do custo marginal depende do passo de compressão e ele deve ser encontrado primeiro.

O custo direto da atividade após sua aceleração para a versão exponencial convexa é dado por:

$$Cd = Ca - [(Ca - Cn) \cdot (1 - e^{-(\Delta t - p)})] \quad (4.1)$$

As variáveis Ca e Cn são constantes fornecidas na tabela de entrada do artefato antes do processo de *Crashing*. Cn é o custo normal da atividade e Ca é o custo da atividade totalmente acelerada.

A cada iteração, a diferença Δt é a variação entre os instantes inicial (Tn = tempo normal) e final (Ta = tempo acelerado) de aceleração possível ($Tn - Ta$) antes de se aplicar o passo calculado de compressão p , o qual se efetivado, atualizará o instante final (Ta) para futuras iterações.

O resultado Cd é o custo direto da atividade durante o processo de compressão.

Como a função do custo marginal está sendo aproximada por uma curva exponencial convexa decrescente, o que se tem é um deslocamento temporal no eixo das abscissas ($\Delta t - p$) e um deslocamento de custo no eixo das ordenadas ($Ca - Cn$) (LATHI, 2007).

Devido às características da função por aproximação exponencial convexa descrita na equação 4.1, o módulo da primeira derivada na equação 4.2 fornece o aumento do custo direto da atividade ocasionado

pela aceleração desta em p unidades de tempo no ponto específico ($\Delta t - p$).

$$Cm = |-(Ca - Cn). e^{-(\Delta t - p)}| \quad (4.2)$$

O aumento do custo direto da atividade devido a aceleração de p unidades de tempo por aproximação linear é dado por:

$$Cm = \left(\frac{Ca - Cn}{Dn - Da}\right). p \quad (4.3)$$

A aceleração p é a diferença entre o tempo normal da atividade e sua duração após a última compressão.

O custo indireto é fornecido pela equação (4.4) em que D é a somatória da duração das atividades do caminho crítico e Cf é o Custo Fixo;

$$Ci = Cf . D \quad (4.4)$$

O custo direto do projeto é a somatória dos custos normais (Cn) com o aumento no custo direto devido às compressões e custos marginais (Cm):

$$Cdp = \sum Cn + \sum Cm \quad (4.5)$$

O custo total é a soma dos custos diretos e indiretos é dado por:

$$Ct = Ci + Cdp \quad (4.6)$$

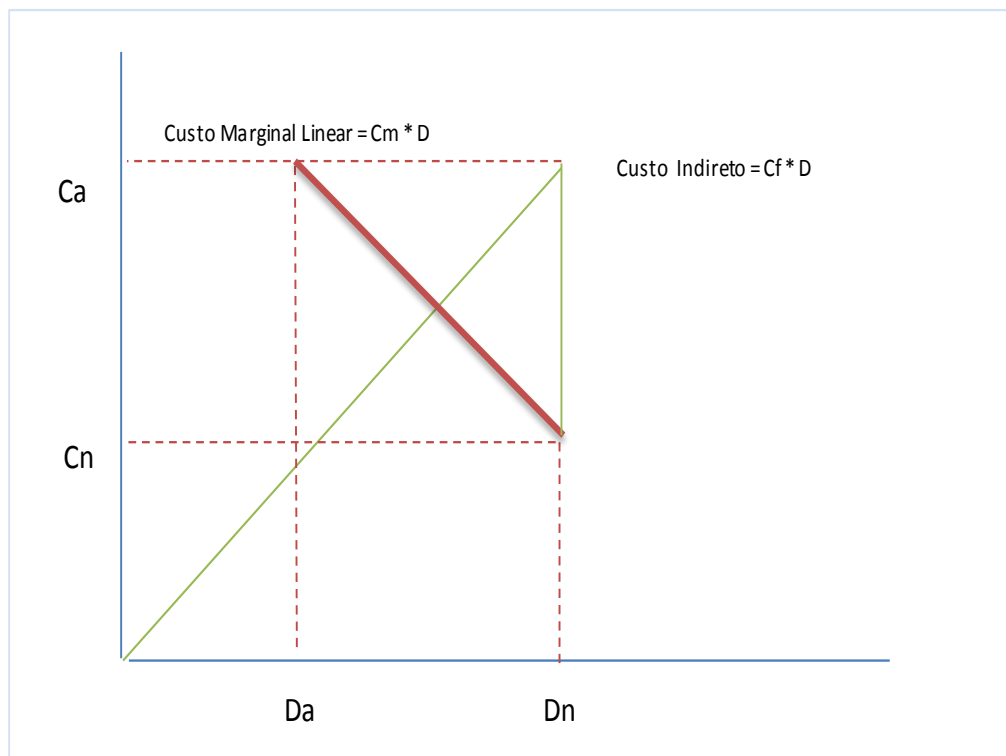
Na versão linear, é escolhido o conjunto, cujas atividades contidas nele, possuem o menor somatório de custos marginais. Havendo empate, considera-se o conjunto que propicia a maior compressão, e se ainda houver empate, é escolhido o menor conjunto em número de atividades.

A evolução do custo marginal linear (equação 4.3), é uma reta de inclinação oposta à inclinação da reta de evolução do custo indireto do projeto (equação 4.4). Então, para a mesma compressão em unidades de tempo, o custo final do projeto pode assumir três situações diferentes respectivamente quando o custo marginal é menor, igual ou maior que o custo fixo em valores absolutos: decresce; fica inalterado; aumenta.

Portanto, na versão linear, basta escolher o conjunto de aceleração que possui o menor custo marginal constante calculado na entrada dos dados, e isto implicará na melhor aproximação da função objetivo determinada pelo critério de parada escolhido.

A Figura 4.9 mostra um exemplo ilustrativo do comportamento linear da evolução do custo indireto e do custo marginal para uma atividade de um determinado projeto. No gráfico, o custo normal é C_n , o custo acelerado é C_a , a duração normal é D_n , a duração acelerada é D_a , e D é a duração do caminho crítico.

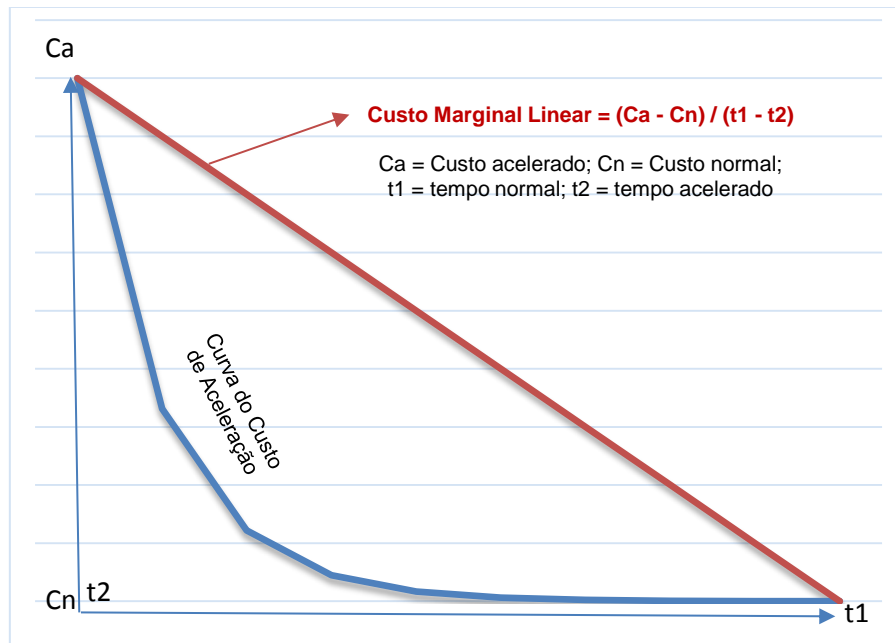
Figura 4.9 – Evolução dos custos, indireto e marginal para uma dada atividade de um projeto



Fonte: Elaborado pelo autor (2017).

A Figura 4.10 mostra a evolução do custo marginal linear e uma curva de aproximação da evolução do custo de aceleração para uma dada atividade de um projeto.

Figura 4.10 – Gráfico de evolução do custo marginal linear e convexo



Fonte: Elaborado pelo autor (2017).

4.1.6.6. Encontrar o Passo de Compressão

Na versão exponencial convexa, é necessário descobrir qual conjunto de aceleração propicia o menor custo marginal. Nesse contexto, o custo marginal de cada atividade precisa ser calculado dinamicamente a cada iteração do *Crashing*. E como o custo está associado a uma função de aproximação que depende da passagem do tempo de duração (equação 4.1), é necessário encontrar primeiro a melhor compressão. Essa compressão é usada na primeira derivada da função do custo para encontrar o custo marginal (equação 4.2).

Sendo T_i , o tempo normal de duração da atividade i e T_{ai} , o tempo acelerado de duração da mesma atividade i , a cada iteração k , para encontrar o passo de compressão p do projeto em unidades de tempo, busca-se o mínimo entre o menor dos valores $(T_i - T_{ai})$ para todas as atividades i do conjunto de atividades a serem aceleradas e a menor folga entre os caminhos que não contenham quaisquer subconjuntos de atividades do conjunto de aceleráveis (CONTADOR, 1998). Nas duas versões, o procedimento para esta busca é igual e pode ser verificado na função (EncontrarCompressão) do artefato nos Apêndices A e B.

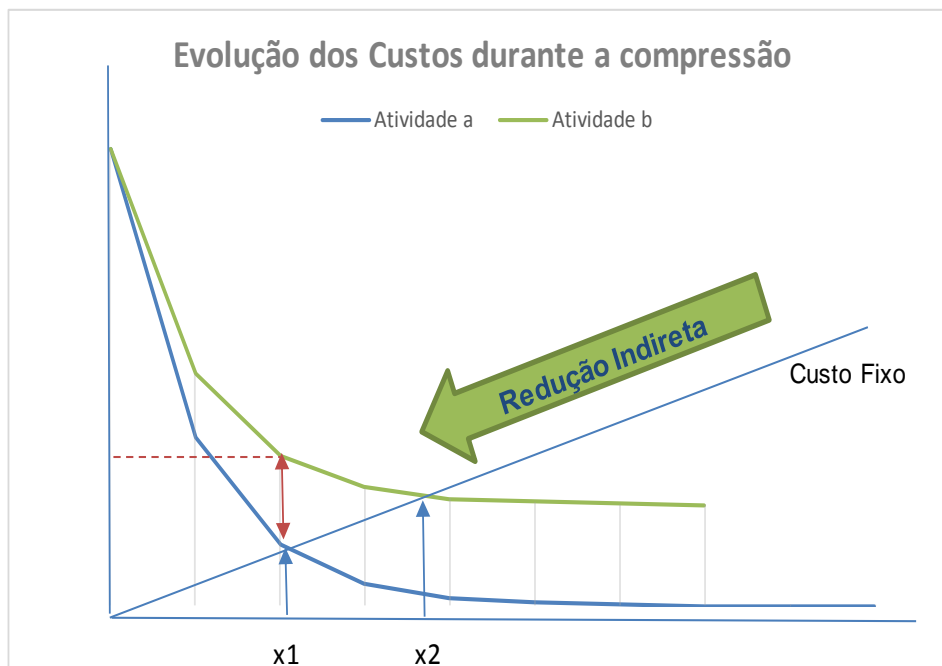
4.1.6.7. Encontrar o Custo Marginal na Curva de aproximação

Na versão exponencial convexa, após ser encontrada a melhor compressão, se tem lugar um processo iterativo em que são calculados os custos marginais do conjunto acelerável.

O custo marginal é determinado pela derivada em um dado ponto (dado pelo passo) da curva de aproximação. Como o conjunto de atividades aceleráveis pode conter várias atividades e, portanto, várias curvas, precisa-se encontrar qual a melhor compressão dentro do intervalo que vai de 1 até o passo de compressão encontrado p . Portanto, no contexto não linear, nem sempre o passo encontrado é o melhor ponto de compressão.

Na Figura 4.11 é mostrado o exemplo de um conjunto com duas atividades a acelerar. Os pontos resultantes da compressão ($\Delta t - p$) para as durações x_1 e x_2 mostram que a compressão para x_2 é melhor que em x_1 . No ponto x_1 , o crescimento exponencial da atividade (b) dado pela curva inviabiliza a compressão em comparação com o ponto x_2 .

Figura 4.11 – Gráfico de evolução dos custos durante a compressão



Fonte: Elaborado pelo autor (2017).

Pela equação 4.4, o custo indireto do projeto é o produto entre o custo fixo e a duração do projeto, que é a somatória das durações das atividades do caminho crítico.

Como a duração do caminho crítico é um intervalo de tempo e o passo de compressão encontrado (p) também o é, neste ponto eles são grandezas escalares equivalentes para o cálculo de custo indireto que decresce enquanto o custo direto aumenta, e o que se quer calcular a cada iteração é a redução do custo indireto do projeto, apontada no gráfico da Figura 4.11 como redução indireta.

Se o custo indireto do projeto, dado pela equação 4.4 é o produto do custo fixo pela duração do caminho crítico ($Cf.D$) então a redução do custo indireto pode ser dada pelo produto do custo fixo pela passagem do tempo calculada no passo de compressão p e subtraindo desse total o custo marginal encontrado para o mesmo passo de compressão.

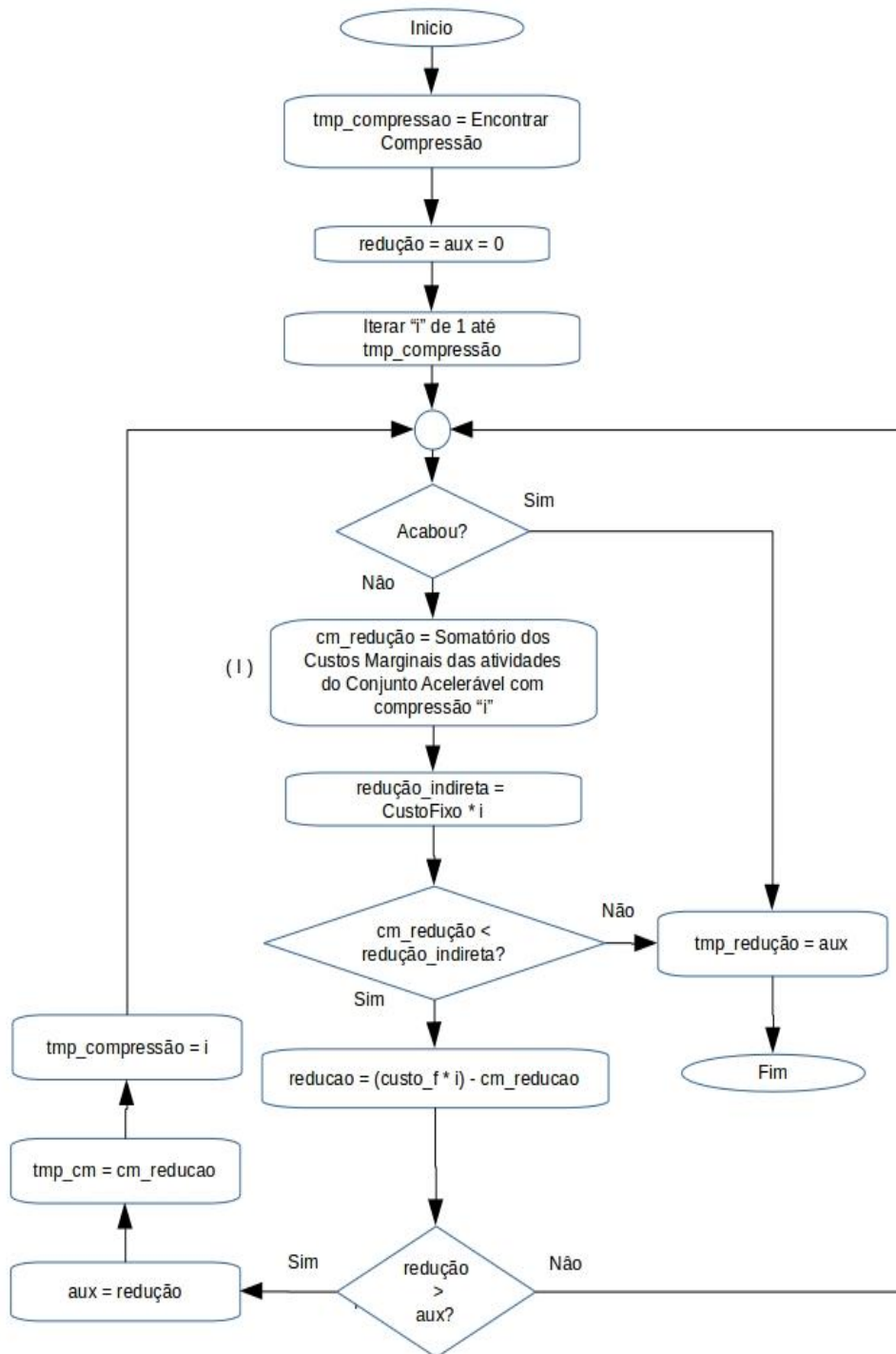
É preciso lembrar que os custos marginais das atividades aceleradas com o passo p serão adicionados aos custos diretos atuais destas atividades, e portanto eles devem ser subtraídos da redução indireta que é dada pela equação 4.7 a seguir, aplicada apenas na versão exponencial convexa do artefato desenvolvido.

$$Ri = (Cf.p) - Cm \quad (4.7)$$

Na Figura 4.12 o bloco (I) do fluxograma da versão exponencial convexa para o filtro do custo mínimo, mostra o ponto onde o custo marginal do conjunto de aceleração é calculado dinamicamente.

Esse módulo pode ser verificado na função (EncontrarMaiorReducao) do artefato no Apêndice B e, para a versão linear pode ser verificado na função (Filtro_CustoMinimo) do artefato no apêndice A.

Figura 4.12 – Fluxograma Filtro Custo Mínimo



Fonte: Elaborado pelo autor (2017).

Os dois primeiros filtros são de programação mais simples e por esta razão os fluxogramas não foram mostrados. O último filtro é o mais intrínseco e importante para o TCTP e é nele que acontece a mudança mais significativa entre as versões do artefato.

4.1.7. Passo 8 - Processar Compressão e Calcular Custos

Neste ponto haverá um registro em memória com um conjunto de atividades que propiciam a melhor redução de custo indireto, que resultam na compressão de duração do tempo de conclusão do projeto a um custo total mínimo.

Este módulo pode ser verificado nas funções (Registrar_Aceleracao) e (Processar_Aceleracao) do artefato no apêndice B.

4.1.8. Passo 9 - Verificar critério de parada

O artefato possui dois critérios de parada, seja pelo custo mínimo ou pelo prazo. Pelo custo mínimo, o custo marginal calculado a cada iteração é comparado ao custo fixo do projeto. Se o custo marginal for maior ou igual ao custo mínimo, a compressão deixa de ser vantajosa. Pelo critério do prazo, as compressões são feitas para se chegar o mais próximo possível do prazo desejado ou alcançá-lo exatamente. As iterações terminam quando é satisfeito o critério escolhido.

Este módulo pode ser verificado na função (VerificarCritério_Parada) do artefato nos apêndices A e B.

De acordo com a heurística adotada, no custo mínimo, o artefato comprime a duração do projeto verificando os custos de aceleração e chega no menor custo possível, a partir do qual ele começa a aumentar.

No prazo final, o artefato comprime a duração do projeto até o mais próximo possível do prazo estipulado, levando em consideração sempre os menores custos de aceleração.

A seção seguinte detalha dois estudos de caso para as duas versões do artefato.

5 ESTUDOS DE CASO

5.1. Caso 1 – Versão Linear

O primeiro estudo de caso é um exercício acadêmico igualmente proposto em Carvalho (2011) e Contador (1998).

A entrada para o algoritmo é mostrada na Tabela 5.1.

Tabela 5.1: Dados de entrada do Algoritmo – Caso 1

9 (número de registros da tabela)					
Ativ	Precedente	Tempo de duração		Custos	
		Acelerado	Normal	Acelerado	Normal
A	Nula	6	11	40	30
B	Nula	5	12	63	42
C	Nula	11	19	31	23
D	A	7	12	71	51
E	BC	5	11	159	105
F	C	9	13	210	170
G	DE	13	22	135	90
H	BC	7	12	135	100
I	F	9	14	135	130

Fonte: Elaborado pelo autor (2017).

O artefato lê esta tabela de entrada e registra os dados em memória. A primeira linha da tabela aponta o número de registros úteis contidos na própria tabela. Nesse caso, o número de registros é 9. Nas linhas seguintes, as colunas são os campos: atividade, atividades precedentes (antecessoras imediatas), duração acelerada, duração normal, custo acelerado e custo normal.

Neste estudo de caso, o projeto possui um custo fixo por unidade de tempo igual a 8 unidades monetárias. Esta informação é passada como parâmetro de entrada ao se executar junto com o critério de parada que será pelo custo mínimo. Os resultados do artefato para esse caso são mostrados nas tabelas a seguir. O cenário inicial, antes das compressões, é mostrado na Tabela 5.2.

Tabela 5.2 – Relatório Cenário Inicial (Caso 1, Versão Linear)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	19	11	31	23	1.00	0	0
(D)	12	12	7	71	51	4.00	11	18
(E)	11	11	5	159	105	9.00	19	19
(F)	13	13	9	210	170	10.00	19	19
(G)	22	22	13	135	90	5.00	30	30
(H)	12	12	7	135	100	7.00	19	19
(I)	14	14	9	135	130	1.00	32	38
Custos: Indiretos = 416 Diretos = 741 Projeto = 1157								
Caminhos		Durações		Folgas				
ADG		45		7				
BEG		45		7				
BH		24		28				
CEG		52		0		→ Crítico		
CFI		46		6				
CH		31		21				

Fonte: Elaborado pelo autor (2017).

Abaixo da linha que mostra os custos, as legendas são autoexplicativas, mas as 9 colunas acima significam respectivamente: Atividade, duração normal, duração atual, duração acelerada, custo atual, custo marginal, data cedo e data tarde.

No final do *Crashing*, a coluna com a duração atual (D_Atu) (entre a normal e a acelerada) conterà o solução do problema em relação à estimativa tempo de duração das atividades do projeto. A coluna pode servir de comparação mostrando as diferentes soluções entre as versões do artefato.

Nessa primeira passagem, sem nenhuma aceleração, a duração do caminho crítico calculada a partir da somatória das durações das atividades que compõem o caminho CEG foi de 52 unidades de tempo.

De acordo com as equações descritas na seção 4.1.6.5 e diante dos valores informados e calculados, o artefato já consegue calcular o custo total do projeto sem as compressões (acelerações), conforme é mostrado na tabela 5.3.

Tabela 5.3 – Resultados calculados antes da compressão para o caso 1
(versão Linear)

Variável	Valores obtidos	Metodologia de cálculo
Cf	$Cf = 8$	Informado como parâmetro na chamada do artefato
D	$D = 52$	Calculado pela função (Duracoes E Folgas) do
Ci	$Ci = Cf \cdot D = 8 \cdot (52) = 416$	Dado por (4.4)
Cdp	$Cdp = 741$	Dado por (4.5)
Ct	$Ct = 416 + 741 = 1157$	Dado por (4.6)

Fonte: Elaborado pelo autor (2017).

A partir deste ponto, o artefato começa a verificar se há possibilidades de compressão e as executa fazendo iterações até atingir o critério de parada escolhido. Como por enquanto existe apenas um caminho crítico, é fácil visualizar os conjuntos possíveis de aceleração.

Cada atividade do caminho pode ser acelerada se houver delta de aceleração. O artefato decide pela atividade (C) porque ela possui o menor custo marginal igual a um (1), que por sua vez, é menor que o custo fixo 8. A primeira iteração com essa compressão é mostrada na tabela 5.4.

Tabela 5.4 – Cenário 1ª iteração (Caso 1, Versão Linear)

Acelerando Conjunto { C } em 7 unidades de tempo - (Iteração: 1)								
Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	12	11	31	30	1.00	0	0
(D)	12	12	7	71	51	4.00	11	11
(E)	11	11	5	159	105	9.00	12	12
(F)	13	13	9	210	170	10.00	12	12
(G)	22	22	13	135	90	5.00	23	23
(H)	12	12	7	135	100	7.00	12	12
(I)	14	14	9	135	130	1.00	25	31
Custos: Indiretos = 360 Diretos = 748 Projeto = 1108								
Caminhos		Durações		Folgas				
ADG		45		0		→ Crítico		
BEG		45		0		→ Crítico		
BH		24		21				
CEG		45		0		→ Crítico		
CFI		39		6				
CH		24		21				

Fonte: Elaborado pelo autor (2017).

O artefato decidiu acelerar (C) por este possuir o menor custo marginal entre os conjuntos (C) = 1, (E) = 9 e (G) = 5.

O passo da aceleração é o valor mínimo entre o menor delta de aceleração entre as atividades do conjunto e a menor folga entre os caminhos que não possuem a atividade acelerada, como já descrito anteriormente. Como o conjunto possui apenas a atividade (C), esta atividade já possui o menor delta. Como a atividade (C) tem duração normal de 19 unidades de tempo e duração acelerada de 11 unidades de tempo, o delta possível de aceleração é 8.

Os caminhos que não possuem a atividade acelerada são (ADG) com folga 7, (BEG) com folga 7 e (BH) com folga 28. O mínimo entre (7, 7, 28) é 7 e o mínimo entre (8, 7) é 7. Portanto, como visto na primeira iteração, o passo da aceleração calculado foi 7.

Como o custo marginal de (C) em unidades monetárias é igual a um (1) e o projeto está sendo acelerado em 7 unidades de tempo, os custos diretos do projeto aumentarão em 7 unidades monetárias. Mas,

paralelamente, os custos indiretos do projeto estão sendo diminuídos à razão de 8 unidades monetárias ($CF = 8$) à cada unidade de tempo acelerado, dando um decréscimo de 56 unidades monetárias (produto do passo da aceleração pelo custo fixo). Portanto, o custo do projeto caiu em $(56 - 7)$, ou 49 unidades monetárias, totalizando 1.108, como confere o resultado na Tabela 5.5 usando as equações descritas anteriormente.

Tabela 5.5 - Resultados calculados no 1º cenário do caso 1 (versão Linear)

Variável	Valores obtidos	Metodologia de cálculo
Cf	$Cf = 8$	Informado como parâmetro na chamada do artefato
D	$D = 45$	Calculado pela função (Duracoes_E_Folgas) do artefato
Ci	$Ci = Cf \cdot D = 8 \cdot (45) = 360$	Dado por (4.4)
Cdp	$Cdp = 748$	Dado por (4.5)
Ct	$Ct = 360 + 748 = 1108$	Dado por (4.6)

Fonte: Elaborado pelo autor (2017).

Pelo critério de parada escolhido, o ponto de custo mínimo do projeto será alcançado somente quando o custo marginal do conjunto encontrado for igual ou maior que o custo fixo. Enquanto esse custo for menor e as atividades dos caminhos críticos possuírem delta de aceleração, o projeto pode ser comprimido.

Desse ponto em diante, a inferência dos conjuntos passíveis de aceleração já se torna mais trabalhosa, justamente pelo fato de haver agora três caminhos críticos. A segunda, terceira e quarta iterações são mostradas nas Tabelas 5.6, 5.7 e 5.8, respectivamente.

Tabela 5.6 – Cenário 2ª iteração (Caso 1, Versão Linear)

Acelerando Conjunto { G } em 6 unidades de tempo – (Iteração: 2)								
Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	12	11	31	30	1.00	0	0
(D)	12	12	7	71	51	4.00	11	11
(E)	11	11	5	159	105	9.00	12	12
(F)	13	13	9	210	170	10.00	12	12
(G)	22	16	13	135	120	5.00	23	23
(H)	12	12	7	135	100	7.00	12	12
(I)	14	14	9	135	130	1.00	25	25
Custos: Indiretos = 312 Diretos = 778 Projeto = 1090								
Caminhos		Durações		Folgas				
ADG		39		0		→ Crítico		
BEG		39		0		→ Crítico		
BH		24		15				
CEG		39		0		→ Crítico		
CFI		39		0		→ Crítico		
CH		24		15				

Fonte: Elaborado pelo autor (2017).

Tabela 5.7 – Cenário 3ª iteração (Caso 1, Versão Linear)

Acelerando Conjunto { G I } em 3 unidades de tempo – (Iteração: 3)								
Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	12	11	31	30	1.00	0	0
(D)	12	12	7	71	51	4.00	11	11
(E)	11	11	5	159	105	9.00	12	12
(F)	13	13	9	210	170	10.00	12	12
(G)	22	13#	13	135	135	5.00	23	23
(H)	12	12	7	135	100	7.00	12	12
(I)	14	11	9	135	133	1.00	25	25
Custos: Indiretos = 288 Diretos = 796 Projeto = 1084								
Caminhos		Durações		Folgas				
ADG		36		0		→ Crítico		
BEG		36		0		→ Crítico		
BH		24		12				
CEG		36		0		→ Crítico		
CFI		36		0		→ Crítico		
CH		24		12				

Fonte: Elaborado pelo autor (2017).

Tabela 5.8 – Cenário 4ª iteração (Caso 1 - Versão Linear)

Acelerando Conjunto { A B C } em 1 unidade de tempo – (Iteração: 4)								
Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	10	6	40	32	2.00	0	0
(B)	12	11	5	63	45	3.00	0	0
(C)	19	11#	11	31	31	1.00	0	0
(D)	12	12	7	71	51	4.00	10	10
(E)	11	11	5	159	105	9.00	11	11
(F)	13	13	9	210	170	10.00	11	11
(G)	22	13#	13	135	135	5.00	22	22
(H)	12	12	7	135	100	7.00	11	11
(I)	14	11	9	135	133	1.00	24	24
Custos: Indiretos = 280 Diretos = 802 Projeto = 1082								
Caminhos		Durações		Folgas				
ADG		35		0		→ Crítico		
BEG		35		0		→ Crítico		
BH		24		11				
CEG		35		0		→ Crítico		
CFI		35		0		→ Crítico		
CH		24		11				

Fonte: Elaborado pelo autor (2017).

O artefato alcança o critério de parada neste momento. Sabe-se que as acelerações devem ocorrer em paralelo, atendendo todos os caminhos críticos, mas, nota-se que um dos caminhos críticos não pode mais ser acelerado.

No caminho (CEG), as atividades (C) e (G) não possuem mais delta de aceleração, pois estão totalmente aceleradas. Por sua vez, a atividade (E) não é passível de aceleração por este critério de parada, justamente porque possui custo marginal 9, que é maior que o custo fixo 8 do projeto. Portanto, o critério de parada foi alcançado com o custo mínimo do projeto em 1082 unidades monetárias, conforme foi demonstrado.

O relatório completo do estudo de caso 1 na versão linear pode ser visto no Apêndice C.

5.2. Caso 2 – Versão Linear

O segundo estudo de caso é um exemplo hipotético em que as estimativas de tempo de duração e custos das atividades foram escolhidas de forma pseudoaleatória. Foram escolhidas estimativas aleatórias para durações e custos normais e, posteriormente, foram escolhidas as respectivas durações e custos acelerados, de maneira que a razão resultasse em número inteiro na maioria dos casos para facilitar a leitura do exemplo na versão linear. A entrada para o algoritmo é mostrada na Tabela 5.9 a seguir.

Tabela 5.9: Dados de entrada do Algoritmo – Caso 2

23 (número de registros)					
Ativ	Precedente	Tempo de duração		Custos	
		Acelerado	Normal	Acelerado	Normal
A	nula	3	10	300	160
B	A	1	4	146	125
C	A	2	10	250	170
D	A	2	5	72	60
E	A	3	12	198	135
F	E	6	21	430	215
G	CDE	5	8	96	90
H	BG	5	11	132	120
I	H	1	5	290	190
J	G	3	7	84	80
K	F	3	8	280	180
L	K	1	11	420	320
M	IJL	4	24	1240	620
N	M	8	23	380	260
O	I	3	13	570	410
P	N	9	14	260	180
Q	N	7	17	1020	900
R	P	4	13	380	200
S	Q	6	9	270	210
T	R	10	19	522	432
U	ST	1	5	200	160
V	OU	2	12	350	250
X	V	3	11	424	336

Fonte: Elaborado pelo autor (2017).

Como neste caso o número de atividades é relativamente alto, para facilitar a leitura, foram colocados aqui apenas os resultados do *Crashing*. O artefato foi executado com critério de parada por prazo final e alcançou a meta em 19 iterações, comprimindo o projeto de 183 para 58 unidades de tempo.

A Tabela 5.10 mostra o relatório do primeiro cenário, sem compressão.

5.2.1. Relatório do primeiro cenário sem compressão – Caso 2

Tabela 5.10 – Primeiro cenário sem compressão (Caso 2, Versão Linear)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	12	3	198	135	7.00	10	10
(F)	21	21	6	430	215	14.33	22	22
(G)	8	8	5	96	90	2.00	22	22
(H)	11	11	5	132	120	2.00	30	46
(I)	5	5	1	290	190	25.00	41	57
(J)	7	7	3	84	80	1.00	30	46
(K)	8	8	3	280	180	20.00	43	43
(L)	11	11	1	420	320	10.00	51	51
(M)	24	24	4	1240	620	31.00	62	62
(N)	23	23	8	380	260	8.00	86	86
(O)	13	13	3	570	410	16.00	46	62
(P)	14	14	9	260	180	16.00	109	109
(Q)	17	17	7	1020	900	12.00	109	109
(R)	13	13	4	380	200	20.00	123	123
(S)	9	9	6	270	210	20.00	126	146
(T)	19	19	10	522	432	10.00	136	136
(U)	5	5	1	200	160	10.00	155	155
(V)	12	12	2	350	250	10.00	160	160
(X)	11	11	3	424	336	11.00	172	172
Custos > Indiretos: 3660 Diretos: 5803 Projeto = 9463 Prazo = 183								
Caminhos			Durações	Folgas				
ABHIMNPRTUVX			151	32				
ABHIMNQSUVX			131	52				
ABHIOVX			66	117				
ACGHIMNPRTUVX			165	18				
ACGHIMNQSUVX			145	38				
ACGHIOVX			80	103				
ACGJMNPRTUVX			156	27				
ACGJMNQSUVX			136	47				

ADGHIMNPRTUVX	160	23	
ADGHIMNQSUVX	140	43	
ADGHIOVX	75	108	
ADGJMNPRUVX	151	32	
ADGJMNSUVX	131	52	
AEFKLMNPRTUVX	183	0	→ Crítico
AEFKLMNQSUVX	163	20	
AEGHIMNPRTUVX	167	16	
AEGHIMNQSUVX	147	36	
AEGHIOVX	82	101	
AEGJMNPRUVX	158	25	
AEGJMNSUVX	138	45	

Fonte: Elaborado pelo autor (2017).

Este exemplo de rede de projeto possui vinte caminhos possíveis e no começo há apenas um 1 caminho crítico. Na décima segunda iteração, o número de caminhos críticos vai para 4, na décima quarta iteração sobe para 6 e na décima quinta iteração, já está em oito (8) caminhos críticos. Na décima oitava iteração, o número de caminhos críticos alcança o seu máximo, que é 14 e se mantém na última iteração. O relatório completo desse estudo de caso pode ser visto no Apêndice D.

A Tabela 5.11 mostra o último cenário após o *Crashing*.

Tabela 5.11 – Último cenário após o *Crashing* (Caso 2, Versão Linear)

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	54	4	
ABHIMNQSUVX	54	4	
ABHIOVX	31	27	
ACGHIMNPRTUVX	58	0	→ Crítico
ACGHIMNQSUVX	58	0	→ Crítico
ACGHIOVX	35	23	
ACGJMNPRUVX	58	0	→ Crítico
ACGJMNSUVX	58	0	→ Crítico
ADGHIMNPRTUVX	58	0	→ Crítico
ADGHIMNQSUVX	58	0	→ Crítico
ADGHIOVX	35	23	
ADGJMNPRUVX	58	0	→ Crítico
ADGJMNSUVX	58	0	→ Crítico
AEFKLMNPRTUVX	58	0	→ Crítico
AEFKLMNQSUVX	58	0	→ Crítico
AEGHIMNPRTUVX	58	0	→ Crítico
AEGHIMNQSUVX	58	0	→ Crítico
AEGHIOVX	35	23	
AEGJMNPRUVX	58	0	→ Crítico
AEGJMNSUVX	58	0	→ Crítico

Fonte: Elaborado pelo autor (2017).

A Tabela 5.12 mostra a evolução do *Crashing* com os dados mais relevantes extraídos do Apêndice D.

Tabela 5.12 – Evolução do *Crashing* (Caso 2, Versão Linear)

Contexto de Aceleração - Caso 2 - Versão Linear					
Iteração	Conjunto Acelerado	Passo	Número de Caminhos Críticos	Duração do Projeto	Custo Total do Projeto
0		0	1	183	9.463,00
1	E	9	1	174	9.346,00
2	N	15	1	159	9.166,00
3	V	10	1	149	9.066,00
4	T	9	1	140	8.976,00
5	L	9	2	131	8.886,00
6	U	4	2	127	8.846,00
7	X	8	2	119	8.774,00
8	HL	1	2	118	8.766,00
9	P	5	2	113	8.746,00
10	F H	5	2	108	8.728,00
11	F G	3	2	105	8.717,00
12	R	6	4	99	8.717,00
13	A	7	4	92	8.717,00
14	C F	5	6	87	8.739,00
15	C D F	2	8	85	8.755,00
16	M	20	8	65	8.975,00
17	Q R	3	8	62	9.011,00
18	I K	3	14	59	9.086,00
19	I J K	1	14	58	9.112,00

Fonte: Elaborado pelo autor (2017).

5.3. Caso 1 – Versão Exponencial Convexa

Neste tópico é apresentado o mesmo estudo de caso da sessão 5.2., porém usando a versão exponencial convexa do artefato. A tabela de entrada e os parâmetros passados são os mesmos, bem como o artefato trata a entrada do mesmo modo. O cenário inicial, antes das compressões, é mostrado na Tabela 5.13.

Tabela 5.13 – Relatório Cenário Inicial – (Caso 1, Versão Exponencial Convexa)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30.00	2.00	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	19	11	31	23.00	1.00	0	0
(D)	12	12	7	71	51.00	4.00	11	18
(E)	11	11	5	159	105.00	9.00	19	19
(F)	13	13	9	210	170.00	10.00	19	19
(G)	22	22	13	135	90.00	5.00	30	30
(H)	12	12	7	135	100.00	7.00	19	19
(I)	14	14	9	135	130.00	1.00	32	38
#Atividade totalmente acelerada								
Custos: Diretos = 741 Acelerado = 741.00 Projeto = 1157.00								
Caminhos		Durações		Folgas				
ADG		45		7				
BEG		45		7				
BH		24		28				
CEG		52		0		→ Crítico		
CFI		46		6				
CH		31		21				

Fonte: Elaborado pelo autor (2017).

Nesta versão exponencial Convexa, os custos das atividades do projeto estão associados à função exponencial convexa de aproximação, descrita na equação (4.1).

Como a evolução dos custos durante a aceleração é uma curva, a função exponencial convexa pode refletir valores mais próximos da realidade, mas isso não significa que a abordagem exponencial convexa seja melhor. Nos testes realizados, a versão por aproximação linear foi mais rápida em alguns casos podendo talvez ser melhor indicado utilizar a versão linear, já em outros casos, a variação de custo foi maior, em que a recomendação poderia ser pela versão exponencial convexa.

Inicialmente os custos marginais são colocados primeiro com os valores dos coeficientes angulares iguais aos da versão linear. Depois, durante as acelerações, eles são calculados como a primeira derivada da função exponencial convexa nas equações (4.1) e (4.2), descritas no capítulo anterior.

Como foi visto anteriormente, as únicas diferenças entre as versões linear e exponencial convexa do artefato são os cálculos dos custos diretos e a escolha da melhor compressão. Nesse estudo de caso, os passos de compressão e os conjuntos de aceleração coincidiram algumas vezes, apenas por acaso.

O primeiro relatório, antes das iterações de aceleração, é igual ao anterior com as mesmas opções de escolha de aceleração, que são os conjuntos (C), (E) e (G). A seguir, serão detalhados os contextos das três possíveis escolhas usando as equações (4.2) e (4.7), conforme é mostrado na Tabela 5.14.

Tabela 5.14 – Custos das compressões possíveis no 1º cenário do caso 1 (versão exponencial convexa)

Variáveis	Expressão de desenvolvimento	Equação
<i>C; p = 7</i>	$Cm = -(31 - 23). e^{-(19-11-7)} = -8. e^{-1} = 2,94$	(4.2)
	$Ri = 8. (7) - 2,94 = 56 - 2,94 = 53,06$	(4.7)
	$Custo Final = 1157 - 53,06 = 1103,94$	
<i>G; p = 6</i>	$Cm = -(135 - 90). e^{-(22-13-6)} = -45. e^{-3} $ $= 2,24$	(4.2)
	$Ri = 8. (6) - 2,24 = 48 - 2,24 = 45,76$	(4.7)
	$Custo Final = 1157 - 45,76 = 1111,24$	
<i>E; p = 6</i>	$Cm = -(159 - 105). e^{-(11-5-6)} = -54. e^0 = 54$	(4.2)
	$Ri = 8. (6) - 54 = 48 - 54 = -6$	(4.7)
	$Custo Final = 1157 - (-6) = 1163$	

Fonte: Elaborado pelo autor (2017).

Portanto a melhor escolha foi, coincidentemente, o conjunto (C) por propiciar uma redução maior no custo final do projeto, como mostra o cenário da primeira iteração na Tabela 5.15 a seguir.

Tabela 5.15 – Cenário 1ª iteração – (Caso 1, Versão Exp. Convexa)

Acelerando Conjunto { C } em 7 unidades de tempo - (Iteração: 1)								
Atv	D Nor	D Atu	D Acel	C Acel	C Atu	C Marg	Cedo	Tarde
(A)	11	11	6	40	30.00	2.00	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	11	11
(E)	11	11	5	159	105.00	9.00	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	22	13	135	90.00	5.00	23	23
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	14	9	135	130.00	1.00	25	31
#Atividade totalmente acelerada								
Custos: Indiretos = 360 Diretos = 743,94 Projeto = 1103.94								
Caminhos		Durações		Folgas				
ADG		45		0		→ Crítico		
BEG		45		0		→ Crítico		
BH		24		21				
CEG		45		0		→ Crítico		
CFI		39		6				
CH		24		21				

Fonte: Elaborado pelo autor (2017).

A seguir, serão mostrados os cenários da segunda, terceira e quarta da iteração, respectivamente, nas tabelas 5.16, 5,17 e 5.18.

Tabela 5.16 – Cenário 2ª iteração – (Caso 1, Versão Exp. Convexa)

Acelerando Conjunto { G } em 6 unidades de tempo - (Iteração: 2)								
Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30.00	2.00	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	11	11
(E)	11	11	5	159	105.00	9.00	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	16	13	135	92.24	2.24	23	23
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	14	9	135	130.00	1.00	25	25
#Atividade totalmente acelerada								
Custos: Indiretos = 312 Diretos = 746,18 Projeto = 1058,18								
Caminhos		Durações		Folgas				
ADG		39		0		→ Crítico		
BEG		39		0		→ Crítico		
BH		24		15				
CEG		39		0		→ Crítico		
CFI		39		0		→ Crítico		
CH		24		15				

Fonte: Elaborado pelo autor (2017).

Tabela 5.17 – Cenário 3ª iteração – (Caso 1, Versão Exp. Convexa)

Acelerando Conjunto { A E I } em 3 unidades de tempo - (Iteração: 3)								
Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	8	8
(E)	11	8	5	159	107.69	2.69	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	16	13	135	92.24	2.24	20	20
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	11	9	135	130.68	0.68	25	25
#Atividade totalmente acelerada								
Custos: Indiretos = 288 Diretos = 750,90 Projeto = 1038,90								
Caminhos		Durações		Folgas				
ADG		36		0		→ Crítico		
BEG		36		0		→ Crítico		
BH		24		12				
CEG		36		0		→ Crítico		
CFI		36		0		→ Crítico		
CH		24		12				

Fonte: Elaborado pelo autor (2017).

Tabela 5.18 – Cenário 4ª iteração – (Caso 1, Versão Exp. Convexa)

Acelerando Conjunto { G I } em 1 unidade de tempo - (Iteração: 4)								
Atv	D Nor	D Atu	D Acel	C Acel	C Atu	C Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	8	8
(E)	11	8	5	159	107.69	2.69	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	15	13	135	96.09	6.09	20	20
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	10	9	135	131.84	1.84	25	25
#Atividade totalmente acelerada								
Custos: Indiretos = 280 Diretos = 755,91 Projeto = 1035,91								
Caminhos		Durações		Folgas				
ADG		35		0		→ Crítico		
BEG		35		0		→ Crítico		
BH		24		11				
CEG		35		0		→ Crítico		
CFI		35		0		→ Crítico		
CH		24		11				

Fonte: Elaborado pelo autor (2017).

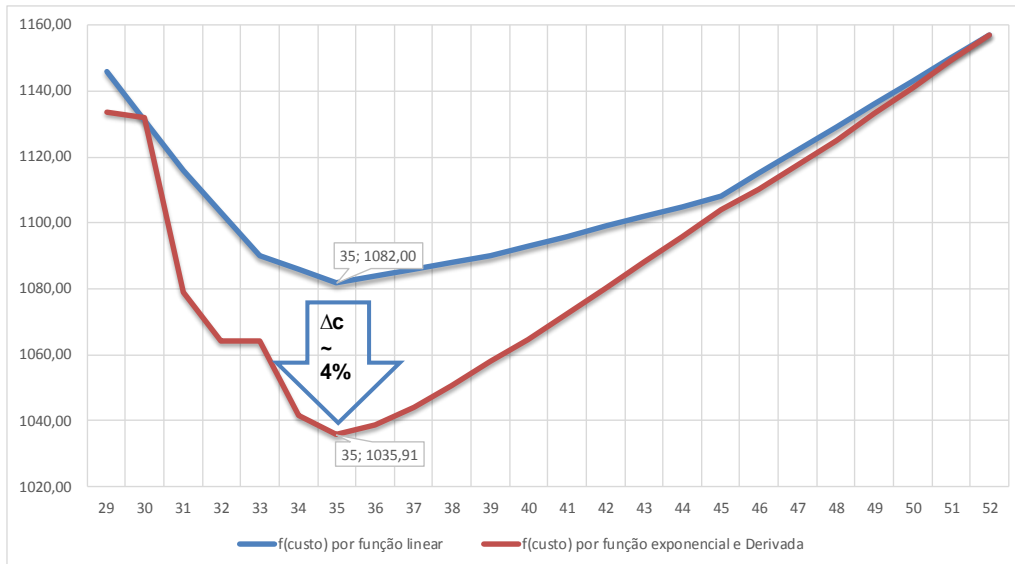
O artefato encontra o critério de parada nesse ponto, pois, diferentemente da versão linear, não aconteceu nenhuma atividade acelerada até o máximo de redução, porém a partir desse ponto, qualquer tentativa de aceleração- a função exponencial convexa do custo por atividade, eleva o custo final do projeto para além do mínimo encontrado na última iteração. Portanto o critério de parada foi alcançado com o custo mínimo do projeto em 1.035,91 unidades monetárias.

O relatório completo do *Crashing* do estudo de caso 1 na versão exponencial convexa pode ser conferido no Apêndice E.

A Figura 5.1 mostra um gráfico comparativo do *Crashing* feito nas duas versões do artefato. Para realizar um comparativo completo dos custos e durações, o artefato foi executado com prazo final igual a zero como critério de parada. A compressão parte de 52 unidades de tempo, no caminho crítico inicial, indo até 29. A curva em azul mostra a variação do

custo final do projeto na versão linear e a curva em vermelho, mostra a variação do custo final do projeto na versão exponencial convexa.

Figura 5.1 – Gráfico comparativo das compressões linear e exponencial convexa (Caso 1)



Compressão total ==>	ΔC	0,95%
		2,01%
	ΔT	44,23%

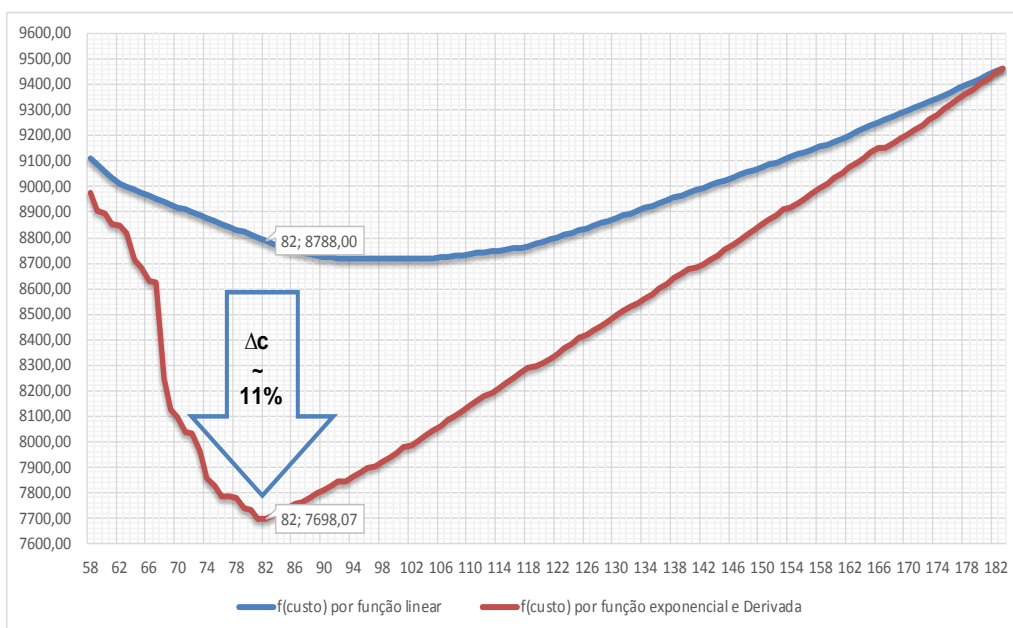
Fonte: Elaborado pelo autor (2017).

5.4. Caso 2 – Versão Exponencial Convexa

Neste estudo de caso, o artefato teve um pouco mais de trabalho para levar o *Crashing* a atingir a função objetivo. O ponto de custo mínimo do projeto aconteceu na décima quinta iteração ao custo de 7698,06 unidades monetárias, contra onze da versão linear, ao custo de 8717 unidades monetárias. A máxima compressão aconteceu na vigésima nona iteração, contra 19 da versão linear. Em contrapartida, a variação do custo é maior no ponto de mínimo, como pode ser visto no gráfico da Figura 5.2. Estes dados podem ser conferidos no relatório completo desse estudo de caso no Apêndice F.

A Figura 5.2 mostra um gráfico comparativo das duas versões do *Crashing* para o estudo de caso 2. Para realizar um comparativo completo dos custos e durações, o artefato foi executado com prazo final como critério de parada. A compressão parte de 183 unidades de tempo, no caminho crítico inicial, indo até 82. A curva em azul mostra a variação do custo final do projeto na versão linear, já a curva em vermelho, mostra a variação do custo final do projeto na versão exponencial convexa.

Figura 5.2 - Gráfico comparativo das compressões linear e exponencial convexa (Caso 2)



Compressão total ==>	ΔC	3,71%
		5,16%
	ΔT	68,31%

Fonte: Elaborado pelo autor (2017).

6 QUANTIFICAÇÃO DOS RISCOS

A simulação pode mostrar resultados que ainda não são conhecidos quando as premissas e as regras do objeto simulado resultam num sistema complexo. A quantificação dos riscos possui esta motivação. “Conhecendo os princípios que governam o comportamento de um Sistema com poucas variáveis, é possível prever seu funcionamento. Mas com um grande número de variáveis precisamos de um sistema digital para extrair as implicações da interação entre elas. A simulação pode nos dizer coisas que ainda não sabemos” (SIMON, 1981).

Com o propósito de realizar uma simulação simples, porém com confiabilidade, realizou-se a quantificação dos riscos utilizando gerador de números aleatórios e funções gráficas do ambiente *Microsoft Excel*®.

A quantificação necessita de um número de amostras aleatórias (N), que foi calculado de acordo com a equação 2.1. Como o desvio padrão depende da amostra e é ela que se quer calcular, tira-se o desvio padrão da equação usando-se o erro máximo da amostragem como uma porcentagem do desvio padrão. O grau de confiança escolhido (Z) foi de 95% (Tabela 2.1).

$$N = \left(\frac{Z \cdot \sigma}{E}\right)^2 = \left(\frac{1,96 \cdot \sigma}{0,05 \cdot \sigma}\right)^2 = \left(\frac{1,96}{0,05}\right)^2 = (39,2)^2 = 1536,64 \sim 1540 \quad (2.1)$$

Para esta quantificação foi escolhida a variável tempo. Portanto, para a construção dos gráficos de simulação no ambiente de planilha eletrônica, foram gerados 1540 números aleatórios entre a mínima e a máxima durações esperadas para o caminho crítico do projeto. Para isso, precisa-se executar o artefato com o prazo final igual a zero como critério de parada, ou outro valor, caso se deseje outro intervalo de durações mínima e máxima do caminho crítico.

Nesse trabalho, o critério de parada escolhido foi o prazo final igual a zero, para se obter a máxima compressão possível e, assim, conseguir os valores de máximo e mínimo do tempo de duração do projeto.

A duração máxima do projeto está no valor do caminho crítico do primeiro cenário antes das iterações do *Crashing* e a duração mínima está no valor do caminho crítico do último cenário.

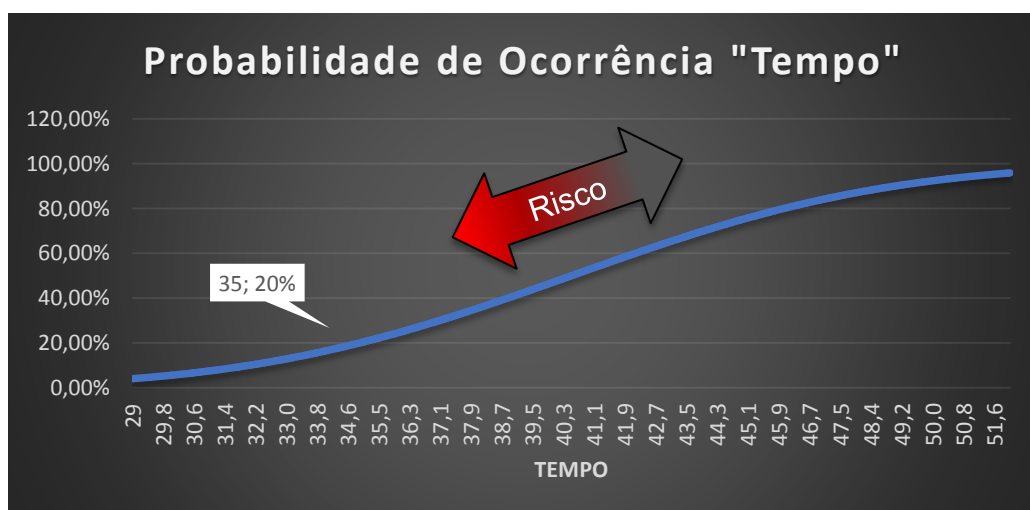
Como para cada tempo de duração tem-se um custo associado, basta, para cada probabilidade de tempo de conclusão encontrada pela simulação, associar o tempo mais próximo e o custo encontrados pelo artefato.

6.1. Quantificação do Risco do Estudo de caso 1

Como é mostrado na Figura 5.1, no estudo de caso 1, o *Crashing* comprimiu a duração do projeto, partindo de 52 para 29 unidades de tempo (Apêndices C e E) e foram esses os valores passados para a simulação.

Nesse caso, a versão do artefato não faz diferença, pois o ponto de custo mínimo encontrado para a variável tempo de duração, é o mesmo nas duas versões, sendo 35 unidades de tempo (lembrando que o ponto de custo mínimo estará entre a máxima e a mínima compressão). A Figura 6.1 mostra o gráfico resultante da simulação para o caso 1.

Figura 6.1 – Distribuição de Frequência Acumulada da Simulação do Estudo de Caso 1



Fonte: Elaborado pelo autor (2017).

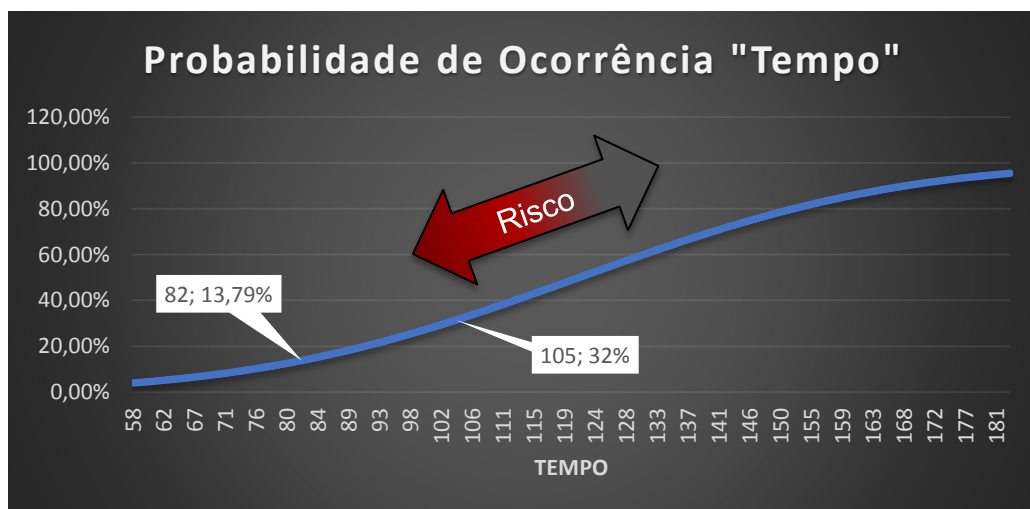
Usando as funções da planilha eletrônica, foram gerados 1540 números aleatórios entre 29 e 52 que são as durações do caminho crítico com nenhuma compressão ou compressão máxima, e em seguida foi construído o gráfico com as frequências acumuladas dos números aleatórios da amostra.

As frequências acumuladas mostraram que o ponto de custo mínimo do projeto em 35 unidades de tempo aconteceu em aproximadamente 20% das ocorrências. Portanto, a simulação quantifica um risco aproximado de 80% de não conseguir terminar o projeto acelerado ao custo mínimo, contra 20% de chances de sucesso.

6.2. Quantificação do Risco do Estudo de Caso 2

No estudo de caso 2, visto resumidamente no gráfico da Figura 5.2, o *Crashing* comprimiu a duração do projeto, partindo de 183 para 58 unidades de tempo (Apêndices D e F). O ponto onde o projeto atinge o menor custo é em cento e cinco (105) unidades de tempo na versão linear (Apêndice D) e oitenta e duas (82) unidades de tempo na versão exponencial convexa (Apêndice F). A Figura 6.2 mostra o gráfico resultante da simulação para o caso 2.

Figura 6.2 – Distribuição de Frequência Acumulada da Simulação do Estudo de Caso 2



Fonte: Elaborado pelo autor (2017).

Aqui foi usado o mesmo tamanho de amostra, ou seja 1540 números aleatórios entre 58 e 183 que são as durações do caminho crítico com nenhuma compressão ou compressão máxima, e em seguida foi construído o gráfico com as frequências acumuladas dos números aleatórios da amostra.

Na aproximação linear, as frequências acumuladas mostraram que o ponto de custo mínimo do projeto em 105 unidades de tempo aconteceu em aproximadamente 32% das ocorrências. Portanto, a simulação quantifica um risco aproximado de 68%% de não conseguir terminar o projeto acelerado ao custo mínimo, contra 32% de chances de sucesso.

Na aproximação exponencial convexa, as frequências acumuladas mostraram que o ponto de custo mínimo em 82 unidades de tempo aconteceu em aproximadamente 14% das ocorrências. Portanto, a simulação mostrou um risco aproximado de 86% de não conseguir terminar o projeto, contra 14% de chances de sucesso.

Como já era previsto, o destaque do risco na Figura 5.2 mostra que quanto mais se comprime a estimativa de tempo de duração para se completar o projeto, aumentam os riscos de fracasso e diminuem as probabilidades de sucesso.

7 CONCLUSÕES

O objetivo geral desta pesquisa foi plenamente atendido por meio da sistematização computacional do algoritmo de *Crashing* ao abranger os problemas de TCT. O objetivo geral mostrou-se cumprido sobretudo pela identificação na diferença dos resultados quando a abordagem é feita por aproximação linear ou exponencial convexa. Após o *Crashing*, a quantificação dos riscos mostrou as diferentes probabilidades de sucesso ou fracasso diante das incertezas das estimativas de tempo e custo.

Como os problemas de TCT partem do pressuposto básico de que estas estimativas são geradas pela equipe de projeto, as incertezas associadas a elas foram cobertas pela quantificação dos riscos que, com a ajuda da aplicação do método Monte Carlo, demonstrou-se múltiplos cenários aleatórios com diferentes probabilidades.

A utilidade do artefato como ferramenta de otimização de projetos é comprovada na dificuldade em se executar o *Crashing* sem o apoio computacional. O estudo de caso 2, elucida bem esta dificuldade no projeto hipotético com 23 atividades e um número inviável de caminhos críticos para uma abordagem manual.

O artefato executa o trabalho árduo de enumerar todas as combinações possíveis de conjuntos de aceleração com as atividades dos caminhos críticos. Fazer essa enumeração manualmente em um projeto com até 10 atividades já é trabalhoso. Acima disso é inviável e a dificuldade cresce exponencialmente, tornando o trabalho humano quase impossível.

Os exemplos encontrados na bibliografia, juntamente com os estudos de caso corroboram esta argumentação onde o número de atividades raramente passa de dez (10).

Como resultado deste trabalho, originou-se uma ferramenta de otimização e quantificação de risco que pode ser disponibilizada para uso experimental no INPE com dados reais de projetos, sobretudo para que

possa fornecer subsídios adicionais aos tomadores de decisão no gerenciamento de recursos.

O artefato de otimização produz informações de apoio aos gerentes e tomadores de decisão que, por sua vez podem acompanhar a variabilidade dos custos e dos prazos estipulados podendo decidir ou não pela otimização em face das características contratuais e econômicas de determinado projeto. A quantificação dos riscos amplia o poder de se decidir por um dentre dois caminhos possíveis, que são: nada fazer e esperar pelos acontecimentos assumindo uma posição reativa frente aos resultados, ou então, adotar uma posição proativa fazendo um planejamento de resposta aos efeitos dos riscos.

O código do artefato de *software* foi desenvolvido em linguagem de programação C, que é uma linguagem que pode ser empregada do mais simples ao mais complexo artefato de *software* e que tem a vantagem de possuir uma grande comunidade internacional de programadores experientes e atuantes.

Portanto, se o código fonte do artefato puder ser disponibilizado sob uma licença de código aberto, cujo registro já vem sendo providenciado, poderá ser constantemente aperfeiçoado e incrementado com mais funcionalidades.

7.1. Limitações

O mapeamento binário para gerar as combinações possíveis de atividades aceleráveis possui uma limitação que se refere ao tamanho da palavra binária que é dependente do *hardware* e compilador utilizado. Nesse trabalho o artefato utiliza o tamanho máximo de palavra permitido pelo compilador C utilizado que é de 64 *bits*.

Como no mapeamento binário o artefato poderá contar no máximo de (1) até 2^{64} e ele utiliza um bit como teste de parada (capítulo 4), o resultado é um limite de 63, aqui nominado como núcleo de compressão. Isso significa que em um projeto hipotético, se o conjunto não redundante de

todas as atividades dos caminhos críticos, contiver mais do que 63 ocorrências, o artefato não executará.

Portanto, para vencer essa limitação, o módulo que trata essa função poderia trabalhar com vetor de palavras que por sua vez impõe outra limitação: um número muito grande de atividades pode tornar inviável essa abordagem dado o crescimento exponencial de interações entre as variáveis envolvidas.

Diante da bibliografia e amplas pesquisas realizadas em meio eletrônico juntamente com a dificuldade de se encontrar estudos de caso com mais de uma dezena de atividades, o artefato na condição atual poderia ser usado em muitos projetos de pequeno e médio porte. Em projetos de grande porte, as atividades poderiam ser agrupadas em pacotes de trabalho para vencer a limitação do núcleo de compressão.

7.2. Posicionamento do Artefato em Relação à Bibliografia

A motivação principal que gerou o artefato foi o de sistematizar o algoritmo de *Crashing*, que resolve heurísticamente os problemas de balanceamento entre tempo e custo (TCTP), quando inseridos no contexto de relacionamento linear mantendo, em cada atividade, o custo marginal constante durante o processo de compressão.

Posteriormente, como contribuição, foi usada uma função exponencial convexa para desenvolver a segunda versão do artefato. Dos exemplos encontrados na bibliografia, esta curva é a que melhor descreve o comportamento de evolução dos custos marginais em função do tempo.

A Tabela 7.1 a seguir mostra um quadro com o resumo da sistematização das duas versões do artefato.

Tabela 7.1 – Quadro comparativo da sistematização do *Crashing* por aproximação linear e convexa

Quadro comparativo das versões do artefato desenvolvido					
Ano-Author	Método	Nº Atividades	Segundos	Características	Observações
2014 Alexandre Balistrieri	Aproximação Linear de Contador (1998).	9 a 23	0,02 a 0,1	Verifica custos indiretos e custos marginais lineares de todas as combinações possíveis de atividades aceleráveis nos Caminhos Críticos.	Não garante a solução ótima. Limita o núcleo de Compressão a no máximo 63 atividades.
2016 Alexandre Balistrieri	Aproximação exponencial convexa como contribuição.	9 a 23	0,02 a 1	Análogo ao anterior porém aproximando Custo por função exponencial convexa.	Análogo ao anterior. Mais sensível quando se diminuem os custos indiretos.

Fonte: Elaborado pelo autor (2017).

Nas abordagens pesquisadas, sejam heurísticas ou meta-heurísticas, parte-se do pressuposto de que as estimativas de custos e tempo de duração das atividades são baseadas em estimativas válidas e incertezas estocásticas. As abordagens TCTP, e este trabalho também, partem do mesmo pressuposto.

7.3. Considerações finais

A modelagem computacional do *Crashing* nos dois estudos de caso mostrou resultados que não eram evidentes no início. No ponto de custo mínimo do projeto, a diferença de custo entre a aproximação linear e a exponencial convexa ficou próxima de 4% (Figura 5.1) no estudo de caso 1 e aproximadamente 11% (Figura 5.2) no estudo de caso 2.

No estudo de caso 1, a variação do tempo entre a compressão máxima e nenhuma compressão no caminho crítico do projeto foi de 44% e do custo foi de quase 1% na versão linear e 2% na versão exponencial convexa (Figura 5.1). No estudo de caso 2, a variação do tempo foi de 68% e do custo foi de aproximadamente 3,5% na versão linear, contra aproximadamente 5% na versão exponencial convexa (Figura 5.2).

Essa informação mostra que, independente da aproximação do custo marginal ser linear ou exponencial convexa, quando o projeto foi levado à

sua compressão máxima, fazendo todas as acelerações possíveis dos caminhos críticos, sempre ao menor incremento de custo, a variação no tempo foi consideravelmente maior no tempo em relação ao custo: tempo variou 44% no caso 1 e 68% no caso 2; custo variou 1% a 2% no caso 1 e entre 3,5% a 5% no caso 2.

De acordo com as estimativas de tempo e custo, se esse comportamento comprova uma tendência, abre-se espaço para outras abordagens de gerenciamento de projetos, em que pode-se trabalhar com a compressão máxima no tempo de duração do projeto, com pouquíssimo custo incremental mas uma folga administrável próxima ou superior a 50%.

Portanto, todas essas informações mostram que, dada a complexidade dos projetos, na maioria dos casos, o resultado de todas as interações possíveis entre suas variáveis não é algo que possa ser inferido de imediato sem o apoio computacional.

Em qualquer tomada de decisão sempre haverá intrinsecamente algum grau de incerteza. Em redes de projetos, qualquer atividade que se inicia, implica em entrega futura, por mais imediata que possa parecer. As estimativas das atividades fornecidas pela equipe de projeto são baseadas geralmente na experiência e em amostras passadas. Logo, como jamais haverá 100% de certeza em qualquer tomada de decisão, esta ferramenta de otimização e quantificação de riscos fornece um horizonte mais tangível aos gerentes de projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

- AFSHAR, A.; ZIARATY, A. K.; KAVEH, A.; SHARIF, F. Nondominated archiving multicolony ant algorithm in time-cost trade-off optimization. **Journal of Construction Engineering and Management-Asce**, v.135, n.7, p. 668-674, 2009.
- AMINBAKHSH, S. **Hybrid particle swarm optimization algorithm for obtaining pareto front of discrete time-cost trade-off problem**. 2013. 106p. Thesis (Master of Science in Civil Engineering) - Middle East Technical University, Ankara, Turkey, 2013.
- ASHURI, B.; TAVAKOLAN, M. Fuzzy enabled Hybrid Genetic Algorithm – Particle Swarm Optimization approach to solve TCRO problems in construction project planning. **Journal of Construction Engineering and Management**, v.138, n. 9, p.1065-1074, 2012.
- BERSTEIN, P. L. **Desafio aos deuses**: a fascinante história do risco. Rio de Janeiro- RJ: Campus/Elsevier, 1997.
- BUSTOS, O. H.; FRERY, A. C. **Simulação estocástica**: teoria e algoritmos. Rio de Janeiro: IMPA, 1992.
- CARVALHO, M. M.; RABECHINI, R, Jr. **Fundamentos em gestão de projetos** – construindo competências para gerenciar projetos. São Paulo-SP, Editora Atlas, 2011.
- CHEN, S. P.; TSAI, M. J. Time-Cost Trade-off analysis of project networks in fuzzy environments. **European Journal of Operational Research**. n. 212, p.386–397, 2011.
- CHOI, B. C.; PARK, M. J. A continuous Time-Cost Tradeoff problem with multiple milestones and completely ordered jobs. **European Journal of Operational Research**. n. 244, p.748–752, 2015.

- COLORNI, A.; DORIGO, M.; MANIEZZO, V. Distributed optimization by Ant Colonies. **Actes de la première conférence européenne sur la vie artificielle**, Paris, France, Elsevier publishing, p. 134-142, 1991.
- CONTADOR, J. L. Gerenciamento de projetos com PERT e CPM. In: CONTADOR, J. C. **Gestão de operações**. São Paulo-SP, Editora Edgard Blucher, 1998. Cap. 20, p. 257-285.
- CONTADOR, J. L. **O problema CPM com funções de custo especiais**. 1988, 187p. Tese (Doutorado em Pesquisa Operacional) - Tecnológico de Aeronáutica (ITA), São José dos Campos-SP, 1988.
- DEMEULEMEESTER, E. L.; HERROELEN, W. S.; ELMAGHARABY, S. E. Optimal procedures for the Discrete Time/Cost Trade-off Problem in project networks. **European journal of operational research**, p.50-68, 1996.
- DEMEULEMEESTER, E. L.; HERROELEN, W. S. **Project scheduling – a research handbook**. Kluwer Academic Publishers. 2002.
- DEVROYE, L. (1986). **Non-uniform random variate generation**. New York, Springer-Verlag, 1986.
- FENG, C. W.; LIU, L.; BURNS, S. A. Using genetic algorithms to solve construction time-cost trade-off problems. **Journal of Computing in Civil Engineering**, 11(3), p. 184-189, 1997.
- FENG, C. W.; LIU, L.; BURNS, S. A. Stochastic construction time-cost trade-off analysis. **Journal of Computing in Civil Engineering**, v. 14, n. 2, p. 117-126, 2000.
- FONSECA, J. S.; MARTINS, G. A. **Curso de estatística**. São Paulo - SP, Editora Atlas, 1982.
- FOLDES, S.; SOUMIS, F. Pert and crashing revisited – mathematical generalizations. **European Journal of Operational Research**, v, 64, n. 2, p. 286-294, 1993.

FULKERSON, D. R. A network flow computation for Project Cost Curves. *Management Science*, v. 7, 2, p. 167-178, 1961.

GARBI, G. P. **Técnica da estimativa conjugada para o planejamento de cronogramas, orçamentos e quantificação dos riscos dos projetos**. 2016. 297 p. IBI: <8JMKD3MGP3W34P/3KK9J35>. (sid.inpe.br/mtc-m21b/2015/11.17.14.38-TDI). Tese (Doutorado em Engenharia e Gerenciamento de Sistemas Espaciais) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2016. Disponível em: <<http://urlib.net/8JMKD3MGP3W34P/3KK9J35>>.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. São Paulo – SP: Artmed Editora S. A., 2010.

HILLSON, D. **Effective opportunity management for projects** – exploiting positive risk. Taylor and Francis Group, 2004.

HOLLAND, J. H. **Adaptation in natural and artificial systems** – an introductory analysis with applications to biology, control and artificial intelligence. University of Michigan Press, 1975.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). **Plano diretor 2011 – 2015**. São José dos Campos: Ministério da Ciência, Tecnologia e Informação, 2011.

KELLEY, J. E. Critical-path planning and scheduling – mathematical basis. **Operations Research**, v. 9, n. 3, p. 296-320, 1961.

KELLEY, J. E.; WALKER, M. R. Critical-path planning and scheduling. In: EASTERN JOINT COMPUTER CONFERENCE, 1959, Boston, Mass. **Proceedings...** Boston: IRE-AIEE-ACM, 1959. p. 160-173.

KERZNER, H. **Project management** – a systems approach to planning, scheduling and controlling. New Jersey. Published John Wiley & Sons Inc. 2009.

LATHI, B. P. **Sinais e sistemas lineares**. Porto Alegre – RS, Bookman, 2007.

PROJECT MANAGEMENT INSTITUTE (PMI). **A Guide to the Project Management Body of Knowledge (PMBok® guide)**. Pennsylvania: Publised by PMI, 2013.

RIPLEY, B. D. **Stochastic simulation**. New York. Wiley, 1987.

SIEMENS, N. A simple CPM time-cost tradeoff algorithm. **Management Science**, v.17, n. 6, p. b354-b363, 1971.

SIMON, H. A. **As ciências do artificial**. Coimbra Portugal: Armenio Amado – Editor, Sucessor, 1981.

YANG, I. T. Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis. **Journal of Construction Engineering and Management – ASCE**. v.133, 7, p. 498-505, 2007.

ZHANG, H.; XING, F. Fuzzy-multi-objective particle swarm optimization for time-cost-quality tradeoff in construction. **Automation in Construction**, v. 19, n 8, p. 1067-1075, 2010.

ZHENG, D. X. M.; NG, S. T. Stochastic time-cost optimization model incorporating fuzzy sets theory and nonreplaceable front. **Journal of Construction Engineering and Management – ASCE**. v. 131, 2, p. 176-186, 2005.

APÊNDICE A – Artefato de Software Crashing Versão Linear

```
/* Artefato de Software para Otimização de Redes de Projetos
 * aplicado ao TCTP por aproximação Linear
 * Por: Alexandre Balistrieri - Out/2013
 *
 * Última atualização - 06/03/2017
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define HOMEUSR "/Users/bali/Dropbox/meus_dados"
#define HOMEDIR HOMEUSR"/Acelerador_deProjetos"
#define WORKDIR HOMEDIR"/dados"
#define ARRAY_SIZE 500
#define SERVE 0
#define NAOSERVE 1
#define MAX_ATIVS 63
#define MAX_INT 999999999
#define ASC_N(x) x - 48
#define ASC_A(x) x + 48
#define MIN(X,Y) ((X)<(Y))?(X):(Y)
#define MAX(X,Y) ((X)>(Y))?(X):(Y)
#define CONTINUA 0
#define PARA 1
#define USOU 1
#define NAO_USOU 0
#define LIMPO 0
#define MARCADO 1
#define CUSTO_MINIMO 0
#define PRAZO_FINAL 1

typedef unsigned char byte;
FILE *out;

/* Estrutura que armazenará a entrada de dados */
typedef struct {
    byte ativ[2];
    byte prec[ARRAY_SIZE];
    int d_a, d_n, d_at, c_a, c_n;
    float c_m;
} REGISTRO;

/* Contexto da aceleração */
struct {
    byte conj[ARRAY_SIZE];
    float custo_m;
    int compressao;
} acel;

struct {
    byte filho[ARRAY_SIZE];
    int data_tarde;
} pai[ARRAY_SIZE];

/* Serão armazenados um máximo de 'ARRAY_SIZE' registros de entrada*/
REGISTRO t[ARRAY_SIZE];

/* Limites dos arrays */
int otp, ncp, nulas, ncc, sz_ativs, sz_vetor, acelerador, alvo;

/* Totalizações */
int soma, custo_indireto, custo_direto_n, custo_direto_a, custo_f, custo_total,
prazo, npais, data_t[ARRAY_SIZE], data_T[ARRAY_SIZE], fl_c[ARRAY_SIZE],
duracao_c[ARRAY_SIZE], iteracao;

/* Arrays de índices e caminhos */
int indices_cc[ARRAY_SIZE];
byte caminho[ARRAY_SIZE][ARRAY_SIZE], vetor[ARRAY_SIZE], ativs[ARRAY_SIZE],
filhos[MAX_ATIVS], tracos[81];

/* Prototipagem */
int VerificarArgumentos(int n, char **args),
```

```

    Filtar_Aceleracao(byte *str),
    Filtro_CaminhoCritico(byte *str),
    Filtro_CustoMinimo(byte *str);

void EncontrarFilhos(byte str),
    EncontrarCompressao(byte *str);

void Preparacao(),
    EntrarDados(),
    GravarSaida(),
    VerificarCriterio_Parada(),
    EncontrarCaminhos(),
    AgruparIrmaos(),
    Acelerar(),
    Duracoes_E_Folgas(),
    DataCedo(),
    DataTarde(),
    MontarAtividades(),
    GerarConjuntos_Aceleracao(),
    ClassificarAtividades();
/* Fim Prototipagem */

int main(int argc, char**argv) {
    int status;

    memset(tracos, '-', sizeof (tracos));
    tracos[80] = NULL;

    status = VerificarArgumentos(argc, argv);
    if (status != 0) return (1);

    out = fopen(WORKDIR"/saida_bali.txt", "w");
    Preparacao();

    fprintf(out, "Legendas:\t# => Atividade totalmente acelerada\n");
    fprintf(out, "Custo Fixo = %d | Prazo = %d\n", custo_f, prazo);
    fprintf(out, "Critério de parada escolhido: %s\n",
        (alvo == CUSTO_MINIMO) ? "Custo mínimo" : "Prazo");
    fprintf(out, "Aceleração Total = %d\n", custo_direto_a);
    fprintf(out, "Versão = Linear\n");
    fprintf(out, "%s\n\n", tracos);

    acelerador = CONTINUA;
    iteracao = 0;
    while (acelerador == CONTINUA) {
        Acelerar();
        GravarSaida();
    }
    fclose(out);
    return 0;
}

int VerificarArgumentos(int n, char **args) {

    if (n < 3) {
        printf("Preciso dos argumentos 'custo n' ou 'prazo n m'\n");
        return (1);
    }
    custo_f = prazo = 0;
    if (strcmp(args[1], "custo") == 0) {
        sscanf(args[2], "%d", &custo_f);
        alvo = CUSTO_MINIMO;
        return (0);
    }
    if (strcmp(args[1], "prazo") == 0) {
        sscanf(args[2], "%d", &prazo);
        sscanf(args[3], "%d", &custo_f);
        alvo = PRAZO_FINAL;
        return (0);
    }
    printf("Preciso dos argumentos 'custo n' ou 'prazo n'\n");
    return (1);
}

void Preparacao() {

    EntrarDados();

```

```

    EncontrarCaminhos();
    AgruparIrmaos();
}

void Acelerar() {
    Duracoes_E_Folgas();
    DataCedo();
    DataTarde();
    MontarAtividades();
    GerarConjuntos_Aceleracao();
}

void EntrarDados() {
    FILE *in;
    REGISTRO reg;
    int i;

    custo_direto_n = custo_direto_a = 0;

    in = fopen(WORKDIR"/tabela_atividades.txt", "r");

    fscanf(in, "%d", &otp); /* Lê qtde de registros da tabela */

    for (i = 0; i <= otp - 1; i++) {
        fscanf(in, "%s \t%s \t%d \t%d \t%d \t%d \t%d",
            reg.ativ, reg.prec, &reg.d_a, &reg.d_n,
            &reg.c_a, &reg.c_n);

        custo_direto_n = custo_direto_n + reg.c_n;
        custo_direto_a = custo_direto_a + reg.c_a;

        reg.d_at = reg.d_n;
        reg.c_m = ((reg.d_n != reg.d_a) && (reg.c_n != reg.c_a)) ?
            (float) (reg.c_a - reg.c_n) / (reg.d_n - reg.d_a) : (0);

        t[i] = reg;
    }
    fclose(in);
}

void GravarSaida() {
    int i, j, k;
    byte *critico = "*", *plural = "s", *max = "#";

    fprintf(out,
        "Atv\tD_Nor\tD_Atu\tD_Acel\tC_Acel\tC_Atu\tC_Marg\tCedo\tTarde\n");
    for (i = 0; i < otp; i++) {
        max = ((t[i].d_at == t[i].d_a) ? ("#") : (" "));
        fprintf(out, "(%s)\t%d\t%d%s\t%d\t%d\t%.2f\t%d\t%d\n",
            t[i].ativ, t[i].d_n, t[i].d_at, max, t[i].d_a,
            t[i].c_a, t[i].c_n, t[i].c_m, data_t[i], data_T[i]);
    }
    custo_indireto = custo_f * soma;
    custo_total = custo_direto_n + custo_indireto;

    fprintf(out, "\nCustos > Indiretos: %d Diretos: %d "
        "Projeto: %d Prazo: %d\n",
        custo_indireto, custo_direto_n, custo_total, soma);

    fprintf(out, "%-18s%-12s\n", "Caminhos", "Durações Folgas");
    for (i = 0; i < ncp; i++) {
        for (j = 0; j < strlen(caminho[i]); j++) {
            fprintf(out, "%s", t[ASC_N(caminho[i][j])].ativ);
        }
        critico = ((fl_c[i] == 0) ? ("=> Crítico") : (""));
        for (k = 0; k < 20 - j; k++) fprintf(out, " ");
        fprintf(out, "%3d %8d %s\n",
            duracao_c[i], fl_c[i], critico);
    }
    fprintf(out, "\n%s\n", tracos);
}

```

```

VerificarCritério_Parada();

if (acelerador == PARA) return (0);
else {
    for (i = 0; i < strlen(acel.conj); i++) {
        k = ASC_N(acel.conj[i]);
        t[k].d_at = t[k].d_at - acel.compressao;
    }
}
/* Imprime aceleração no arquivo de saída */
iteracao = iteracao + 1;
fprintf(out, "Acelerando Conjunto { ");
for (i = 0; i < strlen(acel.conj); i++) {
    k = ASC_N(acel.conj[i]);
    fprintf(out, "%s ", t[k].ativ);
}
plural = (acel.compressao >= 2) ? ("s") : ("");
fprintf(out, " em %d unidade%s de tempo", acel.compressao, plural);
fprintf(out, " - (Iteração: %d)\n", iteracao);
}

void VerificarCritério_Parada() {
    int delta, cn, ca;

    cn = custo_direto_n;
    ca = custo_direto_a;
    if (cn == ca) printf("Custo bateu igual");
    if (alvo == CUSTO_MINIMO)
        if (acel.custo_m > custo_f || cn == ca)
            acelerador = PARA;
        else
            acelerador = CONTINUA;
    else {
        delta = soma - prazo;
        if ((delta < acel.compressao) || (cn == ca))
            acelerador = PARA;
        else
            if ((delta <= 0) || (cn == ca))
                acelerador = PARA;
            else
                acelerador = CONTINUA;
    }
}

void EncontrarCaminhos() {
    int i, j, ncp_varrendo, varre_caminhos;
    byte atv, aux[MAX_ATIVS];

    /* Atividades de precedência nula são as origens dos caminhos */
    memset(caminho, NULL, sizeof (caminho));
    memset(aux, NULL, sizeof (aux));
    ncp = 0;
    for (i = 0; i < otp; i++) {
        if (strcmp(t[i].prec, "nula") == 0) {
            caminho[ncp][0] = ASC_A(i);
            ncp = ncp + 1;
        }
    }
    /*Lê a tabela de caminhos possíveis com a origem dos caminhos e pega a
    última posição desse caminho. É atividade final de um caminho ou origem
    de algum caminho com atividade sucessora (filho). Procura por atividades
    sucessoras - filhos - dessa atividade origem. Se houver, as atividades
    estarão no vetor 'filhos'. Em seguida varre o vetor 'filhos' para gerar
    os novos caminhos.*/

    nulas = ncp_varrendo = ncp;
    varre_caminhos = CONTINUA;
    while (varre_caminhos == CONTINUA) {
        varre_caminhos = PARA;
        for (i = 0; i < ncp_varrendo; i++) {
            atv = t[ASC_N(caminho[i][strlen(caminho[i]) - 1])].ativ[0];
            EncontrarFilhos(atv);
            strcpy(aux, caminho[i]);
            /*Varre o vetor 'filhos' e anexa cada atividade 'filha'
            no caminho origem. Um novo caminho nasce quando o primeiro
            filho é anexado e/ou os demais são bifurcados.*/

```



```

        for (j = 0; j < strlen(filhos); j++) {
            varre_caminhos = CONTINUA;
            atv = filhos[j];
            if (j == 0)
                /*Anexar Caminho*/
                strncat(caminho[i], &atv, 1);
            else {
                /*Bifurcar Caminho*/
                strcpy(caminho[ncp], aux);
                strncat(caminho[ncp], &atv, 1);
                ncp++;
            }
        }
        ncp_varrendo = ncp;
    }
    /*Classifica em ordem crescente os caminhos. Não é necessário.
    Apenas melhora a leitura do relatório de saída*/
    for (i = 0; i < ncp - 1; i++)
        for (j = i + 1; j < ncp; j++) {
            if (strcmp(caminho[j], caminho[i]) < 0) {
                strcpy(aux, caminho[i]);
                strcpy(caminho[i], caminho[j]);
                strcpy(caminho[j], aux);
            }
        }
}

void EncontrarFilhos(byte str) {
    int i, j;
    byte aux;

    memset(filhos, NULL, sizeof (filhos));
    for (i = nulas; i < otp; i++) {
        for (j = 0; j < strlen(t[i].prec); j++) {
            if (str == t[i].prec[j]) {
                aux = ASC_A(i);
                strncat(filhos, &aux, 1);
                break;
            }
        }
    }
}

void AgruparIrmaos() {
    int i, j, k;
    byte aux;

    npais = 0;
    memset(pai, NULL, sizeof (pai));

    for (i = 0; i < otp; i++)
        for (j = i + 1; j < otp; j++)
            for (k = 0; k < strlen(t[j].prec); k++)
                if (t[i].ativ[0] == t[j].prec[k]) {
                    aux = ASC_A(j);
                    strncat(pai[i].filho, &aux, 1);
                    npais = i;
                    break;
                }
    npais++;
}

void Duracoes_E_Folgas() {
    /*
    Calcula duração de cada caminho em duracao_c[i];
    Calcula caminho de maior duração em duracao_cc;
    Marca cc[i] em 1 quando duracao_c[i] for a maior (C. Crítico);
    Coloca o índice 'i' do caminho crítico em indices_cc[ncc];
    Calcula o número de caminhos críticos em ncc;
    Calcula folga livre de cada caminho e coloca em fl_c[i].
    */
    int i, j, k, duracao_cc, y, x, l;

    duracao_cc = 0;
    for (i = 0; i < ncp; i++) {
        k = strlen(caminho[i]);

```

```

        duracao_c[i] = 0;
        for (j = 0; j < k; j++) {
            l = ASC_N(caminho[i][j]);
            duracao_c[i] = duracao_c[i] + t[l].d_at;
        }
        duracao_cc = MAX(duracao_c[i], duracao_cc);
    }
    ncc = 0;
    for (i = 0; i < ncp; i++) {
        if (duracao_c[i] == duracao_cc) {
            indices_cc[ncc] = i;
            ncc = ncc + 1;
        }
        fl_c[i] = duracao_cc - duracao_c[i];
    }
    soma = duracao_cc;
    /*Se houve aceleração, ajusta custo normal */
    custo_direto_n = 0;
    for (i = 0; i < otp; i++) {
        x = (t[i].d_at - t[i].d_a) * t[i].c_m;
        y = t[i].c_a - t[i].c_n;
        if (x != y) t[i].c_n = t[i].c_n + (y - x);
        custo_direto_n = custo_direto_n + t[i].c_n;
    }
}

void DataCedo() {
    int i, j, k, l, m;

    memset(data_t, 0, sizeof (data_t));
    for (i = 0; i < ncp; i++) {
        k = strlen(caminho[i]);
        for (j = 1; j < k; j++) {
            l = ASC_N(caminho[i][j]);
            m = ASC_N(caminho[i][j - 1]);
            data_t[l] = MAX(data_t[l], (data_t[m] + t[m].d_at));
        }
    }
}

void DataTarde() {
    int i, j, k, l, m, aux;

    for (i = 0; i < otp; i++) data_T[i] = soma;
    /* Encontra data tarde individual de cada atividade */
    for (i = 0; i < ncp; i++) {
        k = strlen(caminho[i]) - 1;
        for (j = k; j >= 0; j--) {
            l = ASC_N(caminho[i][j]);
            if (j == k) aux = soma;
            else {
                m = ASC_N(caminho[i][j + 1]);
                aux = data_T[m];
            }
            data_T[l] = MIN(data_T[l], (aux - t[l].d_at));
        }
    }

    for (i = 0; data_t[i] == 0; i++) data_T[i] = 0;
    for (i = 0; i < npais; i++) pai[i].data_tarde = soma;

    /* Encontra a menor data tarde entre as atividades irmãs */
    for (i = 0; i < npais; i++)
        for (j = 0; j < strlen(pai[i].filho); j++) {
            k = ASC_N(pai[i].filho[j]);
            pai[i].data_tarde = MIN(pai[i].data_tarde, data_T[k]);
        }

    /* Ajusta a data tarde entre as atividades irmãs */
    for (i = 0; i < npais; i++) {
        k = strlen(pai[i].filho);
        for (j = 0; j < k; j++) {
            l = ASC_N(pai[i].filho[j]);
            data_T[l] = pai[i].data_tarde;
        }
    }
}
}

```

```

void MontarAtividades() {
    int i, j, k, aux, mapa_ativs[ARRAY_SIZE];

    sz_vetor = 0;
    for (i = 0; i < ncc; i++)
        sz_vetor += strlen(caminho[indices_cc[i]]);
    memset(vetor, NULL, sz_vetor);
    /* Coloca todas atividades dentro do vetor */
    k = 0;
    for (i = 0; i < ncc; i++) {
        for (j = 0; j < strlen(caminho[indices_cc[i]]); j++)
            vetor[k + j] = caminho[indices_cc[i]][j];
        k = k + strlen(caminho[indices_cc[i]]);
    }
    ClassificarAtividades();

    /* Le as atividades do vetor e coloca sem repetição
    * no array ativs as atividades passíveis de aceleração.
    * No fim, a variavel 'sz_ativs' terá o tamanho de ativs,
    * e este terá todas as atividades a serem combinadas em
    * conjuntos de aceleração. */
    k = 0;
    memset(ativs, NULL, sizeof (ativs));
    memset(mapa_ativs, NAO_USOU, sizeof (mapa_ativs));
    for (i = 0; i < sz_vetor; i++) {
        if (mapa_ativs[ASC_N(vetor[i])] == NAO_USOU) {
            mapa_ativs[ASC_N(vetor[i])] = USOU;
            aux = ASC_N(vetor[i]);

            /*Descarta atividade se não houver delta de aceleração*/
            if (t[aux].d_at == t[aux].d_a) continue;

            ativs[k] = vetor[i];
            k++;
        }
    }
    sz_ativs = k;
    acel.custo_m = acel.compressao = MAX_INT;
}

void ClassificarAtividades() {
    int i, j, k, f1, f2;
    byte aux;

    k = sz_vetor - 1;
    for (i = 0; i < sz_vetor; i++) {
        for (j = 0; j < k; j++) {
            f1 = ASC_N(vetor[j]);
            f2 = ASC_N(vetor[j + 1]);
            if (f1 > f2) {
                aux = vetor[j];
                vetor[j] = vetor[j + 1];
                vetor[j + 1] = aux;
            }
        }
        k--;
    }
}

void GerarConjuntos_Aceleracao() {
    unsigned long paridade, mascara, cont ;
    int i, j, criterio;
    byte str[MAX_ATIVS] ;

    /*Coloca bit ligado em 1 na enésima posição da palavra 'paridade'
    (n = tamanho do string) e inverte a palavra para o teste de parada. */
    paridade = ~(1 << strlen(ativs)); /*Ex. n=5, paridade=>10000=>01111 */

    /* A cada iteração o campo cont terá uma combinação de bits desligados
    (zeros) e ligados (1s). As posições dos bits ligados darão as posições
    da string com as atividades que resultam em uma combinação possível. */
    cont = 1;
    while ( paridade & cont ) { /*Pra n=5, qdo cont = 10000 o loop termina. */
        mascara = 1 ;
        i = j = 0 ;
        /*Para cada combinação, a máscara começa com 1 e a cada iteração esse

```

```

'1' é deslocado uma posição à esquerda. Quando a posição desse '1' for
a mesma posição do bit que está em 'zero' do campo paridade, foi feito
o mapeamento da combinação corrente. */
while ( paridade & mascara ) {
    if ( cont & mascara ) {
        /* Gera a combinação em str */
        str[i] = ativs[j] ;
        i++ ;
    }
    j++ ;
    mascara = mascara << 1 ;
}
str[i]=0 ;

criterio = Filtrar_Aceleracao(str);
if (criterio == SERVE) {
    EncontrarCompressao(str);
    strcpy(acel.conj, str);
}
cont++;
}
return 0;
}

int Filtrar_Aceleracao(byte str[MAX_ATIVS]) {
    int criterio;

    criterio = Filtro_Redundancia(str);

    if (criterio == SERVE)
        criterio = Filtro_CaminhoCritico(str);
    else
        return (NAOSERVE);

    if (criterio == SERVE)
        criterio = Filtro_CustoMinimo(str);
    else
        return (NAOSERVE);

    return (criterio);
}

int Filtro_Redundancia(byte str[MAX_ATIVS]) {
    int i, j, k, qtde;

    /* Esse filtro garante que uma atividade não pode pertencer ao
    * caminho de outra atividade do mesmo conjunto de aceleração */
    for (i = 0; i < ncc; i++) {
        qtde = 0;
        for (j = 0; j < strlen(str); j++) {
            for (k = 0; k < strlen(caminho[indices_cc[i]]); k++)
                if (str[j] == caminho[indices_cc[i]][k])
                    qtde++;
        }
        if (qtde > 1) return (NAOSERVE);
    }
    return (SERVE);
}

int Filtro_CaminhoCritico(byte str[MAX_ATIVS]) {
    int i, j, k, qtde;

    /* Esse filtro garante que cada caminho crítico seja contemplado
    * com apenas uma atividade, verificando no final se a quantidade
    * de caminhos contemplados é igual ao número de caminhos
    * críticos (variável ncc) */
    qtde = 0;
    for (i = 0; i < ncc; i++) {
        for (j = 0; j < strlen(str); j++) {
            for (k = 0; k < strlen(caminho[indices_cc[i]]); k++) {
                if (str[j] == caminho[indices_cc[i]][k])
                    qtde++;
            }
        }
    }
    if (qtde == ncc) return (SERVE);
    else return (NAOSERVE);
}

```

```

}

int Filtro_CustoMinimo(byte str[MAX_ATIVS]) {
    int i, k, aux2, delta;
    float aux1;

    /* Esse filtro garante o registro do conjunto que possuir o menor
     * somatório dos custos marginais. Se houver empate dos custos
     * marginais então registrará o conjunto que propiciar maior
     * compressão. E se houver empate de compressão registrará o
     * conjunto de menor tamanho, pois por possuir menos atividades
     * em paralelo a acelerar possui risco menor de gerar mais
     * caminhos críticos */
    aux1 = 0;
    aux2 = MAX_INT;
    for (i = 0; i < strlen(str); i++) {
        k = ASC_N(str[i]);
        aux1 += t[k].c_m;
        delta = t[k].d_at - t[k].d_a;
        aux2 = MIN(aux2, delta);
    }
    /* Se custo marginal atual aux1 for menor que o custo marginal
     * registrado então serve: registra o conjunto atual e sai */
    if (aux1 < acel.custo_m) {
        acel.custo_m = aux1;
        acel.compressao = aux2;
        return (SERVE);
    }
    /* Se custo marginal atual aux1 for maior que o custo marginal
     * registrado então não serve e sai */
    if (aux1 > acel.custo_m) return (NAOSERVE);

    /* Custo marginal atual aux1 = custo marginal registrado */
    /* Se compressão atual aux2 for maior que compressão registrada
     * então serve: registra o conjunto atual e sai */
    if (aux2 > acel.compressao) {
        acel.custo_m = aux1;
        acel.compressao = aux2;
        return (SERVE);
    }
    /* Se compressão atual aux2 for menor que a compressão
     * registrada então não serve e sai */
    if (aux2 < acel.compressao) return (NAOSERVE);

    /* Compressão atual aux2 = compressão registrada */
    /* Se o tamanho do conjunto é menor ou igual ao anterior registrado
     * então registra o conjunto atual e sai, caso contrário atual não
     * serve e sai */
    if (strlen(str) <= strlen(acel.conj)) {
        acel.custo_m = aux1;
        acel.compressao = aux2;
        return (SERVE);
    } else return (NAOSERVE);
}

void EncontrarCompressao(byte str[MAX_ATIVS]) {
    int i, j, k, criterio;

    /* Esse módulo registra a melhor compressão buscando o mínimo
     * entre 'o menor delta duração normal - acelerada' e a 'menor
     * folga livre entre os caminhos que não contenham quaisquer
     * subconjunto de atividades do conjunto a acelerar'. Notar que
     * aqui, os parâmetros constantes SERVE e NAOSERVE são usados
     * apenas localmente. */
    for (i = 0; i < ncp; i++) {
        criterio = SERVE;
        for (j = 0; j < strlen(caminho[i]); j++) {
            for (k = 0; k < strlen(str); k++) {
                if (str[k] == caminho[i][j]) {
                    criterio = NAOSERVE;
                    break;
                }
            }
            if (criterio == NAOSERVE) break;
        }
        if (criterio == SERVE)
            acel.compressao = MIN(acel.compressao, fl_c[i]);
    }
}

```

```
}  
if (alvo == PRAZO_FINAL)  
    acel.compressao = MIN(acel.compressao, (soma - prazo));  
}
```

APÊNDICE B – Artefato de Software para o Crashing Versão Convexa

```
/* Artefato de Software para Otimização de Redes de Projetos
 * aplicado ao TCTP por aproximação Exponencial Convexa
 * Por: Alexandre Balistrieri - Jun/2016
 *
 * Última atualização - 06/03/2017
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>

#define HOMEUSR "/Users/bali/Dropbox/meus_dados"
#define HOMEDIR HOMEUSR"/NetBeansProjects/Crashing_PERT_CPM"
#define WORKDIR HOMEDIR"/dados"
#define ARRAY_SIZE 500
#define SERVE 0
#define NAOSERVE 1
#define ACABOU 0
#define NAOACABOU 1
#define MAX_ATIVS 63
#define MAX_INT 999999999
#define ASC_N(x) x - 48
#define ASC_A(x) x + 48
#define MIN(x,y) (((x)<(y))?(x):(y))
#define MAX(x,y) (((x)>(y))?(x):(y))
#define CONTINUA 0
#define PARA 1
#define CUSTO_MINIMO 0
#define PRAZO_FINAL 1
#define DESLIGADO 0
#define LIGADO 1
#define NAO_USOU 0
#define USOU 1
#define PRIMOGENITO 0
#define CUSTO(x,y,z) x - (y * (1 - exp(-z)))
#define C_M(x,y) x * exp(-y)

typedef unsigned char byte;
FILE *out;

/* Estrutura que armazenará a entrada de dados */
typedef struct {
    byte ativ[2];
    byte prec[ARRAY_SIZE];
    int d_a, d_n, d_at, c_a, coef;
    float c_n, c_m;
} REGISTRO;

/* Contexto da aceleração */
struct {
    byte conj[ARRAY_SIZE];
    int quis1, quis2, compressao, flag;
    float custo_m, reducao;
} acel;

struct {
    byte filho[ARRAY_SIZE];
    int data_tarde;
} pai[ARRAY_SIZE];

/* Serão armazenados um máximo de 'ARRAY_SIZE' registros de entrada*/
REGISTRO t[ARRAY_SIZE];

/* Limites dos arrays */
int otp,ncp,nulas,ncc,sz_ativs,sz_vetor,acelerador,alvo,the_end;

/* Totalizações */
int custo_f,soma,custo_indireto,custo_direto_a,iteracao,
    tmp_compressao,prazo,npais,data_t[ARRAY_SIZE],data_T[ARRAY_SIZE],
    fl_c[ARRAY_SIZE],duracao_c[ARRAY_SIZE];
```

```

float custo_direto_n, custo_total, custo_total_processado,
      custo_min, custo_max, tmp_cm, tmp_reducao;

/* Arrays de índices e caminhos */
int indices_cc[ARRAY_SIZE];
byte caminho[ARRAY_SIZE][ARRAY_SIZE],vetor[ARRAY_SIZE],ativs[ARRAY_SIZE],
      filhos[MAX_ATIVS],tracos[81];

/* Prototipagem */
int VerificarArgumentos(int n, char **args),
    Filtar_Aceleracao(byte *str),
    Filtro_CaminhoCritico(byte *str),
    Filtro_CustoMinimo(byte *str);

void EncontrarFilhos(byte str),
    RegistrarAceleracao(byte *str),
    EncontrarCompressao(byte *str),
    EncontrarMaiorReducao(byte *str);

void Preparacao(),
    EntrarDados(),
    GravarSaida(),
    VerificarCriterio_Parada(),
    EncontrarCaminhos(),
    AgruparIrmaos(),
    Acelerar(),
    Duracoes_E_Folgas(),
    DataCedo(),
    DataTarde(),
    MontarAtividades(),
    ProcessarAceleracao(),
    GerarConjuntos_Aceleracao(),
    ClassificarAtividades();
/* Fim Prototipagem */

int main(int argc, char**argv) {
    int status;

    memset(tracos,'-',sizeof(tracos));
    tracos[80] = NULL;

    status = VerificarArgumentos(argc,argv);
    if (status != 0) return(1);

    out = fopen(WORKDIR"/saida_bali.txt","w");
    acel.flag = DESLIGADO;
    the_end = NAOACABOU;

    Preparacao();

    fprintf (out,"Legendas:\t# => Atividade totalmente acelerada\n");
    fprintf (out,"Custo Fixo = %d | Prazo = %d\n",custo_f,prazo);
    fprintf (out,"Critério de parada escolhido: %s\n",
            (alvo == CUSTO_MINIMO)? "Custo mínimo":"Prazo Final");
    fprintf (out,"Aceleração Total = %d\n",custo_direto_a);
    fprintf (out,"Versão = Exponencial Convexa\n");
    fprintf (out,"%s\n\n",tracos);

    iteracao = 0;
    custo_min = MAX_INT;
    custo_max = 0;
    acelerador = CONTINUA;
    while (acelerador == CONTINUA) {
        Acelerar();
        GravarSaida();
    }

    /*printf("Mínimo = %.2f\tMáximo = %.2f\n",custo_min,custo_max);*/

    fclose(out);
    return 0;
}

int VerificarArgumentos(int n, char **args) {
    if (n < 3) {
        printf("Preciso dos argumentos 'custo n' ou 'prazo n m'\n");
    }
}

```



```

        return(1);
    }
    custo_f = prazo = 0;
    if (strcmp(args[1],"custo") == 0) {
        sscanf(args[2],"%d",&custo_f);
        alvo = CUSTO_MINIMO;
        return(0);
    }
    if (strcmp(args[1],"prazo") == 0) {
        sscanf(args[2],"%d",&prazo);
        sscanf(args[3],"%d",&custo_f);
        alvo = PRAZO_FINAL;
        return(0);
    }
    printf("Preciso dos argumentos 'custo n' ou 'prazo n'\n");
    return(1);
}

void Preparacao() {

    EntrarDados();

    EncontrarCaminhos();

    AgruparIrmaos();
}

void Acelerar() {

    Duracoes_E_Folgas();

    DataCedo();

    DataTarde();

    MontarAtividades();

    GerarConjuntos_Aceleracao();
}

void EntrarDados() {
    FILE *in;
    REGISTRO reg;
    int i;

    custo_direto_n = custo_direto_a = 0;

    in = fopen(WORKDIR"/tabela_atividades.txt","r");

    fscanf(in,"%d",&otp); /* Lê qtde de registros da tabela */

    for (i = 0; i <= otp - 1; i++) {
        fscanf(in, "%s %t%s %t%d %t%d %t%d %t%f",
            reg.ativ, reg.prec, &reg.d_a, &reg.d_n,
            &reg.c_a, &reg.c_n);

        reg.d_at = reg.d_n;
        reg.coef = reg.c_a - reg.c_n;
        reg.c_m = ((reg.d_n != reg.d_a) && (reg.c_n != reg.c_a)) ?
            (reg.c_a - reg.c_n) / (reg.d_n - reg.d_a) : (0);

        custo_direto_n = custo_direto_n + reg.c_n;
        custo_direto_a = custo_direto_a + reg.c_a;

        t[i] = reg;
    }
    fclose(in);
}

void GravarSaida() {
    int i,j,k;
    byte *critico="=> Critico",*plural="s", *max="#";

    fprintf(out,
        "Atv\tD_Nor\tD_Atu\tD_Acel\tC_Acel\tC_Atu\tC_Marg\tCedo\tTarde\n");
    for (i = 0; i < otp; i++) {
        max = ((t[i].d_at == t[i].d_a) ? ("#") : (""));
        fprintf(out, "(%s)\t%d\t%d%s\t%d\t%d\t%.2f\t%.2f\t%d\t%d\n",
            t[i].ativ,t[i].d_n,t[i].d_at,max,t[i].d_a,

```

```

        t[i].c_a,t[i].c_n,t[i].c_m,data_t[i],data_T[i]);
    }

    custo_indireto = custo_f * soma;
    custo_total = custo_direto_n + custo_indireto;
    custo_min = MIN(custo_min,custo_total);
    custo_max = MAX(custo_max,custo_total);
    fprintf(out,"\nCustos > Indiretos: %d Diretos: %.2f "
        "Projeto: %.2f Prazo: %d\n"
        ,custo_indireto,custo_direto_n,custo_total,soma);

    fprintf(out,"%-18s%-12s\n","Caminhos","Duracoes Folgas");
    for (i=0; i < ncp; i++) {
        for (j = 0; j < strlen(caminho[i]); j++) {
            fprintf(out,"%s",t[ASC_N(caminho[i][j])].ativ);
        }
        critico = ((fl_c[i] == 0) ? ("=> Crítico"):(""));
        for (k = 0; k < 20 - j; k++) fprintf(out," ");
        fprintf(out,"%3d      %3d\t%s\n",
            duracao_c[i],fl_c[i],critico);
    }
    fprintf(out,"%s\n",tracos);

    if (acel.flag == LIGADO)
        ProcessarAceleracao();

    VerificarCriterio_Parada();

    if (acelerador == PARA) return;

    /* Imprime aceleração no arquivo de saída */
    iteracao = iteracao + 1;
    fprintf(out,"\nAcelerando Conjunto { ");
    for (i = 0; i < strlen(acel.conj); i++) {
        k = ASC_N(acel.conj[i]);
        fprintf(out,"%s ",t[k].ativ);
    }
    plural = (acel.compressao > 1) ? ("s"):("");
    fprintf(out,") em %d unidade%s de tempo",acel.compressao,plural);
    fprintf(out," - (Iteração: %d)\n",iteracao);
}

void VerificarCriterio_Parada() {
    int i,delta,ca;
    float cn;

    cn = custo_direto_n;
    ca = custo_direto_a;

    if (the_end == NAOACABOU)
        if (alvo == CUSTO_MINIMO)
            if (acel.flag == DESLIGADO)
                acelerador = PARA;
            else
                acel.flag = DESLIGADO;
        else {
            delta = soma - prazo;
            if ((delta < acel.compressao) || (delta <= 0) || (cn == ca))
                acelerador = PARA;
            else
                acelerador = CONTINUA;
        }
    else
        acelerador = PARA;
}

void EncontrarCaminhos() {
    int i,j,ncp_varrendo,varre_caminhos;
    byte atv,aux[MAX_ATIVS];

    /* Atividades de precedencia nula são as origens dos caminhos */
    memset(caminho,NULL,sizeof(caminho));
    memset(aux,NULL,sizeof(aux));
    ncp = 0;
    for (i = 0; i < otp; i++) {
        if (strcmp(t[i].prec,"nula") == 0) {
            caminho[ncp][0] = ASC_A(i);
            ncp = ncp + 1;
        }
    }
}

```

```

}
/*Lê a tabela de caminhos possíveis com a origem dos caminhos e pega a
última posição desse caminho. É atividade final de um caminho ou origem
de algum caminho com atividade sucessora (filho). Procura por atividades
sucessoras - filhos - dessa atividade origem. Se houver, as atividades
estarão no vetor 'filhos'. Em seguida varre o vetor 'filhos' para gerar
os novos caminhos.*/

nulas = ncp_varrendo = ncp;
varre_caminhos = CONTINUA;
while (varre_caminhos == CONTINUA) {
    varre_caminhos = PARA;
    for (i = 0; i < ncp_varrendo; i++) {
        atv = t[ASC_N(caminho[i][strlen(caminho[i]) - 1]].ativ[0];
        EncontrarFilhos(atv);
        strcpy(aux, caminho[i]);
        /*Varre o vetor 'filhos' e anexa cada atividade 'filha'
        no caminho origem. Um novo caminho surge quando o primeiro
        filho é anexado (PRIMOGENITO) e/ou os demais são bifurcados.*/
        for (j = 0; j < strlen(filhos); j++) {
            varre_caminhos = CONTINUA;
            atv = filhos[j];
            if (j == PRIMOGENITO)
                /*Anexar Caminho*/
                strncat(caminho[i], &atv, 1);
            else {
                /*Bifurcar Caminho*/
                strcpy(caminho[ncp], aux);
                strncat(caminho[ncp], &atv, 1);
                ncp++;
            }
        }
    }
    ncp_varrendo = ncp;
}
/*Classifica em ordem crescente os caminhos. Não é necessário.
Apenas melhora a leitura do relatório de saída*/
for (i = 0; i < ncp - 1; i++)
    for (j = i + 1; j < ncp; j++) {
        if (strcmp(caminho[j], caminho[i]) < 0) {
            strcpy(aux, caminho[i]);
            strcpy(caminho[i], caminho[j]);
            strcpy(caminho[j], aux);
        }
    }
}
void EncontrarFilhos(byte str) {
    int i, j;
    byte aux;

    memset(filhos, NULL, sizeof(filhos));
    for (i = nulas; i < otp; i++) {
        for (j = 0; j < strlen(t[i].prec); j++) {
            if (str == t[i].prec[j]) {
                aux = ASC_A(i);
                strncat(filhos, &aux, 1);
                break;
            }
        }
    }
}
void AgruparIrmaos() {
    int i, j, k;
    byte aux;

    npais = 0;
    memset(pai, NULL, sizeof(pai));

    for (i = 0; i < otp; i++)
        for (j = i + 1; j < otp; j++)
            for (k = 0; k < strlen(t[j].prec); k++)
                if (t[i].ativ[0] == t[j].prec[k]) {
                    aux = ASC_A(j);
                    strncat(pai[i].filho, &aux, 1);
                    npais = i;
                    break;
                }
}

```

```

    npais++;
}
void Duracoes_E_Folgas() {
    /*
    Calcula duração de cada caminho em duracao_c[i];
    Calcula caminho de maior duração em duracao_cc;
    Marca cc[i] em 1 quando duracao_c[i] for a maior (C. Crítico);
    Coloca o índice 'i' do caminho crítico em indices_cc[ncc];
    Calcula o número de caminhos críticos em ncc;
    Calcula folga livre de cada caminho e coloca em fl_c[i].
    */
    int i,j,k,l,duracao_cc;

    duracao_cc = 0;
    for (i = 0; i < ncp; i++) {
        k = strlen(caminho[i]);
        duracao_c[i] = 0;
        for (j = 0; j < k; j++) {
            l = ASC_N(caminho[i][j]);
            duracao_c[i] = duracao_c[i] + t[l].d_at;
        }
        duracao_cc = MAX(duracao_c[i],duracao_cc);
    }
    ncc = 0;
    for (i = 0; i < ncp; i++) {
        if (duracao_c[i] == duracao_cc) {
            indices_cc[ncc] = i;
            ncc = ncc + 1;
        }
        fl_c[i] = duracao_cc - duracao_c[i];
    }
    soma = duracao_cc;
}
void DataCedo() {
    int i,j,k,u,p;

    memset(data_t,0,sizeof(data_t));
    for (i = 0; i < ncp; i++) { /*Varre todos caminhos possíveis*/
        k = strlen(caminho[i]);
        for (j = 1; j < k; j++) { /*Varre vetor caminho corrente*/
            u = ASC_N(caminho[i][j]); /*Idx última atividade corrente*/
            p = ASC_N(caminho[i][j - 1]); /*Idx penúltima atividade corrente*/
            data_t[u] = MAX(data_t[u],(data_t[p] + t[p].d_at));
        }
    }
}
void DataTarde() {
    int i,j,k,l,m,aux;

    for (i = 0; i < otp; i++) data_T[i] = soma;
    /* Encontra data tarde individual de cada atividade */
    for (i = 0; i < ncp; i++) {
        k = strlen(caminho[i]) - 1;
        for (j = k; j >= 0; j--) {
            l = ASC_N(caminho[i][j]);
            if (j == k) aux = soma;
            else {
                m = ASC_N(caminho[i][j + 1]);
                aux = data_T[m];
            }
            data_T[l] = MIN(data_T[l],(aux - t[l].d_at));
        }
    }

    for (i = 0; data_t[i] == 0; i++) data_T[i] = 0;
    for (i = 0; i < npais; i++) pai[i].data_tarde = soma;

    /* Encontra a menor data tarde entre as atividades irmãs */
    for (i = 0; i < npais; i++)
        for (j = 0; j < strlen(pai[i].filho); j++) {
            k = ASC_N(pai[i].filho[j]);
            pai[i].data_tarde = MIN(pai[i].data_tarde, data_T[k]);
        }

    /* Ajusta a data tarde entre as atividades irmãs */
    for (i = 0; i < npais; i++) {
        k = strlen(pai[i].filho);

```

```

        for (j = 0; j < k; j++) {
            l = ASC_N(pai[i].filho[j]);
            data_T[l] = pai[i].data_tarde;
        }
    }
}
void MontarAtividades(){
    int i,j,k,aux,mapa_ativs[ARRAY_SIZE];

    sz_vetor = 0;
    for (i = 0; i < ncc; i++)
        sz_vetor += strlen(caminho[indices_cc[i]]);
    memset(vetor,NULL,sz_vetor);
    /* Coloca todas atividades dentro do vetor */
    k = 0;
    for (i = 0; i < ncc; i++) {
        for (j = 0; j < strlen(caminho[indices_cc[i]]); j++)
            vetor[k + j] = caminho[indices_cc[i]][j];
        k = k + strlen(caminho[indices_cc[i]]);
    }
    ClassificarAtividades();

    /* Le as atividades do vetor e coloca sem repetição
     * no array ativs as atividades passíveis de aceleração.
     * No fim, a variavel 'sz_ativs' terá o tamanho de ativs,
     * e este terá todas as atividades a serem combinadas em
     * conjuntos de aceleração. */
    k = 0;
    memset(ativs,NULL,sizeof(ativs));
    memset(mapa_ativs,NAO_USOU,sizeof(mapa_ativs));
    for (i = 0; i < sz_vetor; i++) {
        if (mapa_ativs[ASC_N(vetor[i])] == NAO_USOU) {
            mapa_ativs[ASC_N(vetor[i])] = USOU;
            aux = ASC_N(vetor[i]);

            /*Descarta atividade se não houver delta de aceleração*/
            if (t[aux].d_at == t[aux].d_a) continue;

            ativs[k] = vetor[i];
            k++;
        }
    }
    sz_ativs = k;
    acel.custo_m = acel.compressao = MAX_INT;
    acel.redução = 0;
}
void ClassificarAtividades() {
    int i,j,k,f1,f2;
    byte aux;

    k = sz_vetor - 1;
    for (i = 0; i < sz_vetor; i++) {
        for (j = 0; j < k; j++) {
            f1 = ASC_N(vetor[j]);
            f2 = ASC_N(vetor[j + 1]);
            if (f1 > f2) {
                aux = vetor[j];
                vetor[j] = vetor[j + 1];
                vetor[j + 1] = aux;
            }
        }
        k--;
    }
}
void GerarConjuntos_Aceleracao() {
    unsigned long paridade, mascara, cont ;
    int i, j, criterio;
    byte str[MAX_ATIVS] ;

    /*Coloca bit ligado em 1 na enésima posição da palavra 'paridade'
    (n = tamanho do string) e inverte a palavra para o teste de parada. */
    paridade = ~(1 << strlen(ativs)); /*Ex. n=5, paridade=>10000=>01111 */

    /* A cada iteração o campo cont terá uma combinação de bits desligados
    (zeros) e ligados (1s). As posições dos bits ligados darão as posições
    da string com as atividades que resultam em uma combinação possível. */
    cont = 1;

```

```

while ( paridade & cont ) { /*Pra n=5, qdo cont = 10000 o loop termina. */
    mascara = 1 ;
    i = j = 0 ;
    /*Para cada combinação, a máscara começa com 1 e a cada iteração esse
    '1' é deslocado uma posição à esquerda. Quando a posição desse '1' for
    a mesma posição do bit que está em 'zero' do campo paridade, foi feito
    o mapeamento da combinação corrente. */
    while ( paridade & mascara ) {
        if ( cont & mascara ) {
            /* Gera a combinação em str */
            str[i] = ativs[j] ;
            i++ ;
        }
        j++ ;
        mascara = mascara << 1 ;
    }
    str[i]=0 ;

    critério = Filtrar_Aceleracao(str);
    if (critério == SERVE)
        RegistrarAceleracao(str);

    cont++ ;
}
}
int Filtrar_Aceleracao(byte str[MAX_ATIVS]) {
    int critério;

    critério = Filtro_Redundancia(str);

    if (critério == SERVE)
        critério = Filtro_CaminhoCritico(str);
    else
        return(NAOSERVE);

    if (critério == SERVE)
        critério = Filtro_CustoMinimo(str);
    else
        return(NAOSERVE);

    return(critério);
}
int Filtro_Redundancia(byte str[MAX_ATIVS]) {
    int i,j,k,qtde;

    /* Esse filtro garante que uma atividade não pode pertencer ao
    * caminho de outra atividade do mesmo conjunto de aceleração */
    for (i = 0; i < ncc; i++) {
        qtde = 0;
        for (j = 0; j < strlen(str); j++){
            for (k = 0; k < strlen(caminho[indices_cc[i]]); k++){
                if (str[j] == caminho[indices_cc[i]][k])
                    qtde++;
            }
            if (qtde > 1) return(NAOSERVE);
        }
    }
    return(SERVE);
}
int Filtro_CaminhoCritico(byte str[MAX_ATIVS]) {
    int i,j,k,qtde;

    /* Esse filtro garante que cada caminho crítico seja contemplado
    * com apenas uma atividade, verificando no final se a quantidade
    * de caminhos contemplados é igual ao número de caminhos
    * críticos (variável ncc) */
    qtde = 0;
    for (i = 0; i < ncc; i++) {
        for (j = 0; j < strlen(str); j++){
            for (k = 0; k < strlen(caminho[indices_cc[i]]); k++) {
                if (str[j] == caminho[indices_cc[i]][k]) qtde++;
            }
        }
    }
    if (qtde == ncc) return(SERVE);
    else return(NAOSERVE);
}
int Filtro_CustoMinimo(byte str[MAX_ATIVS]) {

```

```

/* Esse filtro garante o registro do conjunto que propiciar a maior
 * redução no custo indireto. O custo indireto é o custo fixo
 * multiplicado pela duração do caminho crítico. Como a duração do
 * caminho crítico varia de acordo com a compressão aplicada, cada uma
 * delas é aplicada ao custo fixo e ao custo marginal. Ou seja, cada
 * compressão é multiplicada pelo custo fixo e desse valor é subtraído
 * o somatório dos custos marginais calculado para a mesma compressão.
 */

EncontrarCompressao(str);

EncontrarMaiorReducao(str);

if (tmp_reducao >= acel.reducao)
    return(SERVE);
else
    return(NAOSERVE);
}

void EncontrarCompressao(byte str[MAX_ATIVS]) {
    int i,j,k,criterio;
    /* Esse módulo registra a melhor compressão buscando o mínimo
     * entre 'o menor delta duração normal - acelerada' e a 'menor
     * folga livre entre os caminhos que não contenham quaisquer
     * subconjunto de atividades do conjunto a acelerar'. Notar que
     * aqui, os parâmetros constantes SERVE e NAOSERVE são usados
     * apenas localmente. */

    acel.quisi1 = acel.quisi2 = MAX_INT;
    for (i = 0; i < strlen(str); i++) {
        k = ASC_N(str[i]);
        acel.quisi1 = MIN(acel.quisi1, (t[k].d_at - t[k].d_a));
    }

    for (i = 0; i < ncp; i++) {
        criterio = SERVE;
        for (j = 0; j < strlen(caminho[i]); j++) {
            for (k = 0; k < strlen(str); k++) {
                if (str[k] == caminho[i][j]) {
                    criterio = NAOSERVE;
                    break;
                }
            }
            if (criterio == NAOSERVE) break;
        }
        if (criterio == SERVE)
            acel.quisi2 = MIN(acel.quisi2, fl_c[i]);
    }

    if (alvo == PRAZO_FINAL)
        acel.quisi2 = MIN(acel.quisi2, (soma - prazo));

    tmp_compressao = MIN(acel.quisi1, acel.quisi2);
}

void EncontrarMaiorReducao(byte str[MAX_ATIVS]) {
    int i,j,k,compressao, reducao_indireta;
    float reducao, cm_reducao, aux;

    reducao = aux = 0;
    compressao = tmp_compressao;
    for (i = 1; i <= compressao; i++) {
        cm_reducao = 0;
        for (j = 0; j < strlen(str); j++) {
            k = ASC_N(str[j]);
            cm_reducao += (C_M(t[k].coef, (t[k].d_at - t[k].d_a - i)));
        }
        reducao_indireta = custo_f * i;
        if (cm_reducao < reducao_indireta) {
            reducao = (custo_f * i) - cm_reducao;
            if (reducao > aux) {
                aux = reducao;
                tmp_cm = cm_reducao;
                tmp_compressao = i;
            }
        }
        else
            break;
    }
}

```

```

    tmp_reducao = aux;
}
void RegistrarAceleracao(byte str[MAX_ATIVS]) {
    int i,k;

    strcpy(accel.conj,str);

    accel.custo_m = tmp_cm;
    accel.compressao = tmp_compressao;
    accel.reducao = tmp_reducao;
    accel.flag = LIGADO;
}
void ProcessarAceleracao() {
    int i,k,delta_t;

    /* Ajusta Custo Normal e Custo Marginal */
    for (i = 0;i < strlen(accel.conj);i++) {
        k = ASC_N(accel.conj[i]);
        delta_t = t[k].d_at - t[k].d_a - accel.compressao;
        t[k].d_at = t[k].d_at - accel.compressao;
        t[k].c_n = CUSTO(t[k].c_a,t[k].coef,delta_t);
        t[k].c_m = C_M(t[k].coef,delta_t);
    }
    custo_direto_n = 0;
    for (i = 0; i < otp; i++) custo_direto_n += t[i].c_n;
    custo_total_processado =
        custo_direto_n + custo_f * (soma - accel.compressao);
    if (alvo == CUSTO_MINIMO) {
        the_end =
            (custo_total_processado < custo_total) ? (NAOACABOU):(ACABOU);
    }
}
}

```


APÊNDICE C – Relatório Estudo de Caso 1 Versão Linear

Legendas: # => Atividade totalmente acelerada
 Custo Fixo = 8 | Prazo = 0
 Critério de parada escolhido: Prazo
 Aceleração Total = 979
 Versão = Linear

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	19	11	31	23	1.00	0	0
(D)	12	12	7	71	51	4.00	11	18
(E)	11	11	5	159	105	9.00	19	19
(F)	13	13	9	210	170	10.00	19	19
(G)	22	22	13	135	90	5.00	30	30
(H)	12	12	7	135	100	7.00	19	19
(I)	14	14	9	135	130	1.00	32	38

Custos > Indiretos: 416 Diretos: 741 Projeto: 1157 Prazo: 52

Caminhos	Durações	Folgas
ADG	45	7
BEG	45	7
BH	24	28
CEG	52	0 ==> Crítico
CFI	46	6
CH	31	21

 Acelerando Conjunto { C } em 7 unidades de tempo - (Iteração: 1)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	12	11	31	30	1.00	0	0
(D)	12	12	7	71	51	4.00	11	11
(E)	11	11	5	159	105	9.00	12	12
(F)	13	13	9	210	170	10.00	12	12
(G)	22	22	13	135	90	5.00	23	23
(H)	12	12	7	135	100	7.00	12	12
(I)	14	14	9	135	130	1.00	25	31

Custos > Indiretos: 360 Diretos: 748 Projeto: 1108 Prazo: 45

Caminhos	Durações	Folgas
ADG	45	0 ==> Crítico
BEG	45	0 ==> Crítico
BH	24	21
CEG	45	0 ==> Crítico
CFI	39	6
CH	24	21

 Acelerando Conjunto { G } em 6 unidades de tempo - (Iteração: 2)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	12	11	31	30	1.00	0	0
(D)	12	12	7	71	51	4.00	11	11
(E)	11	11	5	159	105	9.00	12	12
(F)	13	13	9	210	170	10.00	12	12
(G)	22	16	13	135	120	5.00	23	23
(H)	12	12	7	135	100	7.00	12	12
(I)	14	14	9	135	130	1.00	25	25

Custos > Indiretos: 312 Diretos: 778 Projeto: 1090 Prazo: 39

Caminhos	Durações	Folgas
ADG	39	0 ==> Crítico
BEG	39	0 ==> Crítico
BH	24	15
CEG	39	0 ==> Crítico
CFI	39	0 ==> Crítico
CH	24	15

Acelerando Conjunto { G I } em 3 unidades de tempo - (Iteração: 3)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30	2.00	0	0
(B)	12	12	5	63	42	3.00	0	0
(C)	19	12	11	31	30	1.00	0	0
(D)	12	12	7	71	51	4.00	11	11
(E)	11	11	5	159	105	9.00	12	12
(F)	13	13	9	210	170	10.00	12	12
(G)	22	13#	13	135	135	5.00	23	23
(H)	12	12	7	135	100	7.00	12	12
(I)	14	11	9	135	133	1.00	25	25

Custos > Indiretos: 288 Diretos: 796 Projeto: 1084 Prazo: 36

Caminhos	Durações	Folgas
ADG	36	0 ==> Crítico
BEG	36	0 ==> Crítico
BH	24	12
CEG	36	0 ==> Crítico
CFI	36	0 ==> Crítico
CH	24	12

Acelerando Conjunto { A B C } em 1 unidade de tempo - (Iteração: 4)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	10	6	40	32	2.00	0	0
(B)	12	11	5	63	45	3.00	0	0
(C)	19	11#	11	31	31	1.00	0	0
(D)	12	12	7	71	51	4.00	10	10
(E)	11	11	5	159	105	9.00	11	11
(F)	13	13	9	210	170	10.00	11	11
(G)	22	13#	13	135	135	5.00	22	22
(H)	12	12	7	135	100	7.00	11	11
(I)	14	11	9	135	133	1.00	24	24

Custos > Indiretos: 280 Diretos: 802 Projeto: 1082 Prazo: 35

Caminhos	Durações	Folgas
ADG	35	0 ==> Crítico
BEG	35	0 ==> Crítico
BH	23	12
CEG	35	0 ==> Crítico
CFI	35	0 ==> Crítico
CH	23	12

Acelerando Conjunto { A E I } em 2 unidades de tempo - (Iteração: 5)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	36	2.00	0	0
(B)	12	11	5	63	45	3.00	0	0
(C)	19	11#	11	31	31	1.00	0	0
(D)	12	12	7	71	51	4.00	8	8
(E)	11	9	5	159	123	9.00	11	11
(F)	13	13	9	210	170	10.00	11	11
(G)	22	13#	13	135	135	5.00	20	20
(H)	12	12	7	135	100	7.00	11	11
(I)	14	9#	9	135	135	1.00	24	24

Custos > Indiretos: 264 Diretos: 826 Projeto: 1090 Prazo: 33

Caminhos	Durações	Folgas
ADG	33	0 ==> Crítico
BEG	33	0 ==> Crítico
BH	23	10
CEG	33	0 ==> Crítico
CFI	33	0 ==> Crítico
CH	23	10

Acelerando Conjunto { A E F } em 2 unidades de tempo - (Iteração: 6)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	6#	6	40	40	2.00	0	0
(B)	12	11	5	63	45	3.00	0	0
(C)	19	11#	11	31	31	1.00	0	0
(D)	12	12	7	71	51	4.00	6	6
(E)	11	7	5	159	141	9.00	11	11
(F)	13	11	9	210	190	10.00	11	11
(G)	22	13#	13	135	135	5.00	18	18
(H)	12	12	7	135	100	7.00	11	11
(I)	14	9#	9	135	135	1.00	22	22

Custos > Indiretos: 248 Diretos: 868 Projeto: 1116 Prazo: 31

Caminhos	Durações	Folgas	
ADG	31	0	==> Crítico
BEG	31	0	==> Crítico
BH	23	8	
CEG	31	0	==> Crítico
CFI	31	0	==> Crítico
CH	23	8	

Acelerando Conjunto { D E F } em 2 unidades de tempo - (Iteração: 7)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	6#	6	40	40	2.00	0	0
(B)	12	11	5	63	45	3.00	0	0
(C)	19	11#	11	31	31	1.00	0	0
(D)	12	10	7	71	59	4.00	6	6
(E)	11	5#	5	159	159	9.00	11	11
(F)	13	9#	9	210	210	10.00	11	11
(G)	22	13#	13	135	135	5.00	16	16
(H)	12	12	7	135	100	7.00	11	11
(I)	14	9#	9	135	135	1.00	20	20

Custos > Indiretos: 232 Diretos: 914 Projeto: 1146 Prazo: 29

Caminhos	Durações	Folgas	
ADG	29	0	==> Crítico
BEG	29	0	==> Crítico
BH	23	6	
CEG	29	0	==> Crítico
CFI	29	0	==> Crítico
CH	23	6	

APÊNDICE D – Relatório Estudo de Caso 2 Versão Linear

Legendas: # => Atividade totalmente acelerada
 Custo Fixo = 20 | Prazo = 0
 Critério de parada escolhido: Prazo
 Aceleração Total = 8314
 Versão = Linear

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	12	3	198	135	7.00	10	10
(F)	21	21	6	430	215	14.33	22	22
(G)	8	8	5	96	90	2.00	22	22
(H)	11	11	5	132	120	2.00	30	46
(I)	5	5	1	290	190	25.00	41	57
(J)	7	7	3	84	80	1.00	30	46
(K)	8	8	3	280	180	20.00	43	43
(L)	11	11	1	420	320	10.00	51	51
(M)	24	24	4	1240	620	31.00	62	62
(N)	23	23	8	380	260	8.00	86	86
(O)	13	13	3	570	410	16.00	46	62
(P)	14	14	9	260	180	16.00	109	109
(Q)	17	17	7	1020	900	12.00	109	109
(R)	13	13	4	380	200	20.00	123	123
(S)	9	9	6	270	210	20.00	126	146
(T)	19	19	10	522	432	10.00	136	136
(U)	5	5	1	200	160	10.00	155	155
(V)	12	12	2	350	250	10.00	160	160
(X)	11	11	3	424	336	11.00	172	172

Custos > Indiretos: 3660 Diretos: 5803 Projeto: 9463 Prazo: 183

Caminhos	Durações	Folgas
ABHIMNPRTUVX	151	32
ABHIMNQSUVX	131	52
ABHIOVX	66	117
ACGHIMNPRTUVX	165	18
ACGHIMNQSUVX	145	38
ACGHIOVX	80	103
ACGJMNPTUVX	156	27
ACGJMNSUVX	136	47
ADGHIMNPRTUVX	160	23
ADGHIMNQSUVX	140	43
ADGHIOVX	75	108
ADGJMNPTUVX	151	32
ADGJMNSUVX	131	52
AEFKLMNPRTUVX	183	0 ==> Crítico
AEFKLMNQSUVX	163	20
AEGHIMNPRTUVX	167	16
AEGHIMNQSUVX	147	36
AEGHIOVX	82	101
AEGJMNPTUVX	158	25
AEGJMNSUVX	138	45

 Acelerando Conjunto { E } em 9 unidades de tempo - (Iteração: 1)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	11	5	132	120	2.00	28	37
(I)	5	5	1	290	190	25.00	39	48
(J)	7	7	3	84	80	1.00	28	37
(K)	8	8	3	280	180	20.00	34	34
(L)	11	11	1	420	320	10.00	42	42
(M)	24	24	4	1240	620	31.00	53	53
(N)	23	23	8	380	260	8.00	77	77
(O)	13	13	3	570	410	16.00	44	53

(P)	14	14	9	260	180	16.00	100	100
(Q)	17	17	7	1020	900	12.00	100	100
(R)	13	13	4	380	200	20.00	114	114
(S)	9	9	6	270	210	20.00	117	137
(T)	19	19	10	522	432	10.00	127	127
(U)	5	5	1	200	160	10.00	146	146
(V)	12	12	2	350	250	10.00	151	151
(X)	11	11	3	424	336	11.00	163	163

Custos > Indiretos: 3480 Diretos: 5866 Projeto: 9346 Prazo: 174

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	151	23	
ABHIMNQSUVX	131	43	
ABHIOVX	66	108	
ACGHIMNPRTUVX	165	9	
ACGHIMNQSUVX	145	29	
ACGHIOVX	80	94	
ACGJMNPTUVX	156	18	
ACGJMNSUVX	136	38	
ADGHIMNPRTUVX	160	14	
ADGHIMNQSUVX	140	34	
ADGHIOVX	75	99	
ADGJMNPRTUVX	151	23	
ADGJMNSUVX	131	43	
AEFKLMNPRTUVX	174	0	==> Crítico
AEFKLMNQSUVX	154	20	
AEGHIMNPRTUVX	158	16	
AEGHIMNQSUVX	138	36	
AEGHIOVX	73	101	
AEGJMNPRTUVX	149	25	
AEGJMNSUVX	129	45	

 Acelerando Conjunto { N } em 15 unidades de tempo - (Iteração: 2)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	11	5	132	120	2.00	28	37
(I)	5	5	1	290	190	25.00	39	48
(J)	7	7	3	84	80	1.00	28	37
(K)	8	8	3	280	180	20.00	34	34
(L)	11	11	1	420	320	10.00	42	42
(M)	24	24	4	1240	620	31.00	53	53
(N)	23	8#	8	380	380	8.00	77	77
(O)	13	13	3	570	410	16.00	44	53
(P)	14	14	9	260	180	16.00	85	85
(Q)	17	17	7	1020	900	12.00	85	85
(R)	13	13	4	380	200	20.00	99	99
(S)	9	9	6	270	210	20.00	102	122
(T)	19	19	10	522	432	10.00	112	112
(U)	5	5	1	200	160	10.00	131	131
(V)	12	12	2	350	250	10.00	136	136
(X)	11	11	3	424	336	11.00	148	148

Custos > Indiretos: 3180 Diretos: 5986 Projeto: 9166 Prazo: 159

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	136	23	
ABHIMNQSUVX	116	43	
ABHIOVX	66	93	
ACGHIMNPRTUVX	150	9	
ACGHIMNQSUVX	130	29	
ACGHIOVX	80	79	
ACGJMNPRTUVX	141	18	
ACGJMNSUVX	121	38	
ADGHIMNPRTUVX	145	14	
ADGHIMNQSUVX	125	34	
ADGHIOVX	75	84	
ADGJMNPRTUVX	136	23	
ADGJMNSUVX	116	43	
AEFKLMNPRTUVX	159	0	==> Crítico
AEFKLMNQSUVX	139	20	
AEGHIMNPRTUVX	143	16	

AEGHIMNQSUVX	123	36
AEGHIOVX	73	86
AEGJMNPRUVX	134	25
AEGJMNQSUVX	114	45

Acelerando Conjunto { V } em 10 unidades de tempo - (Iteração: 3)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	11	5	132	120	2.00	28	37
(I)	5	5	1	290	190	25.00	39	48
(J)	7	7	3	84	80	1.00	28	37
(K)	8	8	3	280	180	20.00	34	34
(L)	11	11	1	420	320	10.00	42	42
(M)	24	24	4	1240	620	31.00	53	53
(N)	23	8#	8	380	380	8.00	77	77
(O)	13	13	3	570	410	16.00	44	53
(P)	14	14	9	260	180	16.00	85	85
(Q)	17	17	7	1020	900	12.00	85	85
(R)	13	13	4	380	200	20.00	99	99
(S)	9	9	6	270	210	20.00	102	122
(T)	19	19	10	522	432	10.00	112	112
(U)	5	5	1	200	160	10.00	131	131
(V)	12	2#	2	350	350	10.00	136	136
(X)	11	11	3	424	336	11.00	138	138

Custos > Indiretos: 2980 Diretos: 6086 Projeto: 9066 Prazo: 149

Caminhos	Durações	Folgas
ABHIMNPRTUVX	126	23
ABHIMNQSUVX	106	43
ABHIOVX	56	93
ACGHIMNPRTUVX	140	9
ACGHIMNQSUVX	120	29
ACGHIOVX	70	79
ACGJMNPRUVX	131	18
ACGJMNQSUVX	111	38
ADGHIMNPRTUVX	135	14
ADGHIMNQSUVX	115	34
ADGHIOVX	65	84
ADGJMNPRUVX	126	23
ADGJMNQSUVX	106	43
AEFKLMNPRTUVX	149	0 ==> Crítico
AEFKLMNQSUVX	129	20
AEGHIMNPRTUVX	133	16
AEGHIMNQSUVX	113	36
AEGHIOVX	63	86
AEGJMNPRUVX	124	25
AEGJMNQSUVX	104	45

Acelerando Conjunto { T } em 9 unidades de tempo - (Iteração: 4)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	11	5	132	120	2.00	28	37
(I)	5	5	1	290	190	25.00	39	48
(J)	7	7	3	84	80	1.00	28	37
(K)	8	8	3	280	180	20.00	34	34
(L)	11	11	1	420	320	10.00	42	42
(M)	24	24	4	1240	620	31.00	53	53
(N)	23	8#	8	380	380	8.00	77	77
(O)	13	13	3	570	410	16.00	44	53
(P)	14	14	9	260	180	16.00	85	85
(Q)	17	17	7	1020	900	12.00	85	85
(R)	13	13	4	380	200	20.00	99	99
(S)	9	9	6	270	210	20.00	102	113

(T)	19	10#	10	522	522	10.00	112	112
(U)	5	5	1	200	160	10.00	122	122
(V)	12	2#	2	350	350	10.00	127	127
(X)	11	11	3	424	336	11.00	129	129

Custos > Indiretos: 2800 Diretos: 6176 Projeto: 8976 Prazo: 140

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	117	23	
ABHIMNQSUVX	106	34	
ABHIOVX	56	84	
ACGHIMNPRTUVX	131	9	
ACGHIMNQSUVX	120	20	
ACGHIOVX	70	70	
ACGJMNPTUVX	122	18	
ACGJMNSUVX	111	29	
ADGHIMNPRTUVX	126	14	
ADGHIMNQSUVX	115	25	
ADGHIOVX	65	75	
ADGJMNPRTUVX	117	23	
ADGJMNSUVX	106	34	
AEFKLMNPRTUVX	140	0	==> Crítico
AEFKLMNQSUVX	129	11	
AEGHIMNPRTUVX	124	16	
AEGHIMNQSUVX	113	27	
AEGHIOVX	63	77	
AEGJMNPRTUVX	115	25	
AEGJMNSUVX	104	36	

 Acelerando Conjunto { L } em 9 unidades de tempo - (Iteração: 5)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	11	5	132	120	2.00	28	28
(I)	5	5	1	290	190	25.00	39	39
(J)	7	7	3	84	80	1.00	28	28
(K)	8	8	3	280	180	20.00	34	34
(L)	11	2	1	420	410	10.00	42	42
(M)	24	24	4	1240	620	31.00	44	44
(N)	23	8#	8	380	380	8.00	68	68
(O)	13	13	3	570	410	16.00	44	44
(P)	14	14	9	260	180	16.00	76	76
(Q)	17	17	7	1020	900	12.00	76	76
(R)	13	13	4	380	200	20.00	90	90
(S)	9	9	6	270	210	20.00	93	104
(T)	19	10#	10	522	522	10.00	103	103
(U)	5	5	1	200	160	10.00	113	113
(V)	12	2#	2	350	350	10.00	118	118
(X)	11	11	3	424	336	11.00	120	120

Custos > Indiretos: 2620 Diretos: 6266 Projeto: 8886 Prazo: 131

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	117	14	
ABHIMNQSUVX	106	25	
ABHIOVX	56	75	
ACGHIMNPRTUVX	131	0	==> Crítico
ACGHIMNQSUVX	120	11	
ACGHIOVX	70	61	
ACGJMNPRTUVX	122	9	
ACGJMNSUVX	111	20	
ADGHIMNPRTUVX	126	5	
ADGHIMNQSUVX	115	16	
ADGHIOVX	65	66	
ADGJMNPRTUVX	117	14	
ADGJMNSUVX	106	25	
AEFKLMNPRTUVX	131	0	==> Crítico
AEFKLMNQSUVX	120	11	
AEGHIMNPRTUVX	124	7	
AEGHIMNQSUVX	113	18	
AEGHIOVX	63	68	
AEGJMNPRTUVX	115	16	
AEGJMNSUVX	104	27	

 Acelerando Conjunto { U } em 4 unidades de tempo - (Iteração: 6)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	11	5	132	120	2.00	28	28
(I)	5	5	1	290	190	25.00	39	39
(J)	7	7	3	84	80	1.00	28	28
(K)	8	8	3	280	180	20.00	34	34
(L)	11	2	1	420	410	10.00	42	42
(M)	24	24	4	1240	620	31.00	44	44
(N)	23	8#	8	380	380	8.00	68	68
(O)	13	13	3	570	410	16.00	44	44
(P)	14	14	9	260	180	16.00	76	76
(Q)	17	17	7	1020	900	12.00	76	76
(R)	13	13	4	380	200	20.00	90	90
(S)	9	9	6	270	210	20.00	93	104
(T)	19	10#	10	522	522	10.00	103	103
(U)	5	1#	1	200	200	10.00	113	113
(V)	12	2#	2	350	350	10.00	114	114
(X)	11	11	3	424	336	11.00	116	116

Custos > Indiretos: 2540 Diretos: 6306 Projeto: 8846 Prazo: 127

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	113	14	
ABHIMNQSUVX	102	25	
ABHIOVX	56	71	
ACGHIMNPRTUVX	127	0	==> Crítico
ACGHIMNQSUVX	116	11	
ACGHIOVX	70	57	
ACGJMNPTUVX	118	9	
ACGJMNSUVX	107	20	
ADGHIMNPRTUVX	122	5	
ADGHIMNQSUVX	111	16	
ADGHIOVX	65	62	
ADGJMNPTUVX	113	14	
ADGJMNSUVX	102	25	
AEFKLMNPRTUVX	127	0	==> Crítico
AEFKLMNQSUVX	116	11	
AEGHIMNPRTUVX	120	7	
AEGHIMNQSUVX	109	18	
AEGHIOVX	63	64	
AEGJMNPTUVX	111	16	
AEGJMNSUVX	100	27	

 Acelerando Conjunto { X } em 8 unidades de tempo - (Iteração: 7)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	11	5	132	120	2.00	28	28
(I)	5	5	1	290	190	25.00	39	39
(J)	7	7	3	84	80	1.00	28	28
(K)	8	8	3	280	180	20.00	34	34
(L)	11	2	1	420	410	10.00	42	42
(M)	24	24	4	1240	620	31.00	44	44
(N)	23	8#	8	380	380	8.00	68	68
(O)	13	13	3	570	410	16.00	44	44
(P)	14	14	9	260	180	16.00	76	76
(Q)	17	17	7	1020	900	12.00	76	76
(R)	13	13	4	380	200	20.00	90	90
(S)	9	9	6	270	210	20.00	93	104
(T)	19	10#	10	522	522	10.00	103	103
(U)	5	1#	1	200	200	10.00	113	113
(V)	12	2#	2	350	350	10.00	114	114
(X)	11	3#	3	424	424	11.00	116	116

Custos > Indiretos: 2380 Diretos: 6394 Projeto: 8774 Prazo: 119

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	105	14	
ABHIMNQSUVX	94	25	
ABHIOVX	48	71	
ACGHIMNPRTUVX	119	0	==> Crítico
ACGHIMNQSUVX	108	11	
ACGHIOVX	62	57	
ACGJMNPTUVX	110	9	
ACGJMNSUVX	99	20	
ADGHIMNPRTUVX	114	5	
ADGHIMNQSUVX	103	16	
ADGHIOVX	57	62	
ADGJMNPTUVX	105	14	
ADGJMNSUVX	94	25	
AEFKLMNPRTUVX	119	0	==> Crítico
AEFKLMNQSUVX	108	11	
AEGHIMNPRTUVX	112	7	
AEGHIMNQSUVX	101	18	
AEGHIOVX	55	64	
AEGJMNPTUVX	103	16	
AEGJMNSUVX	92	27	

 Acelerando Conjunto { H L } em 1 unidade de tempo - (Iteração: 8)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	10	5	132	122	2.00	28	28
(I)	5	5	1	290	190	25.00	38	38
(J)	7	7	3	84	80	1.00	28	28
(K)	8	8	3	280	180	20.00	34	34
(L)	11	1#	1	420	420	10.00	42	42
(M)	24	24	4	1240	620	31.00	43	43
(N)	23	8#	8	380	380	8.00	67	67
(O)	13	13	3	570	410	16.00	43	43
(P)	14	14	9	260	180	16.00	75	75
(Q)	17	17	7	1020	900	12.00	75	75
(R)	13	13	4	380	200	20.00	89	89
(S)	9	9	6	270	210	20.00	92	103
(T)	19	10#	10	522	522	10.00	102	102
(U)	5	1#	1	200	200	10.00	112	112
(V)	12	2#	2	350	350	10.00	113	113
(X)	11	3#	3	424	424	11.00	115	115

Custos > Indiretos: 2360 Diretos: 6406 Projeto: 8766 Prazo: 118

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	104	14	
ABHIMNQSUVX	93	25	
ABHIOVX	47	71	
ACGHIMNPRTUVX	118	0	==> Crítico
ACGHIMNQSUVX	107	11	
ACGHIOVX	61	57	
ACGJMNPTUVX	110	8	
ACGJMNSUVX	99	19	
ADGHIMNPRTUVX	113	5	
ADGHIMNQSUVX	102	16	
ADGHIOVX	56	62	
ADGJMNPTUVX	105	13	
ADGJMNSUVX	94	24	
AEFKLMNPRTUVX	118	0	==> Crítico
AEFKLMNQSUVX	107	11	
AEGHIMNPRTUVX	111	7	
AEGHIMNQSUVX	100	18	
AEGHIOVX	54	64	
AEGJMNPTUVX	103	15	
AEGJMNSUVX	92	26	

Acelerando Conjunto { P } em 5 unidades de tempo - (Iteração: 9)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	21	6	430	215	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	10	5	132	122	2.00	28	28
(I)	5	5	1	290	190	25.00	38	38
(J)	7	7	3	84	80	1.00	28	28
(K)	8	8	3	280	180	20.00	34	34
(L)	11	1#	1	420	420	10.00	42	42
(M)	24	24	4	1240	620	31.00	43	43
(N)	23	8#	8	380	380	8.00	67	67
(O)	13	13	3	570	410	16.00	43	43
(P)	14	9#	9	260	260	16.00	75	75
(Q)	17	17	7	1020	900	12.00	75	75
(R)	13	13	4	380	200	20.00	84	84
(S)	9	9	6	270	210	20.00	92	98
(T)	19	10#	10	522	522	10.00	97	97
(U)	5	1#	1	200	200	10.00	107	107
(V)	12	2#	2	350	350	10.00	108	108
(X)	11	3#	3	424	424	11.00	110	110

Custos > Indiretos: 2260 Diretos: 6486 Projeto: 8746 Prazo: 113

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	99	14	
ABHIMNQSUVX	93	20	
ABHIOVX	47	66	
ACGHIMNPRTUVX	113	0	==> Crítico
ACGHIMNQSUVX	107	6	
ACGHIOVX	61	52	
ACGJMNPTUVX	105	8	
ACGJMNSUVX	99	14	
ADGHIMNPRTUVX	108	5	
ADGHIMNQSUVX	102	11	
ADGHIOVX	56	57	
ADGJMNPTUVX	100	13	
ADGJMNSUVX	94	19	
AEFKLMNPRTUVX	113	0	==> Crítico
AEFKLMNQSUVX	107	6	
AEGHIMNPRTUVX	106	7	
AEGHIMNQSUVX	100	13	
AEGHIOVX	54	59	
AEGJMNPTUVX	98	15	
AEGJMNSUVX	92	21	

Acelerando Conjunto { F H } em 5 unidades de tempo - (Iteração: 10)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	16	6	430	287	14.33	13	13
(G)	8	8	5	96	90	2.00	20	13
(H)	11	5#	5	132	132	2.00	28	28
(I)	5	5	1	290	190	25.00	33	33
(J)	7	7	3	84	80	1.00	28	28
(K)	8	8	3	280	180	20.00	29	29
(L)	11	1#	1	420	420	10.00	37	37
(M)	24	24	4	1240	620	31.00	38	38
(N)	23	8#	8	380	380	8.00	62	62
(O)	13	13	3	570	410	16.00	38	38
(P)	14	9#	9	260	260	16.00	70	70
(Q)	17	17	7	1020	900	12.00	70	70
(R)	13	13	4	380	200	20.00	79	79
(S)	9	9	6	270	210	20.00	87	93
(T)	19	10#	10	522	522	10.00	92	92
(U)	5	1#	1	200	200	10.00	102	102
(V)	12	2#	2	350	350	10.00	103	103
(X)	11	3#	3	424	424	11.00	105	105

Custos > Indiretos: 2160 Diretos: 6568 Projeto: 8728 Prazo: 108

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	94	14	
ABHIMNQSUVX	88	20	
ABHIOVX	42	66	
ACGHIMNPRTUVX	108	0	==> Crítico
ACGHIMNQSUVX	102	6	
ACGHIOVX	56	52	
ACGJMNPTUVX	105	3	
ACGJMNSUVX	99	9	
ADGHIMNPRTUVX	103	5	
ADGHIMNQSUVX	97	11	
ADGHIOVX	51	57	
ADGJMNPRTUVX	100	8	
ADGJMNSUVX	94	14	
AEFKLMNPRTUVX	108	0	==> Crítico
AEFKLMNQSUVX	102	6	
AEGHIMNPRTUVX	101	7	
AEGHIMNQSUVX	95	13	
AEGHIOVX	49	59	
AEGJMNPRTUVX	98	10	
AEGJMNSUVX	92	16	

Acelerando Conjunto { F G } em 3 unidades de tempo - (Iteração: 11)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	13	6	430	330	14.33	13	13
(G)	8	5#	5	96	96	2.00	20	13
(H)	11	5#	5	132	132	2.00	25	25
(I)	5	5	1	290	190	25.00	30	30
(J)	7	7	3	84	80	1.00	25	25
(K)	8	8	3	280	180	20.00	26	26
(L)	11	1#	1	420	420	10.00	34	34
(M)	24	24	4	1240	620	31.00	35	35
(N)	23	8#	8	380	380	8.00	59	59
(O)	13	13	3	570	410	16.00	35	35
(P)	14	9#	9	260	260	16.00	67	67
(Q)	17	17	7	1020	900	12.00	67	67
(R)	13	13	4	380	200	20.00	76	76
(S)	9	9	6	270	210	20.00	84	90
(T)	19	10#	10	522	522	10.00	89	89
(U)	5	1#	1	200	200	10.00	99	99
(V)	12	2#	2	350	350	10.00	100	100
(X)	11	3#	3	424	424	11.00	102	102

Custos > Indiretos: 2100 Diretos: 6617 Projeto: 8717 Prazo: 105

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	94	11	
ABHIMNQSUVX	88	17	
ABHIOVX	42	63	
ACGHIMNPRTUVX	105	0	==> Crítico
ACGHIMNQSUVX	99	6	
ACGHIOVX	53	52	
ACGJMNPRTUVX	102	3	
ACGJMNSUVX	96	9	
ADGHIMNPRTUVX	100	5	
ADGHIMNQSUVX	94	11	
ADGHIOVX	48	57	
ADGJMNPRTUVX	97	8	
ADGJMNSUVX	91	14	
AEFKLMNPRTUVX	105	0	==> Crítico
AEFKLMNQSUVX	99	6	
AEGHIMNPRTUVX	98	7	
AEGHIMNQSUVX	92	13	
AEGHIOVX	46	59	
AEGJMNPRTUVX	95	10	
AEGJMNSUVX	89	16	

Acelerando Conjunto { R } em 6 unidades de tempo - (Iteração: 12)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160	20.00	0	0
(B)	4	4	1	146	125	7.00	10	10
(C)	10	10	2	250	170	10.00	10	10
(D)	5	5	2	72	60	4.00	10	10
(E)	12	3#	3	198	198	7.00	10	10
(F)	21	13	6	430	330	14.33	13	13
(G)	8	5#	5	96	96	2.00	20	13
(H)	11	5#	5	132	132	2.00	25	25
(I)	5	5	1	290	190	25.00	30	30
(J)	7	7	3	84	80	1.00	25	25
(K)	8	8	3	280	180	20.00	26	26
(L)	11	1#	1	420	420	10.00	34	34
(M)	24	24	4	1240	620	31.00	35	35
(N)	23	8#	8	380	380	8.00	59	59
(O)	13	13	3	570	410	16.00	35	35
(P)	14	9#	9	260	260	16.00	67	67
(Q)	17	17	7	1020	900	12.00	67	67
(R)	13	7	4	380	320	20.00	76	76
(S)	9	9	6	270	210	20.00	84	84
(T)	19	10#	10	522	522	10.00	83	83
(U)	5	1#	1	200	200	10.00	93	93
(V)	12	2#	2	350	350	10.00	94	94
(X)	11	3#	3	424	424	11.00	96	96

Custos > Indiretos: 1980 Diretos: 6737 Projeto: 8717 Prazo: 99

Caminhos	Durações	Folgas
ABHIMNPRTUVX	88	11
ABHIMNQSUVX	88	11
ABHIOVX	42	57
ACGHIMNPRTUVX	99	0 ==> Crítico
ACGHIMNQSUVX	99	0 ==> Crítico
ACGHIOVX	53	46
ACGJMNPTUVX	96	3
ACGJMNSUVX	96	3
ADGHIMNPRTUVX	94	5
ADGHIMNQSUVX	94	5
ADGHIOVX	48	51
ADGJMNPTUVX	91	8
ADGJMNSUVX	91	8
AEFKLMNPRTUVX	99	0 ==> Crítico
AEFKLMNQSUVX	99	0 ==> Crítico
AEGHIMNPRTUVX	92	7
AEGHIMNQSUVX	92	7
AEGHIOVX	46	53
AEGJMNPTUVX	89	10
AEGJMNSUVX	89	10

Acelerando Conjunto { A } em 7 unidades de tempo - (Iteração: 13)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300	20.00	0	0
(B)	4	4	1	146	125	7.00	3	3
(C)	10	10	2	250	170	10.00	3	3
(D)	5	5	2	72	60	4.00	3	3
(E)	12	3#	3	198	198	7.00	3	3
(F)	21	13	6	430	330	14.33	6	6
(G)	8	5#	5	96	96	2.00	13	6
(H)	11	5#	5	132	132	2.00	18	18
(I)	5	5	1	290	190	25.00	23	23
(J)	7	7	3	84	80	1.00	18	18
(K)	8	8	3	280	180	20.00	19	19
(L)	11	1#	1	420	420	10.00	27	27
(M)	24	24	4	1240	620	31.00	28	28
(N)	23	8#	8	380	380	8.00	52	52
(O)	13	13	3	570	410	16.00	28	28
(P)	14	9#	9	260	260	16.00	60	60
(Q)	17	17	7	1020	900	12.00	60	60
(R)	13	7	4	380	320	20.00	69	69
(S)	9	9	6	270	210	20.00	77	77
(T)	19	10#	10	522	522	10.00	76	76
(U)	5	1#	1	200	200	10.00	86	86
(V)	12	2#	2	350	350	10.00	87	87
(X)	11	3#	3	424	424	11.00	89	89

Custos > Indiretos: 1840 Diretos: 6877 Projeto: 8717 Prazo: 92

Caminhos	Durações	Folgas
ABHIMNPRTUVX	81	11
ABHIMNQSUVX	81	11
ABHIOVX	35	57
ACGHIMNPRTUVX	92	0 ==> Crítico
ACGHIMNQSUVX	92	0 ==> Crítico
ACGHIOVX	46	46
ACGJMNPTUVX	89	3
ACGJMNSUVX	89	3
ADGHIMNPRTUVX	87	5
ADGHIMNQSUVX	87	5
ADGHIOVX	41	51
ADGJMNPRTUVX	84	8
ADGJMNSUVX	84	8
AEFKLMNPRTUVX	92	0 ==> Crítico
AEFKLMNQSUVX	92	0 ==> Crítico
AEGHIMNPRTUVX	85	7
AEGHIMNQSUVX	85	7
AEGHIOVX	39	53
AEGJMNPRTUVX	82	10
AEGJMNSUVX	82	10

Acelerando Conjunto { C F } em 5 unidades de tempo - (Iteração: 14)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300	20.00	0	0
(B)	4	4	1	146	125	7.00	3	3
(C)	10	5	2	250	220	10.00	3	3
(D)	5	5	2	72	60	4.00	3	3
(E)	12	3#	3	198	198	7.00	3	3
(F)	21	8	6	430	402	14.33	6	6
(G)	8	5#	5	96	96	2.00	8	6
(H)	11	5#	5	132	132	2.00	13	13
(I)	5	5	1	290	190	25.00	18	18
(J)	7	7	3	84	80	1.00	13	13
(K)	8	8	3	280	180	20.00	14	14
(L)	11	1#	1	420	420	10.00	22	22
(M)	24	24	4	1240	620	31.00	23	23
(N)	23	8#	8	380	380	8.00	47	47
(O)	13	13	3	570	410	16.00	23	23
(P)	14	9#	9	260	260	16.00	55	55
(Q)	17	17	7	1020	900	12.00	55	55
(R)	13	7	4	380	320	20.00	64	64
(S)	9	9	6	270	210	20.00	72	72
(T)	19	10#	10	522	522	10.00	71	71
(U)	5	1#	1	200	200	10.00	81	81
(V)	12	2#	2	350	350	10.00	82	82
(X)	11	3#	3	424	424	11.00	84	84

Custos > Indiretos: 1740 Diretos: 6999 Projeto: 8739 Prazo: 87

Caminhos	Durações	Folgas
ABHIMNPRTUVX	81	6
ABHIMNQSUVX	81	6
ABHIOVX	35	52
ACGHIMNPRTUVX	87	0 ==> Crítico
ACGHIMNQSUVX	87	0 ==> Crítico
ACGHIOVX	41	46
ACGJMNPRTUVX	84	3
ACGJMNSUVX	84	3
ADGHIMNPRTUVX	87	0 ==> Crítico
ADGHIMNQSUVX	87	0 ==> Crítico
ADGHIOVX	41	46
ADGJMNPRTUVX	84	3
ADGJMNSUVX	84	3
AEFKLMNPRTUVX	87	0 ==> Crítico
AEFKLMNQSUVX	87	0 ==> Crítico
AEGHIMNPRTUVX	85	2
AEGHIMNQSUVX	85	2
AEGHIOVX	39	48
AEGJMNPRTUVX	82	5
AEGJMNSUVX	82	5

Acelerando Conjunto { C D F } em 2 unidades de tempo - (Iteração: 15)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300	20.00	0	0
(B)	4	4	1	146	125	7.00	3	3
(C)	10	3	2	250	240	10.00	3	3
(D)	5	3	2	72	68	4.00	3	3
(E)	12	3#	3	198	198	7.00	3	3
(F)	21	6#	6	430	430	14.33	6	6
(G)	8	5#	5	96	96	2.00	6	6
(H)	11	5#	5	132	132	2.00	11	11
(I)	5	5	1	290	190	25.00	16	16
(J)	7	7	3	84	80	1.00	11	11
(K)	8	8	3	280	180	20.00	12	12
(L)	11	1#	1	420	420	10.00	20	20
(M)	24	24	4	1240	620	31.00	21	21
(N)	23	8#	8	380	380	8.00	45	45
(O)	13	13	3	570	410	16.00	21	21
(P)	14	9#	9	260	260	16.00	53	53
(Q)	17	17	7	1020	900	12.00	53	53
(R)	13	7	4	380	320	20.00	62	62
(S)	9	9	6	270	210	20.00	70	70
(T)	19	10#	10	522	522	10.00	69	69
(U)	5	1#	1	200	200	10.00	79	79
(V)	12	2#	2	350	350	10.00	80	80
(X)	11	3#	3	424	424	11.00	82	82

Custos > Indiretos: 1700 Diretos: 7055 Projeto: 8755 Prazo: 85

Caminhos	Durações	Folgas
ABHIMNPRTUVX	81	4
ABHIMNQSUVX	81	4
ABHIOVX	35	50
ACGHIMNPRTUVX	85	0 ==> Crítico
ACGHIMNQSUVX	85	0 ==> Crítico
ACGHIOVX	39	46
ACGJMNPRTUVX	82	3
ACGJMNQSUVX	82	3
ADGHIMNPRTUVX	85	0 ==> Crítico
ADGHIMNQSUVX	85	0 ==> Crítico
ADGHIOVX	39	46
ADGJMNPRTUVX	82	3
ADGJMNQSUVX	82	3
AEFKLMNPRTUVX	85	0 ==> Crítico
AEFKLMNQSUVX	85	0 ==> Crítico
AEGHIMNPRTUVX	85	0 ==> Crítico
AEGHIMNQSUVX	85	0 ==> Crítico
AEGHIOVX	39	46
AEGJMNPRTUVX	82	3
AEGJMNQSUVX	82	3

Acelerando Conjunto { M } em 20 unidades de tempo - (Iteração: 16)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300	20.00	0	0
(B)	4	4	1	146	125	7.00	3	3
(C)	10	3	2	250	240	10.00	3	3
(D)	5	3	2	72	68	4.00	3	3
(E)	12	3#	3	198	198	7.00	3	3
(F)	21	6#	6	430	430	14.33	6	6
(G)	8	5#	5	96	96	2.00	6	6
(H)	11	5#	5	132	132	2.00	11	11
(I)	5	5	1	290	190	25.00	16	16
(J)	7	7	3	84	80	1.00	11	11
(K)	8	8	3	280	180	20.00	12	12
(L)	11	1#	1	420	420	10.00	20	20
(M)	24	4#	4	1240	1240	31.00	21	21
(N)	23	8#	8	380	380	8.00	25	25
(O)	13	13	3	570	410	16.00	21	21
(P)	14	9#	9	260	260	16.00	33	33
(Q)	17	17	7	1020	900	12.00	33	33
(R)	13	7	4	380	320	20.00	42	42
(S)	9	9	6	270	210	20.00	50	50
(T)	19	10#	10	522	522	10.00	49	49
(U)	5	1#	1	200	200	10.00	59	59
(V)	12	2#	2	350	350	10.00	60	60
(X)	11	3#	3	424	424	11.00	62	62

Custos > Indiretos: 1300 Diretos: 7675 Projeto: 8975 Prazo: 65

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	61	4	
ABHIMNQSUVX	61	4	
ABHIOVX	35	30	
ACGHIMNPRTUVX	65	0	==> Crítico
ACGHIMNQSUVX	65	0	==> Crítico
ACGHIOVX	39	26	
ACGJMNPTUVX	62	3	
ACGJMNSUVX	62	3	
ADGHIMNPRTUVX	65	0	==> Crítico
ADGHIMNQSUVX	65	0	==> Crítico
ADGHIOVX	39	26	
ADGJMNPRTUVX	62	3	
ADGJMNSUVX	62	3	
AEFKLMNPRTUVX	65	0	==> Crítico
AEFKLMNQSUVX	65	0	==> Crítico
AEGHIMNPRTUVX	65	0	==> Crítico
AEGHIMNQSUVX	65	0	==> Crítico
AEGHIOVX	39	26	
AEGJMNPRTUVX	62	3	
AEGJMNSUVX	62	3	

Acelerando Conjunto { Q R } em 3 unidades de tempo - (Iteração: 17)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300	20.00	0	0
(B)	4	4	1	146	125	7.00	3	3
(C)	10	3	2	250	240	10.00	3	3
(D)	5	3	2	72	68	4.00	3	3
(E)	12	3#	3	198	198	7.00	3	3
(F)	21	6#	6	430	430	14.33	6	6
(G)	8	5#	5	96	96	2.00	6	6
(H)	11	5#	5	132	132	2.00	11	11
(I)	5	5	1	290	190	25.00	16	16
(J)	7	7	3	84	80	1.00	11	11
(K)	8	8	3	280	180	20.00	12	12
(L)	11	1#	1	420	420	10.00	20	20
(M)	24	4#	4	1240	1240	31.00	21	21
(N)	23	8#	8	380	380	8.00	25	25
(O)	13	13	3	570	410	16.00	21	21
(P)	14	9#	9	260	260	16.00	33	33
(Q)	17	14	7	1020	936	12.00	33	33
(R)	13	4#	4	380	380	20.00	42	42
(S)	9	9	6	270	210	20.00	47	47
(T)	19	10#	10	522	522	10.00	46	46
(U)	5	1#	1	200	200	10.00	56	56
(V)	12	2#	2	350	350	10.00	57	57
(X)	11	3#	3	424	424	11.00	59	59

Custos > Indiretos: 1240 Diretos: 7771 Projeto: 9011 Prazo: 62

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	58	4	
ABHIMNQSUVX	58	4	
ABHIOVX	35	27	
ACGHIMNPRTUVX	62	0	==> Crítico
ACGHIMNQSUVX	62	0	==> Crítico
ACGHIOVX	39	23	
ACGJMNPRTUVX	59	3	
ACGJMNSUVX	59	3	
ADGHIMNPRTUVX	62	0	==> Crítico
ADGHIMNQSUVX	62	0	==> Crítico
ADGHIOVX	39	23	
ADGJMNPRTUVX	59	3	
ADGJMNSUVX	59	3	
AEFKLMNPRTUVX	62	0	==> Crítico
AEFKLMNQSUVX	62	0	==> Crítico
AEGHIMNPRTUVX	62	0	==> Crítico
AEGHIMNQSUVX	62	0	==> Crítico
AEGHIOVX	39	23	
AEGJMNPRTUVX	59	3	
AEGJMNSUVX	59	3	

Acelerando Conjunto { I K } em 3 unidades de tempo - (Iteração: 18)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300	20.00	0	0
(B)	4	4	1	146	125	7.00	3	3
(C)	10	3	2	250	240	10.00	3	3
(D)	5	3	2	72	68	4.00	3	3
(E)	12	3#	3	198	198	7.00	3	3
(F)	21	6#	6	430	430	14.33	6	6
(G)	8	5#	5	96	96	2.00	6	6
(H)	11	5#	5	132	132	2.00	11	11
(I)	5	2	1	290	265	25.00	16	16
(J)	7	7	3	84	80	1.00	11	11
(K)	8	5	3	280	240	20.00	12	12
(L)	11	1#	1	420	420	10.00	17	17
(M)	24	4#	4	1240	1240	31.00	18	18
(N)	23	8#	8	380	380	8.00	22	22
(O)	13	13	3	570	410	16.00	18	18
(P)	14	9#	9	260	260	16.00	30	30
(Q)	17	14	7	1020	936	12.00	30	30
(R)	13	4#	4	380	380	20.00	39	39
(S)	9	9	6	270	210	20.00	44	44
(T)	19	10#	10	522	522	10.00	43	43
(U)	5	1#	1	200	200	10.00	53	53
(V)	12	2#	2	350	350	10.00	54	54
(X)	11	3#	3	424	424	11.00	56	56

Custos > Indiretos: 1180 Diretos: 7906 Projeto: 9086 Prazo: 59

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	55	4	
ABHIMNQSUVX	55	4	
ABHIOVX	32	27	
ACGHIMNPRTUVX	59	0	==> Crítico
ACGHIMNQSUVX	59	0	==> Crítico
ACGHIOVX	36	23	
ACGJMNPRTUVX	59	0	==> Crítico
ACGJMNQSUVX	59	0	==> Crítico
ADGHIMNPRTUVX	59	0	==> Crítico
ADGHIMNQSUVX	59	0	==> Crítico
ADGHIOVX	36	23	
ADGJMNPRTUVX	59	0	==> Crítico
ADGJMNQSUVX	59	0	==> Crítico
AEFKLMNPRTUVX	59	0	==> Crítico
AEFKLMNQSUVX	59	0	==> Crítico
AEGHIMNPRTUVX	59	0	==> Crítico
AEGHIMNQSUVX	59	0	==> Crítico
AEGHIOVX	36	23	
AEGJMNPRTUVX	59	0	==> Crítico
AEGJMNQSUVX	59	0	==> Crítico

 Acelerando Conjunto { I J K } em 1 unidade de tempo - (Iteração: 19)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300	20.00	0	0
(B)	4	4	1	146	125	7.00	3	3
(C)	10	3	2	250	240	10.00	3	3
(D)	5	3	2	72	68	4.00	3	3
(E)	12	3#	3	198	198	7.00	3	3
(F)	21	6#	6	430	430	14.33	6	6
(G)	8	5#	5	96	96	2.00	6	6
(H)	11	5#	5	132	132	2.00	11	11
(I)	5	1#	1	290	290	25.00	16	16
(J)	7	6	3	84	81	1.00	11	11
(K)	8	4	3	280	260	20.00	12	12
(L)	11	1#	1	420	420	10.00	16	16
(M)	24	4#	4	1240	1240	31.00	17	17
(N)	23	8#	8	380	380	8.00	21	21
(O)	13	13	3	570	410	16.00	17	17
(P)	14	9#	9	260	260	16.00	29	29
(Q)	17	14	7	1020	936	12.00	29	29
(R)	13	4#	4	380	380	20.00	38	38
(S)	9	9	6	270	210	20.00	43	43
(T)	19	10#	10	522	522	10.00	42	42
(U)	5	1#	1	200	200	10.00	52	52
(V)	12	2#	2	350	350	10.00	53	53
(X)	11	3#	3	424	424	11.00	55	55

Custos > Indiretos: 1160 Diretos: 7952 Projeto: 9112 Prazo: 58

Caminhos	Durações	Folgas	
ABHIMNPRTUVX	54	4	
ABHIMNQSUVX	54	4	
ABHIOVX	31	27	
ACGHIMNPRTUVX	58	0	==> Crítico
ACGHIMNQSUVX	58	0	==> Crítico
ACGHIOVX	35	23	
ACGJMNPTUVX	58	0	==> Crítico
ACGJMNSUVX	58	0	==> Crítico
ADGHIMNPRTUVX	58	0	==> Crítico
ADGHIMNQSUVX	58	0	==> Crítico
ADGHIOVX	35	23	
ADGJMNPTUVX	58	0	==> Crítico
ADGJMNSUVX	58	0	==> Crítico
AEFKLMNPRTUVX	58	0	==> Crítico
AEFKLMNQSUVX	58	0	==> Crítico
AEGHIMNPRTUVX	58	0	==> Crítico
AEGHIMNQSUVX	58	0	==> Crítico
AEGHIOVX	35	23	
AEGJMNPTUVX	58	0	==> Crítico
AEGJMNSUVX	58	0	==> Crítico

APÊNDICE E – Relatório Estudo de Caso 1 Versão Convexa

Legendas: # => Atividade totalmente acelerada
 Custo Fixo = 8 | Prazo = 0
 Critério de parada escolhido: Prazo Final
 Aceleração Total = 979
 Versão = Exponencial Convexa

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30.00	2.00	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	19	11	31	23.00	1.00	0	0
(D)	12	12	7	71	51.00	4.00	11	18
(E)	11	11	5	159	105.00	9.00	19	19
(F)	13	13	9	210	170.00	10.00	19	19
(G)	22	22	13	135	90.00	5.00	30	30
(H)	12	12	7	135	100.00	7.00	19	19
(I)	14	14	9	135	130.00	1.00	32	38

Custos > Indiretos: 416 Diretos: 741.00 Projeto: 1157.00 Prazo: 52

Caminhos	Duracoes	Folgas	
ADG	45	7	
BEG	45	7	
BH	24	28	
CEG	52	0	==> Crítico
CFI	46	6	
CH	31	21	

Acelerando Conjunto { C } em 7 unidades de tempo - (Iteração: 1)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30.00	2.00	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	11	11
(E)	11	11	5	159	105.00	9.00	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	22	13	135	90.00	5.00	23	23
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	14	9	135	130.00	1.00	25	31

Custos > Indiretos: 360 Diretos: 743.94 Projeto: 1103.94 Prazo: 45

Caminhos	Duracoes	Folgas	
ADG	45	0	==> Crítico
BEG	45	0	==> Crítico
BH	24	21	
CEG	45	0	==> Crítico
CFI	39	6	
CH	24	21	

Acelerando Conjunto { G } em 6 unidades de tempo - (Iteração: 2)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	11	6	40	30.00	2.00	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	11	11
(E)	11	11	5	159	105.00	9.00	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	16	13	135	92.24	2.24	23	23
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	14	9	135	130.00	1.00	25	25

Custos > Indiretos: 312 Diretos: 746.18 Projeto: 1058.18 Prazo: 39

Caminhos	Duracoes	Folgas	
ADG	39	0	==> Crítico
BEG	39	0	==> Crítico
BH	24	15	
CEG	39	0	==> Crítico
CFI	39	0	==> Crítico
CH	24	15	

Acelerando Conjunto { A E I } em 3 unidades de tempo - (Iteração: 3)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	8	8
(E)	11	8	5	159	107.69	2.69	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	16	13	135	92.24	2.24	20	20
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	11	9	135	130.68	0.68	25	25

Custos > Indiretos: 288 Diretos: 750.90 Projeto: 1038.90 Prazo: 36

Caminhos	Duracoes	Folgas	
ADG	36	0	==> Crítico
BEG	36	0	==> Crítico
BH	24	12	
CEG	36	0	==> Crítico
CFI	36	0	==> Crítico
CH	24	12	

Acelerando Conjunto { G I } em 1 unidade de tempo - (Iteração: 4)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	8	8
(E)	11	8	5	159	107.69	2.69	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	15	13	135	96.09	6.09	20	20
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	10	9	135	131.84	1.84	25	25

Custos > Indiretos: 280 Diretos: 755.91 Projeto: 1035.91 Prazo: 35

Caminhos	Duracoes	Folgas	
ADG	35	0	==> Crítico
BEG	35	0	==> Crítico
BH	24	11	
CEG	35	0	==> Crítico
CFI	35	0	==> Crítico
CH	24	11	

Acelerando Conjunto { G I } em 1 unidade de tempo - (Iteração: 5)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	8	8
(E)	11	8	5	159	107.69	2.69	12	12
(F)	13	13	9	210	170.00	10.00	12	12
(G)	22	14	13	135	106.55	16.55	20	20
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	9#	9	135	135.00	5.00	25	25

Custos > Indiretos: 272 Diretos: 769.54 Projeto: 1041.54 Prazo: 34

Caminhos	Duracoes	Folgas	
ADG	34	0	==> Crítico
BEG	34	0	==> Crítico
BH	24	10	
CEG	34	0	==> Crítico
CFI	34	0	==> Crítico
CH	24	10	

Acelerando Conjunto { F G } em 1 unidade de tempo - (Iteração: 6)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	12	7	71	51.00	4.00	8	8
(E)	11	8	5	159	107.69	2.69	12	12
(F)	13	12	9	210	171.99	1.99	12	12
(G)	22	13#	13	135	135.00	45.00	20	20
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	9#	9	135	135.00	5.00	24	24

Custos > Indiretos: 264 Diretos: 799.98 Projeto: 1063.98 Prazo: 33

Caminhos	Duracoes	Folgas	
ADG	33	0	==> Crítico
BEG	33	0	==> Crítico
BH	24	9	
CEG	33	0	==> Crítico
CFI	33	0	==> Crítico
CH	24	9	

Acelerando Conjunto { D E F } em 3 unidades de tempo - (Iteração: 7)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	12	5	63	42.00	3.00	0	0
(C)	19	12	11	31	25.94	2.94	0	0
(D)	12	9	7	71	53.71	2.71	8	8
(E)	11	5#	5	159	159.00	54.00	12	12
(F)	13	9#	9	210	210.00	40.00	12	12
(G)	22	13#	13	135	135.00	45.00	17	17
(H)	12	12	7	135	100.00	7.00	12	12
(I)	14	9#	9	135	135.00	5.00	21	21

Custos > Indiretos: 240 Diretos: 892.00 Projeto: 1132.00 Prazo: 30

Caminhos	Duracoes	Folgas	
ADG	30	0	==> Crítico
BEG	30	0	==> Crítico
BH	24	6	
CEG	30	0	==> Crítico
CFI	30	0	==> Crítico
CH	24	6	

Acelerando Conjunto { B C D } em 1 unidade de tempo - (Iteração: 8)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	11	8	6	40	31.35	1.35	0	0
(B)	12	11	5	63	42.05	0.05	0	0
(C)	19	11#	11	31	31.00	8.00	0	0
(D)	12	8	7	71	58.36	7.36	8	8
(E)	11	5#	5	159	159.00	54.00	11	11
(F)	13	9#	9	210	210.00	40.00	11	11
(G)	22	13#	13	135	135.00	45.00	16	16
(H)	12	12	7	135	100.00	7.00	11	11
(I)	14	9#	9	135	135.00	5.00	20	20

Custos > Indiretos: 232 Diretos: 901.76 Projeto: 1133.76 Prazo: 29

Caminhos	Duracoes	Folgas	
ADG	29	0	==> Crítico
BEG	29	0	==> Crítico
BH	23	6	
CEG	29	0	==> Crítico
CFI	29	0	==> Crítico
CH	23	6	

APÊNDICE F – Relatório Estudo de Caso 2 Versão Convexa

Legendas: # => Atividade totalmente acelerada
 Custo Fixo = 20 | Prazo = 0
 Critério de parada escolhido: Prazo Final
 Aceleração Total = 8314
 Versão = Exponencial Convexa

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	21	6	430	215.00	14.33	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	46
(I)	5	5	1	290	190.00	25.00	41	57
(J)	7	7	3	84	80.00	1.00	30	46
(K)	8	8	3	280	180.00	20.00	43	43
(L)	11	11	1	420	320.00	10.00	51	51
(M)	24	24	4	1240	620.00	31.00	62	62
(N)	23	23	8	380	260.00	8.00	86	86
(O)	13	13	3	570	410.00	16.00	46	62
(P)	14	14	9	260	180.00	16.00	109	109
(Q)	17	17	7	1020	900.00	12.00	109	109
(R)	13	13	4	380	200.00	20.00	123	123
(S)	9	9	6	270	210.00	20.00	126	146
(T)	19	19	10	522	432.00	10.00	136	136
(U)	5	5	1	200	160.00	10.00	155	155
(V)	12	12	2	350	250.00	10.00	160	160
(X)	11	11	3	424	336.00	11.00	172	172

Custos > Indiretos: 3660 Diretos: 5803.00 Projeto: 9463.00 Prazo: 183

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	151	32	
ABHIMNQSUVX	131	52	
ABHIOVX	66	117	
ACGHIMNPRTUVX	165	18	
ACGHIMNQSUVX	145	38	
ACGHIOVX	80	103	
ACGJMNPTUVX	156	27	
ACGJMNSUVX	136	47	
ADGHIMNPRTUVX	160	23	
ADGHIMNQSUVX	140	43	
ADGHIOVX	75	108	
ADGJMNPTUVX	151	32	
ADGJMNSUVX	131	52	
AEFKLMNPRTUVX	183	0	==> Crítico
AEFKLMNQSUVX	163	20	
AEGHIMNPRTUVX	167	16	
AEGHIMNQSUVX	147	36	
AEGHIOVX	82	101	
AEGJMNPTUVX	158	25	
AEGJMNSUVX	138	45	

Acelerando Conjunto { M } em 17 unidades de tempo - (Iteração: 1)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	21	6	430	215.00	14.33	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	46
(I)	5	5	1	290	190.00	25.00	41	57
(J)	7	7	3	84	80.00	1.00	30	46
(K)	8	8	3	280	180.00	20.00	43	43
(L)	11	11	1	420	320.00	10.00	51	51
(M)	24	7	4	1240	650.87	30.87	62	62
(N)	23	23	8	380	260.00	8.00	69	69
(O)	13	13	3	570	410.00	16.00	46	62

(P)	14	14	9	260	180.00	16.00	92	92
(Q)	17	17	7	1020	900.00	12.00	92	92
(R)	13	13	4	380	200.00	20.00	106	106
(S)	9	9	6	270	210.00	20.00	109	129
(T)	19	19	10	522	432.00	10.00	119	119
(U)	5	5	1	200	160.00	10.00	138	138
(V)	12	12	2	350	250.00	10.00	143	143
(X)	11	11	3	424	336.00	11.00	155	155

Custos > Indiretos: 3320 Diretos: 5833.87 Projeto: 9153.87 Prazo: 166

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	134	32	
ABHIMNQSUVX	114	52	
ABHIOVX	66	100	
ACGHIMNPRTUVX	148	18	
ACGHIMNQSUVX	128	38	
ACGHIOVX	80	86	
ACGJMNPTUVX	139	27	
ACGJMNSUVX	119	47	
ADGHIMNPRTUVX	143	23	
ADGHIMNQSUVX	123	43	
ADGHIOVX	75	91	
ADGJMNPTUVX	134	32	
ADGJMNSUVX	114	52	
AEFKLMNPRTUVX	166	0	==> Critico
AEFKLMNQSUVX	146	20	
AEGHIMNPRTUVX	150	16	
AEGHIMNQSUVX	130	36	
AEGHIOVX	82	84	
AEGJMNPTUVX	141	25	
AEGJMNSUVX	121	45	

Acelerando Conjunto { N } em 13 unidades de tempo - (Iteração: 2)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	21	6	430	215.00	14.33	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	46
(I)	5	5	1	290	190.00	25.00	41	57
(J)	7	7	3	84	80.00	1.00	30	46
(K)	8	8	3	280	180.00	20.00	43	43
(L)	11	11	1	420	320.00	10.00	51	51
(M)	24	7	4	1240	650.87	30.87	62	62
(N)	23	10	8	380	276.24	16.24	69	69
(O)	13	13	3	570	410.00	16.00	46	62
(P)	14	14	9	260	180.00	16.00	79	79
(Q)	17	17	7	1020	900.00	12.00	79	79
(R)	13	13	4	380	200.00	20.00	93	93
(S)	9	9	6	270	210.00	20.00	96	116
(T)	19	19	10	522	432.00	10.00	106	106
(U)	5	5	1	200	160.00	10.00	125	125
(V)	12	12	2	350	250.00	10.00	130	130
(X)	11	11	3	424	336.00	11.00	142	142

Custos > Indiretos: 3060 Diretos: 5850.11 Projeto: 8910.11 Prazo: 153

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	121	32	
ABHIMNQSUVX	101	52	
ABHIOVX	66	87	
ACGHIMNPRTUVX	135	18	
ACGHIMNQSUVX	115	38	
ACGHIOVX	80	73	
ACGJMNPTUVX	126	27	
ACGJMNSUVX	106	47	
ADGHIMNPRTUVX	130	23	
ADGHIMNQSUVX	110	43	
ADGHIOVX	75	78	
ADGJMNPTUVX	121	32	
ADGJMNSUVX	101	52	
AEFKLMNPRTUVX	153	0	==> Critico
AEFKLMNQSUVX	133	20	
AEGHIMNPRTUVX	137	16	

AEGHIMNQSUVX	117	36
AEGHIOVX	82	71
AEGJMNPRUVX	128	25
AEGJMNQSUVX	108	45

Acelerando Conjunto { F } em 13 unidades de tempo - (Iteração: 3)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	8	6	430	244.10	29.10	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	33
(I)	5	5	1	290	190.00	25.00	41	44
(J)	7	7	3	84	80.00	1.00	30	33
(K)	8	8	3	280	180.00	20.00	30	30
(L)	11	11	1	420	320.00	10.00	38	38
(M)	24	7	4	1240	650.87	30.87	49	49
(N)	23	10	8	380	276.24	16.24	56	56
(O)	13	13	3	570	410.00	16.00	46	49
(P)	14	14	9	260	180.00	16.00	66	66
(Q)	17	17	7	1020	900.00	12.00	66	66
(R)	13	13	4	380	200.00	20.00	80	80
(S)	9	9	6	270	210.00	20.00	83	103
(T)	19	19	10	522	432.00	10.00	93	93
(U)	5	5	1	200	160.00	10.00	112	112
(V)	12	12	2	350	250.00	10.00	117	117
(X)	11	11	3	424	336.00	11.00	129	129

Custos > Indiretos: 2800 Diretos: 5879.21 Projeto: 8679.21 Prazo: 140

Caminhos	Duracoes	Folgas	
ABHIMNPRUVX	121	19	
ABHIMNQSUVX	101	39	
ABHIOVX	66	74	
ACGHIMNPRUVX	135	5	
ACGHIMNQSUVX	115	25	
ACGHIOVX	80	60	
ACGJMNPRUVX	126	14	
ACGJMNQSUVX	106	34	
ADGHIMNPRUVX	130	10	
ADGHIMNQSUVX	110	30	
ADGHIOVX	75	65	
ADGJMNPRUVX	121	19	
ADGJMNQSUVX	101	39	
AEFKLMNPRUVX	140	0	==> Critico
AEFKLMNQSUVX	120	20	
AEGHIMNPRUVX	137	3	
AEGHIMNQSUVX	117	23	
AEGHIOVX	82	58	
AEGJMNPRUVX	128	12	
AEGJMNQSUVX	108	32	

Acelerando Conjunto { V } em 8 unidades de tempo - (Iteração: 4)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	8	6	430	244.10	29.10	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	33
(I)	5	5	1	290	190.00	25.00	41	44
(J)	7	7	3	84	80.00	1.00	30	33
(K)	8	8	3	280	180.00	20.00	30	30
(L)	11	11	1	420	320.00	10.00	38	38
(M)	24	7	4	1240	650.87	30.87	49	49
(N)	23	10	8	380	276.24	16.24	56	56
(O)	13	13	3	570	410.00	16.00	46	49
(P)	14	14	9	260	180.00	16.00	66	66
(Q)	17	17	7	1020	900.00	12.00	66	66
(R)	13	13	4	380	200.00	20.00	80	80
(S)	9	9	6	270	210.00	20.00	83	103

(T)	19	19	10	522	432.00	10.00	93	93
(U)	5	5	1	200	160.00	10.00	112	112
(V)	12	4	2	350	263.53	13.53	117	117
(X)	11	11	3	424	336.00	11.00	121	121

Custos > Indiretos: 2640 Diretos: 5892.74 Projeto: 8532.74 Prazo: 132

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	113	19	
ABHIMNQSUVX	93	39	
ABHIOVX	58	74	
ACGHIMNPRTUVX	127	5	
ACGHIMNQSUVX	107	25	
ACGHIOVX	72	60	
ACGJMNPTUVX	118	14	
ACGJMNSUVX	98	34	
ADGHIMNPRTUVX	122	10	
ADGHIMNQSUVX	102	30	
ADGHIOVX	67	65	
ADGJMNPTUVX	113	19	
ADGJMNSUVX	93	39	
AEFKLMNPRTUVX	132	0	==> Critico
AEFKLMNQSUVX	112	20	
AEGHIMNPRTUVX	129	3	
AEGHIMNQSUVX	109	23	
AEGHIOVX	74	58	
AEGJMNPTUVX	120	12	
AEGJMNSUVX	100	32	

Acelerando Conjunto { T } em 7 unidades de tempo - (Iteração: 5)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	8	6	430	244.10	29.10	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	33
(I)	5	5	1	290	190.00	25.00	41	44
(J)	7	7	3	84	80.00	1.00	30	33
(K)	8	8	3	280	180.00	20.00	30	30
(L)	11	11	1	420	320.00	10.00	38	38
(M)	24	7	4	1240	650.87	30.87	49	49
(N)	23	10	8	380	276.24	16.24	56	56
(O)	13	13	3	570	410.00	16.00	46	49
(P)	14	14	9	260	180.00	16.00	66	66
(Q)	17	17	7	1020	900.00	12.00	66	66
(R)	13	13	4	380	200.00	20.00	80	80
(S)	9	9	6	270	210.00	20.00	83	96
(T)	19	12	10	522	444.18	12.18	93	93
(U)	5	5	1	200	160.00	10.00	105	105
(V)	12	4	2	350	263.53	13.53	110	110
(X)	11	11	3	424	336.00	11.00	114	114

Custos > Indiretos: 2500 Diretos: 5904.92 Projeto: 8404.92 Prazo: 125

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	106	19	
ABHIMNQSUVX	93	32	
ABHIOVX	58	67	
ACGHIMNPRTUVX	120	5	
ACGHIMNQSUVX	107	18	
ACGHIOVX	72	53	
ACGJMNPTUVX	111	14	
ACGJMNSUVX	98	27	
ADGHIMNPRTUVX	115	10	
ADGHIMNQSUVX	102	23	
ADGHIOVX	67	58	
ADGJMNPTUVX	106	19	
ADGJMNSUVX	93	32	
AEFKLMNPRTUVX	125	0	==> Critico
AEFKLMNQSUVX	112	13	
AEGHIMNPRTUVX	122	3	
AEGHIMNQSUVX	109	16	
AEGHIOVX	74	51	
AEGJMNPTUVX	113	12	
AEGJMNSUVX	100	25	

 Acelerando Conjunto { R } em 7 unidades de tempo - (Iteração: 6)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	8	6	430	244.10	29.10	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	33
(I)	5	5	1	290	190.00	25.00	41	44
(J)	7	7	3	84	80.00	1.00	30	33
(K)	8	8	3	280	180.00	20.00	30	30
(L)	11	11	1	420	320.00	10.00	38	38
(M)	24	7	4	1240	650.87	30.87	49	49
(N)	23	10	8	380	276.24	16.24	56	56
(O)	13	13	3	570	410.00	16.00	46	49
(P)	14	14	9	260	180.00	16.00	66	66
(Q)	17	17	7	1020	900.00	12.00	66	66
(R)	13	6	4	380	224.36	24.36	80	80
(S)	9	9	6	270	210.00	20.00	83	89
(T)	19	12	10	522	444.18	12.18	86	86
(U)	5	5	1	200	160.00	10.00	98	98
(V)	12	4	2	350	263.53	13.53	103	103
(X)	11	11	3	424	336.00	11.00	107	107

Custos > Indiretos: 2360 Diretos: 5929.28 Projeto: 8289.28 Prazo: 118

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	99	19	
ABHIMNQSUVX	93	25	
ABHIOVX	58	60	
ACGHIMNPRTUVX	113	5	
ACGHIMNQSUVX	107	11	
ACGHIOVX	72	46	
ACGJMNPTUVX	104	14	
ACGJMNSUVX	98	20	
ADGHIMNPRTUVX	108	10	
ADGHIMNQSUVX	102	16	
ADGHIOVX	67	51	
ADGJMNPTUVX	99	19	
ADGJMNSUVX	93	25	
AEFKLMNPRTUVX	118	0	==> Critico
AEFKLMNQSUVX	112	6	
AEGHIMNPRTUVX	115	3	
AEGHIMNQSUVX	109	9	
AEGHIOVX	74	44	
AEGJMNPTUVX	106	12	
AEGJMNSUVX	100	18	

 Acelerando Conjunto { X } em 6 unidades de tempo - (Iteração: 7)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	12	3	198	135.00	7.00	10	10
(F)	21	8	6	430	244.10	29.10	22	22
(G)	8	8	5	96	90.00	2.00	22	22
(H)	11	11	5	132	120.00	2.00	30	33
(I)	5	5	1	290	190.00	25.00	41	44
(J)	7	7	3	84	80.00	1.00	30	33
(K)	8	8	3	280	180.00	20.00	30	30
(L)	11	11	1	420	320.00	10.00	38	38
(M)	24	7	4	1240	650.87	30.87	49	49
(N)	23	10	8	380	276.24	16.24	56	56
(O)	13	13	3	570	410.00	16.00	46	49
(P)	14	14	9	260	180.00	16.00	66	66
(Q)	17	17	7	1020	900.00	12.00	66	66
(R)	13	6	4	380	224.36	24.36	80	80
(S)	9	9	6	270	210.00	20.00	83	89
(T)	19	12	10	522	444.18	12.18	86	86
(U)	5	5	1	200	160.00	10.00	98	98
(V)	12	4	2	350	263.53	13.53	103	103
(X)	11	5	3	424	347.91	11.91	107	107

Custos > Indiretos: 2240 Diretos: 5941.19 Projeto: 8181.19 Prazo: 112

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	93	19	
ABHIMNQSUVX	87	25	
ABHIOVX	52	60	
ACGHIMNPRTUVX	107	5	
ACGHIMNQSUVX	101	11	
ACGHIOVX	66	46	
ACGJMNPTUVX	98	14	
ACGJMNSUVX	92	20	
ADGHIMNPRTUVX	102	10	
ADGHIMNQSUVX	96	16	
ADGHIOVX	61	51	
ADGJMNPTUVX	93	19	
ADGJMNSUVX	87	25	
AEFKLMNPRTUVX	112	0	==> Critico
AEFKLMNQSUVX	106	6	
AEGHIMNPRTUVX	109	3	
AEGHIMNQSUVX	103	9	
AEGHIOVX	68	44	
AEGJMNPTUVX	100	12	
AEGJMNSUVX	94	18	

Acelerando Conjunto { E } em 5 unidades de tempo - (Iteração: 8)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	7	3	198	136.15	1.15	10	10
(F)	21	8	6	430	244.10	29.10	17	17
(G)	8	8	5	96	90.00	2.00	20	17
(H)	11	11	5	132	120.00	2.00	28	28
(I)	5	5	1	290	190.00	25.00	39	39
(J)	7	7	3	84	80.00	1.00	28	28
(K)	8	8	3	280	180.00	20.00	25	25
(L)	11	11	1	420	320.00	10.00	33	33
(M)	24	7	4	1240	650.87	30.87	44	44
(N)	23	10	8	380	276.24	16.24	51	51
(O)	13	13	3	570	410.00	16.00	44	44
(P)	14	14	9	260	180.00	16.00	61	61
(Q)	17	17	7	1020	900.00	12.00	61	61
(R)	13	6	4	380	224.36	24.36	75	75
(S)	9	9	6	270	210.00	20.00	78	84
(T)	19	12	10	522	444.18	12.18	81	81
(U)	5	5	1	200	160.00	10.00	93	93
(V)	12	4	2	350	263.53	13.53	98	98
(X)	11	5	3	424	347.91	11.91	102	102

Custos > Indiretos: 2140 Diretos: 5942.34 Projeto: 8082.34 Prazo: 107

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	93	14	
ABHIMNQSUVX	87	20	
ABHIOVX	52	55	
ACGHIMNPRTUVX	107	0	==> Critico
ACGHIMNQSUVX	101	6	
ACGHIOVX	66	41	
ACGJMNPTUVX	98	9	
ACGJMNSUVX	92	15	
ADGHIMNPRTUVX	102	5	
ADGHIMNQSUVX	96	11	
ADGHIOVX	61	46	
ADGJMNPTUVX	93	14	
ADGJMNSUVX	87	20	
AEFKLMNPRTUVX	107	0	==> Critico
AEFKLMNQSUVX	101	6	
AEGHIMNPRTUVX	104	3	
AEGHIMNQSUVX	98	9	
AEGHIOVX	63	44	
AEGJMNPTUVX	95	12	
AEGJMNSUVX	89	18	

Acelerando Conjunto { H L } em 6 unidades de tempo - (Iteração: 9)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	10	3	300	160.00	20.00	0	0
(B)	4	4	1	146	125.00	7.00	10	10
(C)	10	10	2	250	170.00	10.00	10	10
(D)	5	5	2	72	60.00	4.00	10	10
(E)	12	7	3	198	136.15	1.15	10	10
(F)	21	8	6	430	244.10	29.10	17	17
(G)	8	8	5	96	90.00	2.00	20	17
(H)	11	5#	5	132	132.00	12.00	28	28
(I)	5	5	1	290	190.00	25.00	33	33
(J)	7	7	3	84	80.00	1.00	28	28
(K)	8	8	3	280	180.00	20.00	25	25
(L)	11	5	1	420	321.83	1.83	33	33
(M)	24	7	4	1240	650.87	30.87	38	38
(N)	23	10	8	380	276.24	16.24	45	45
(O)	13	13	3	570	410.00	16.00	38	38
(P)	14	14	9	260	180.00	16.00	55	55
(Q)	17	17	7	1020	900.00	12.00	55	55
(R)	13	6	4	380	224.36	24.36	69	69
(S)	9	9	6	270	210.00	20.00	72	78
(T)	19	12	10	522	444.18	12.18	75	75
(U)	5	5	1	200	160.00	10.00	87	87
(V)	12	4	2	350	263.53	13.53	92	92
(X)	11	5	3	424	347.91	11.91	96	96

Custos > Indiretos: 2020 Diretos: 5956.17 Projeto: 7976.17 Prazo: 101

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	87	14	
ABHIMNQSUVX	81	20	
ABHIOVX	46	55	
ACGHIMNPRTUVX	101	0	==> Critico
ACGHIMNQSUVX	95	6	
ACGHIOVX	60	41	
ACGJMNPTUVX	98	3	
ACGJMNSUVX	92	9	
ADGHIMNPRTUVX	96	5	
ADGHIMNQSUVX	90	11	
ADGHIOVX	55	46	
ADGJMNPTUVX	93	8	
ADGJMNSUVX	87	14	
AEFKLMNPRTUVX	101	0	==> Critico
AEFKLMNQSUVX	95	6	
AEGHIMNPRTUVX	98	3	
AEGHIMNQSUVX	92	9	
AEGHIOVX	57	44	
AEGJMNPTUVX	95	6	
AEGJMNSUVX	89	12	

Acelerando Conjunto { A } em 5 unidades de tempo - (Iteração: 10)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	10	2	250	170.00	10.00	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	7	3	198	136.15	1.15	5	5
(F)	21	8	6	430	244.10	29.10	12	12
(G)	8	8	5	96	90.00	2.00	15	12
(H)	11	5#	5	132	132.00	12.00	23	23
(I)	5	5	1	290	190.00	25.00	28	28
(J)	7	7	3	84	80.00	1.00	23	23
(K)	8	8	3	280	180.00	20.00	20	20
(L)	11	5	1	420	321.83	1.83	28	28
(M)	24	7	4	1240	650.87	30.87	33	33
(N)	23	10	8	380	276.24	16.24	40	40
(O)	13	13	3	570	410.00	16.00	33	33
(P)	14	14	9	260	180.00	16.00	50	50
(Q)	17	17	7	1020	900.00	12.00	50	50
(R)	13	6	4	380	224.36	24.36	64	64
(S)	9	9	6	270	210.00	20.00	67	73
(T)	19	12	10	522	444.18	12.18	70	70
(U)	5	5	1	200	160.00	10.00	82	82
(V)	12	4	2	350	263.53	13.53	87	87
(X)	11	5	3	424	347.91	11.91	91	91

Custos > Indiretos: 1920 Diretos: 5975.12 Projeto: 7895.12 Prazo: 96

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	82	14	
ABHIMNQSUVX	76	20	
ABHIOVX	41	55	
ACGHIMNPRTUVX	96	0	==> Critico
ACGHIMNQSUVX	90	6	
ACGHIOVX	55	41	
ACGJMNPTUVX	93	3	
ACGJMNSUVX	87	9	
ADGHIMNPRTUVX	91	5	
ADGHIMNQSUVX	85	11	
ADGHIOVX	50	46	
ADGJMNPTUVX	88	8	
ADGJMNSUVX	82	14	
AEFKLMNPRTUVX	96	0	==> Critico
AEFKLMNQSUVX	90	6	
AEGHIMNPRTUVX	93	3	
AEGHIMNQSUVX	87	9	
AEGHIOVX	52	44	
AEGJMNPTUVX	90	6	
AEGJMNSUVX	84	12	

Acelerando Conjunto { P } em 4 unidades de tempo - (Iteração: 11)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	10	2	250	170.00	10.00	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	7	3	198	136.15	1.15	5	5
(F)	21	8	6	430	244.10	29.10	12	12
(G)	8	8	5	96	90.00	2.00	15	12
(H)	11	5#	5	132	132.00	12.00	23	23
(I)	5	5	1	290	190.00	25.00	28	28
(J)	7	7	3	84	80.00	1.00	23	23
(K)	8	8	3	280	180.00	20.00	20	20
(L)	11	5	1	420	321.83	1.83	28	28
(M)	24	7	4	1240	650.87	30.87	33	33
(N)	23	10	8	380	276.24	16.24	40	40
(O)	13	13	3	570	410.00	16.00	33	33
(P)	14	10	9	260	209.43	29.43	50	50
(Q)	17	17	7	1020	900.00	12.00	50	50
(R)	13	6	4	380	224.36	24.36	60	60
(S)	9	9	6	270	210.00	20.00	67	69
(T)	19	12	10	522	444.18	12.18	66	66
(U)	5	5	1	200	160.00	10.00	78	78
(V)	12	4	2	350	263.53	13.53	83	83
(X)	11	5	3	424	347.91	11.91	87	87

Custos > Indiretos: 1840 Diretos: 6004.55 Projeto: 7844.55 Prazo: 92

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	78	14	
ABHIMNQSUVX	76	16	
ABHIOVX	41	51	
ACGHIMNPRTUVX	92	0	==> Critico
ACGHIMNQSUVX	90	2	
ACGHIOVX	55	37	
ACGJMNPTUVX	89	3	
ACGJMNSUVX	87	5	
ADGHIMNPRTUVX	87	5	
ADGHIMNQSUVX	85	7	
ADGHIOVX	50	42	
ADGJMNPTUVX	84	8	
ADGJMNSUVX	82	10	
AEFKLMNPRTUVX	92	0	==> Critico
AEFKLMNQSUVX	90	2	
AEGHIMNPRTUVX	89	3	
AEGHIMNQSUVX	87	5	
AEGHIOVX	52	40	
AEGJMNPTUVX	86	6	
AEGJMNSUVX	84	8	

Acelerando Conjunto { C K } em 3 unidades de tempo - (Iteração: 12)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	7	2	250	170.54	0.54	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	7	3	198	136.15	1.15	5	5
(F)	21	8	6	430	244.10	29.10	12	12
(G)	8	8	5	96	90.00	2.00	12	12
(H)	11	5#	5	132	132.00	12.00	20	20
(I)	5	5	1	290	190.00	25.00	25	25
(J)	7	7	3	84	80.00	1.00	20	20
(K)	8	5	3	280	193.53	13.53	20	20
(L)	11	5	1	420	321.83	1.83	25	25
(M)	24	7	4	1240	650.87	30.87	30	30
(N)	23	10	8	380	276.24	16.24	37	37
(O)	13	13	3	570	410.00	16.00	30	30
(P)	14	10	9	260	209.43	29.43	47	47
(Q)	17	17	7	1020	900.00	12.00	47	47
(R)	13	6	4	380	224.36	24.36	57	57
(S)	9	9	6	270	210.00	20.00	64	66
(T)	19	12	10	522	444.18	12.18	63	63
(U)	5	5	1	200	160.00	10.00	75	75
(V)	12	4	2	350	263.53	13.53	80	80
(X)	11	5	3	424	347.91	11.91	84	84

Custos > Indiretos: 1780 Diretos: 6018.62 Projeto: 7798.62 Prazo: 89

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	78	11	
ABHIMNQSUVX	76	13	
ABHIOVX	41	48	
ACGHIMNPRTUVX	89	0	==> Critico
ACGHIMNQSUVX	87	2	
ACGHIOVX	52	37	
ACGJMNPTUVX	86	3	
ACGJMNSUVX	84	5	
ADGHIMNPRTUVX	87	2	
ADGHIMNQSUVX	85	4	
ADGHIOVX	50	39	
ADGJMNPRTUVX	84	5	
ADGJMNSUVX	82	7	
AEFKLMNPRTUVX	89	0	==> Critico
AEFKLMNQSUVX	87	2	
AEGHIMNPRTUVX	89	0	==> Critico
AEGHIMNQSUVX	87	2	
AEGHIOVX	52	37	
AEGJMNPRTUVX	86	3	
AEGJMNSUVX	84	5	

Acelerando Conjunto { U } em 3 unidades de tempo - (Iteração: 13)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	7	2	250	170.54	0.54	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	7	3	198	136.15	1.15	5	5
(F)	21	8	6	430	244.10	29.10	12	12
(G)	8	8	5	96	90.00	2.00	12	12
(H)	11	5#	5	132	132.00	12.00	20	20
(I)	5	5	1	290	190.00	25.00	25	25
(J)	7	7	3	84	80.00	1.00	20	20
(K)	8	5	3	280	193.53	13.53	20	20
(L)	11	5	1	420	321.83	1.83	25	25
(M)	24	7	4	1240	650.87	30.87	30	30
(N)	23	10	8	380	276.24	16.24	37	37
(O)	13	13	3	570	410.00	16.00	30	30
(P)	14	10	9	260	209.43	29.43	47	47
(Q)	17	17	7	1020	900.00	12.00	47	47
(R)	13	6	4	380	224.36	24.36	57	57
(S)	9	9	6	270	210.00	20.00	64	66
(T)	19	12	10	522	444.18	12.18	63	63
(U)	5	2	1	200	174.72	14.72	75	75
(V)	12	4	2	350	263.53	13.53	77	77
(X)	11	5	3	424	347.91	11.91	81	81

Custos > Indiretos: 1720 Diretos: 6033.34 Projeto: 7753.34 Prazo: 86

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	75	11	
ABHIMNQSUVX	73	13	
ABHIOVX	41	45	
ACGHIMNPRTUVX	86	0	==> Critico
ACGHIMNQSUVX	84	2	
ACGHIOVX	52	34	
ACGJMNPTUVX	83	3	
ACGJMNSUVX	81	5	
ADGHIMNPRTUVX	84	2	
ADGHIMNQSUVX	82	4	
ADGHIOVX	50	36	
ADGJMNPTUVX	81	5	
ADGJMNSUVX	79	7	
AEFKLMNPRTUVX	86	0	==> Critico
AEFKLMNQSUVX	84	2	
AEGHIMNPRTUVX	86	0	==> Critico
AEGHIMNQSUVX	84	2	
AEGHIOVX	52	34	
AEGJMNPTUVX	83	3	
AEGJMNSUVX	81	5	

Acelerando Conjunto { C E } em 2 unidades de tempo - (Iteração: 14)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	8	5	96	90.00	2.00	10	10
(H)	11	5#	5	132	132.00	12.00	18	18
(I)	5	5	1	290	190.00	25.00	23	23
(J)	7	7	3	84	80.00	1.00	18	18
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	5	1	420	321.83	1.83	23	23
(M)	24	7	4	1240	650.87	30.87	28	28
(N)	23	10	8	380	276.24	16.24	35	35
(O)	13	13	3	570	410.00	16.00	28	28
(P)	14	10	9	260	209.43	29.43	45	45
(Q)	17	17	7	1020	900.00	12.00	45	45
(R)	13	6	4	380	224.36	24.36	55	55
(S)	9	9	6	270	210.00	20.00	62	64
(T)	19	12	10	522	444.18	12.18	61	61
(U)	5	2	1	200	174.72	14.72	73	73
(V)	12	4	2	350	263.53	13.53	75	75
(X)	11	5	3	424	347.91	11.91	79	79

Custos > Indiretos: 1680 Diretos: 6044.16 Projeto: 7724.16 Prazo: 84

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	75	9	
ABHIMNQSUVX	73	11	
ABHIOVX	41	43	
ACGHIMNPRTUVX	84	0	==> Critico
ACGHIMNQSUVX	82	2	
ACGHIOVX	50	34	
ACGJMNPTUVX	81	3	
ACGJMNSUVX	79	5	
ADGHIMNPRTUVX	84	0	==> Critico
ADGHIMNQSUVX	82	2	
ADGHIOVX	50	34	
ADGJMNPTUVX	81	3	
ADGJMNSUVX	79	5	
AEFKLMNPRTUVX	84	0	==> Critico
AEFKLMNQSUVX	82	2	
AEGHIMNPRTUVX	84	0	==> Critico
AEGHIMNQSUVX	82	2	
AEGHIOVX	50	34	
AEGJMNPTUVX	81	3	
AEGJMNSUVX	79	5	

Acelerando Conjunto { G L } em 2 unidades de tempo - (Iteração: 15)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	10	8	380	276.24	16.24	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	10	9	260	209.43	29.43	43	43
(Q)	17	17	7	1020	900.00	12.00	43	43
(R)	13	6	4	380	224.36	24.36	53	53
(S)	9	9	6	270	210.00	20.00	60	62
(T)	19	12	10	522	444.18	12.18	59	59
(U)	5	2	1	200	174.72	14.72	71	71
(V)	12	4	2	350	263.53	13.53	73	73
(X)	11	5	3	424	347.91	11.91	77	77

Custos > Indiretos: 1640 Diretos: 6058.07 Projeto: 7698.07 Prazo: 82

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	75	7	
ABHIMNQSUVX	73	9	
ABHIOVX	41	41	
ACGHIMNPRTUVX	82	0	==> Critico
ACGHIMNQSUVX	80	2	
ACGHIOVX	48	34	
ACGJMNPTUVX	79	3	
ACGJMNSUVX	77	5	
ADGHIMNPRTUVX	82	0	==> Critico
ADGHIMNQSUVX	80	2	
ADGHIOVX	48	34	
ADGJMNPRTUVX	79	3	
ADGJMNSUVX	77	5	
AEFKLMNPRTUVX	82	0	==> Critico
AEFKLMNQSUVX	80	2	
AEGHIMNPRTUVX	82	0	==> Critico
AEGHIMNQSUVX	80	2	
AEGHIOVX	48	34	
AEGJMNPRTUVX	79	3	
AEGJMNSUVX	77	5	

Acelerando Conjunto { X } em 2 unidades de tempo - (Iteração: 16)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	10	8	380	276.24	16.24	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	10	9	260	209.43	29.43	43	43
(Q)	17	17	7	1020	900.00	12.00	43	43
(R)	13	6	4	380	224.36	24.36	53	53
(S)	9	9	6	270	210.00	20.00	60	62
(T)	19	12	10	522	444.18	12.18	59	59
(U)	5	2	1	200	174.72	14.72	71	71
(V)	12	4	2	350	263.53	13.53	73	73
(X)	11	3#	3	424	424.00	88.00	77	77

Custos > Indiretos: 1600 Diretos: 6134.16 Projeto: 7734.16 Prazo: 80

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	73	7	
ABHIMNQSUVX	71	9	
ABHIOVX	39	41	
ACGHIMNPRTUVX	80	0	==> Critico
ACGHIMNQSUVX	78	2	
ACGHIOVX	46	34	
ACGJMNPTUVX	77	3	
ACGJMNSUVX	75	5	
ADGHIMNPRTUVX	80	0	==> Critico
ADGHIMNQSUVX	78	2	
ADGHIOVX	46	34	
ADGJMNPRTUVX	77	3	
ADGJMNSUVX	75	5	
AEFKLMNPRTUVX	80	0	==> Critico
AEFKLMNQSUVX	78	2	
AEGHIMNPRTUVX	80	0	==> Critico
AEGHIMNQSUVX	78	2	
AEGHIOVX	46	34	
AEGJMNPTUVX	77	3	
AEGJMNSUVX	75	5	

Acelerando Conjunto { V } em 2 unidades de tempo - (Iteração: 17)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	10	8	380	276.24	16.24	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	10	9	260	209.43	29.43	43	43
(Q)	17	17	7	1020	900.00	12.00	43	43
(R)	13	6	4	380	224.36	24.36	53	53
(S)	9	9	6	270	210.00	20.00	60	62
(T)	19	12	10	522	444.18	12.18	59	59
(U)	5	2	1	200	174.72	14.72	71	71
(V)	12	2#	2	350	350.00	100.00	73	73
(X)	11	3#	3	424	424.00	88.00	75	75

Custos > Indiretos: 1560 Diretos: 6220.62 Projeto: 7780.62 Prazo: 78

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	71	7	
ABHIMNQSUVX	69	9	
ABHIOVX	37	41	
ACGHIMNPRTUVX	78	0	==> Critico
ACGHIMNQSUVX	76	2	
ACGHIOVX	44	34	
ACGJMNPTUVX	75	3	
ACGJMNSUVX	73	5	
ADGHIMNPRTUVX	78	0	==> Critico
ADGHIMNQSUVX	76	2	
ADGHIOVX	44	34	
ADGJMNPRTUVX	75	3	
ADGJMNSUVX	73	5	
AEFKLMNPRTUVX	78	0	==> Critico
AEFKLMNQSUVX	76	2	
AEGHIMNPRTUVX	78	0	==> Critico
AEGHIMNQSUVX	76	2	
AEGHIOVX	44	34	
AEGJMNPTUVX	75	3	
AEGJMNSUVX	73	5	

Acelerando Conjunto { U } em 1 unidade de tempo - (Iteração: 18)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	10	8	380	276.24	16.24	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	10	9	260	209.43	29.43	43	43
(Q)	17	17	7	1020	900.00	12.00	43	43
(R)	13	6	4	380	224.36	24.36	53	53
(S)	9	9	6	270	210.00	20.00	60	62
(T)	19	12	10	522	444.18	12.18	59	59
(U)	5	1#	1	200	200.00	40.00	71	71
(V)	12	2#	2	350	350.00	100.00	72	72
(X)	11	3#	3	424	424.00	88.00	74	74

Custos > Indiretos: 1540 Diretos: 6245.91 Projeto: 7785.91 Prazo: 77

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	70	7	
ABHIMNQSUVX	68	9	
ABHIOVX	37	40	
ACGHIMNPRTUVX	77	0	==> Critico
ACGHIMNQSUVX	75	2	
ACGHIOVX	44	33	
ACGJMNPTUVX	74	3	
ACGJMNSUVX	72	5	
ADGHIMNPRTUVX	77	0	==> Critico
ADGHIMNQSUVX	75	2	
ADGHIOVX	44	33	
ADGJMNPRTUVX	74	3	
ADGJMNSUVX	72	5	
AEFKLMNPRTUVX	77	0	==> Critico
AEFKLMNQSUVX	75	2	
AEGHIMNPRTUVX	77	0	==> Critico
AEGHIMNQSUVX	75	2	
AEGHIOVX	44	33	
AEGJMNPRTUVX	74	3	
AEGJMNSUVX	72	5	

Acelerando Conjunto { T } em 2 unidades de tempo - (Iteração: 19)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	10	8	380	276.24	16.24	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	10	9	260	209.43	29.43	43	43
(Q)	17	17	7	1020	900.00	12.00	43	43
(R)	13	6	4	380	224.36	24.36	53	53
(S)	9	9	6	270	210.00	20.00	60	60
(T)	19	10#	10	522	522.00	90.00	59	59
(U)	5	1#	1	200	200.00	40.00	69	69
(V)	12	2#	2	350	350.00	100.00	70	70
(X)	11	3#	3	424	424.00	88.00	72	72

Custos > Indiretos: 1500 Diretos: 6323.73 Projeto: 7823.73 Prazo: 75

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	68	7	
ABHIMNQSUVX	68	7	
ABHIOVX	37	38	
ACGHIMNPRTUVX	75	0	==> Crítico
ACGHIMNQSUVX	75	0	==> Crítico
ACGHIOVX	44	31	
ACGJMNPTUVX	72	3	
ACGJMNSUVX	72	3	
ADGHIMNPRTUVX	75	0	==> Crítico
ADGHIMNQSUVX	75	0	==> Crítico
ADGHIOVX	44	31	
ADGJMNPTUVX	72	3	
ADGJMNSUVX	72	3	
AEFKLMNPRTUVX	75	0	==> Crítico
AEFKLMNQSUVX	75	0	==> Crítico
AEGHIMNPRTUVX	75	0	==> Crítico
AEGHIMNQSUVX	75	0	==> Crítico
AEGHIOVX	44	31	
AEGJMNPTUVX	72	3	
AEGJMNSUVX	72	3	

Acelerando Conjunto { R S } em 2 unidades de tempo - (Iteração: 20)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	10	8	380	276.24	16.24	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	10	9	260	209.43	29.43	43	43
(Q)	17	17	7	1020	900.00	12.00	43	43
(R)	13	4#	4	380	380.00	180.00	53	53
(S)	9	7	6	270	232.07	22.07	60	60
(T)	19	10#	10	522	522.00	90.00	57	57
(U)	5	1#	1	200	200.00	40.00	67	67
(V)	12	2#	2	350	350.00	100.00	68	68
(X)	11	3#	3	424	424.00	88.00	70	70

Custos > Indiretos: 1460 Diretos: 6501.44 Projeto: 7961.44 Prazo: 73

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	66	7	
ABHIMNQSUVX	66	7	
ABHIOVX	37	36	
ACGHIMNPRTUVX	73	0	==> Crítico
ACGHIMNQSUVX	73	0	==> Crítico
ACGHIOVX	44	29	
ACGJMNPTUVX	70	3	
ACGJMNSUVX	70	3	
ADGHIMNPRTUVX	73	0	==> Crítico
ADGHIMNQSUVX	73	0	==> Crítico
ADGHIOVX	44	29	
ADGJMNPTUVX	70	3	
ADGJMNSUVX	70	3	
AEFKLMNPRTUVX	73	0	==> Crítico
AEFKLMNQSUVX	73	0	==> Crítico
AEGHIMNPRTUVX	73	0	==> Crítico
AEGHIMNQSUVX	73	0	==> Crítico
AEGHIOVX	44	29	
AEGJMNPTUVX	70	3	
AEGJMNSUVX	70	3	

Acelerando Conjunto { P S } em 1 unidade de tempo - (Iteração: 21)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	10	8	380	276.24	16.24	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	9#	9	260	260.00	80.00	43	43
(Q)	17	17	7	1020	900.00	12.00	43	43
(R)	13	4#	4	380	380.00	180.00	52	52
(S)	9	6#	6	270	270.00	60.00	60	60
(T)	19	10#	10	522	522.00	90.00	56	56
(U)	5	1#	1	200	200.00	40.00	66	66
(V)	12	2#	2	350	350.00	100.00	67	67
(X)	11	3#	3	424	424.00	88.00	69	69

Custos > Indiretos: 1440 Diretos: 6589.94 Projeto: 8029.94 Prazo: 72

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	65	7	
ABHIMNQSUVX	65	7	
ABHIOVX	37	35	
ACGHIMNPRTUVX	72	0	==> Crítico
ACGHIMNQSUVX	72	0	==> Crítico
ACGHIOVX	44	28	
ACGJMNPTUVX	69	3	
ACGJMNSUVX	69	3	
ADGHIMNPRTUVX	72	0	==> Crítico
ADGHIMNQSUVX	72	0	==> Crítico
ADGHIOVX	44	28	
ADGJMNPRTUVX	69	3	
ADGJMNSUVX	69	3	
AEFKLMNPRTUVX	72	0	==> Crítico
AEFKLMNQSUVX	72	0	==> Crítico
AEGHIMNPRTUVX	72	0	==> Crítico
AEGHIMNQSUVX	72	0	==> Crítico
AEGHIOVX	44	28	
AEGJMNPRTUVX	69	3	
AEGJMNSUVX	69	3	

Acelerando Conjunto { N } em 2 unidades de tempo - (Iteração: 22)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	7	4	1240	650.87	30.87	26	26
(N)	23	8#	8	380	380.00	120.00	33	33
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	9#	9	260	260.00	80.00	41	41
(Q)	17	17	7	1020	900.00	12.00	41	41
(R)	13	4#	4	380	380.00	180.00	50	50
(S)	9	6#	6	270	270.00	60.00	58	58
(T)	19	10#	10	522	522.00	90.00	54	54
(U)	5	1#	1	200	200.00	40.00	64	64
(V)	12	2#	2	350	350.00	100.00	65	65
(X)	11	3#	3	424	424.00	88.00	67	67

Custos > Indiretos: 1400 Diretos: 6693.70 Projeto: 8093.70 Prazo: 70

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	63	7	
ABHIMNQSUVX	63	7	
ABHIOVX	37	33	
ACGHIMNPRTUVX	70	0	==> Crítico
ACGHIMNQSUVX	70	0	==> Crítico
ACGHIOVX	44	26	
ACGJMNPTUVX	67	3	
ACGJMNSUVX	67	3	
ADGHIMNPRTUVX	70	0	==> Crítico
ADGHIMNQSUVX	70	0	==> Crítico
ADGHIOVX	44	26	
ADGJMNPRTUVX	67	3	
ADGJMNSUVX	67	3	
AEFKLMNPRTUVX	70	0	==> Crítico
AEFKLMNQSUVX	70	0	==> Crítico
AEGHIMNPRTUVX	70	0	==> Crítico
AEGHIMNQSUVX	70	0	==> Crítico
AEGHIOVX	44	26	
AEGJMNPRTUVX	67	3	
AEGJMNSUVX	67	3	

Acelerando Conjunto { M } em 3 unidades de tempo - (Iteração: 23)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	5	1	290	190.00	25.00	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	3	1	420	333.53	13.53	23	23
(M)	24	4#	4	1240	1240.00	620.00	26	26
(N)	23	8#	8	380	380.00	120.00	30	30
(O)	13	13	3	570	410.00	16.00	26	26
(P)	14	9#	9	260	260.00	80.00	38	38
(Q)	17	17	7	1020	900.00	12.00	38	38
(R)	13	4#	4	380	380.00	180.00	47	47
(S)	9	6#	6	270	270.00	60.00	55	55
(T)	19	10#	10	522	522.00	90.00	51	51
(U)	5	1#	1	200	200.00	40.00	61	61
(V)	12	2#	2	350	350.00	100.00	62	62
(X)	11	3#	3	424	424.00	88.00	64	64

Custos > Indiretos: 1340 Diretos: 7282.83 Projeto: 8622.83 Prazo: 67

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	60	7	
ABHIMNQSUVX	60	7	
ABHIOVX	37	30	
ACGHIMNPRTUVX	67	0	==> Crítico
ACGHIMNQSUVX	67	0	==> Crítico
ACGHIOVX	44	23	
ACGJMNPRTUVX	64	3	
ACGJMNSUVX	64	3	
ADGHIMNPRTUVX	67	0	==> Crítico
ADGHIMNQSUVX	67	0	==> Crítico
ADGHIOVX	44	23	
ADGJMNPRTUVX	64	3	
ADGJMNSUVX	64	3	
AEFKLMNPRTUVX	67	0	==> Crítico
AEFKLMNQSUVX	67	0	==> Crítico
AEGHIMNPRTUVX	67	0	==> Crítico
AEGHIMNQSUVX	67	0	==> Crítico
AEGHIOVX	44	23	
AEGJMNPRTUVX	64	3	
AEGJMNSUVX	64	3	

Acelerando Conjunto { I L } em 2 unidades de tempo - (Iteração: 24)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	3	1	290	203.53	13.53	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	5	3	280	193.53	13.53	18	18
(L)	11	1#	1	420	420.00	100.00	23	23
(M)	24	4#	4	1240	1240.00	620.00	24	24
(N)	23	8#	8	380	380.00	120.00	28	28
(O)	13	13	3	570	410.00	16.00	24	24
(P)	14	9#	9	260	260.00	80.00	36	36
(Q)	17	17	7	1020	900.00	12.00	36	36
(R)	13	4#	4	380	380.00	180.00	45	45
(S)	9	6#	6	270	270.00	60.00	53	53
(T)	19	10#	10	522	522.00	90.00	49	49
(U)	5	1#	1	200	200.00	40.00	59	59
(V)	12	2#	2	350	350.00	100.00	60	60
(X)	11	3#	3	424	424.00	88.00	62	62

Custos > Indiretos: 1300 Diretos: 7382.83 Projeto: 8682.83 Prazo: 65

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	58	7	
ABHIMNQSUVX	58	7	
ABHIOVX	35	30	
ACGHIMNPRTUVX	65	0	==> Crítico
ACGHIMNQSUVX	65	0	==> Crítico
ACGHIOVX	42	23	
ACGJMNPTUVX	64	1	
ACGJMNSUVX	64	1	
ADGHIMNPRTUVX	65	0	==> Crítico
ADGHIMNQSUVX	65	0	==> Crítico
ADGHIOVX	42	23	
ADGJMNPRTUVX	64	1	
ADGJMNSUVX	64	1	
AEFKLMNPRTUVX	65	0	==> Crítico
AEFKLMNQSUVX	65	0	==> Crítico
AEGHIMNPRTUVX	65	0	==> Crítico
AEGHIMNQSUVX	65	0	==> Crítico
AEGHIOVX	42	23	
AEGJMNPRTUVX	64	1	
AEGJMNSUVX	64	1	

Acelerando Conjunto { I K } em 1 unidade de tempo - (Iteração: 25)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	2	1	290	226.79	36.79	21	21
(J)	7	7	3	84	80.00	1.00	16	16
(K)	8	4	3	280	216.79	36.79	18	18
(L)	11	1#	1	420	420.00	100.00	22	22
(M)	24	4#	4	1240	1240.00	620.00	23	23
(N)	23	8#	8	380	380.00	120.00	27	27
(O)	13	13	3	570	410.00	16.00	23	23
(P)	14	9#	9	260	260.00	80.00	35	35
(Q)	17	17	7	1020	900.00	12.00	35	35
(R)	13	4#	4	380	380.00	180.00	44	44
(S)	9	6#	6	270	270.00	60.00	52	52
(T)	19	10#	10	522	522.00	90.00	48	48
(U)	5	1#	1	200	200.00	40.00	58	58
(V)	12	2#	2	350	350.00	100.00	59	59
(X)	11	3#	3	424	424.00	88.00	61	61

Custos > Indiretos: 1280 Diretos: 7429.34 Projeto: 8709.34 Prazo: 64

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	57	7	
ABHIMNQSUVX	57	7	
ABHIOVX	34	30	
ACGHIMNPRTUVX	64	0	==> Crítico
ACGHIMNQSUVX	64	0	==> Crítico
ACGHIOVX	41	23	
ACGJMNPTUVX	64	0	==> Crítico
ACGJMNSUVX	64	0	==> Crítico
ADGHIMNPRTUVX	64	0	==> Crítico
ADGHIMNQSUVX	64	0	==> Crítico
ADGHIOVX	41	23	
ADGJMNPTUVX	64	0	==> Crítico
ADGJMNSUVX	64	0	==> Crítico
AEFKLMNPRTUVX	64	0	==> Crítico
AEFKLMNQSUVX	64	0	==> Crítico
AEGHIMNPRTUVX	64	0	==> Crítico
AEGHIMNQSUVX	64	0	==> Crítico
AEGHIOVX	41	23	
AEGJMNPTUVX	64	0	==> Crítico
AEGJMNSUVX	64	0	==> Crítico

Acelerando Conjunto { I J K } em 1 unidade de tempo - (Iteração: 26)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	8	6	430	244.10	29.10	10	10
(G)	8	6	5	96	92.21	2.21	10	10
(H)	11	5#	5	132	132.00	12.00	16	16
(I)	5	1#	1	290	290.00	100.00	21	21
(J)	7	6	3	84	80.20	0.20	16	16
(K)	8	3#	3	280	280.00	100.00	18	18
(L)	11	1#	1	420	420.00	100.00	21	21
(M)	24	4#	4	1240	1240.00	620.00	22	22
(N)	23	8#	8	380	380.00	120.00	26	26
(O)	13	13	3	570	410.00	16.00	22	22
(P)	14	9#	9	260	260.00	80.00	34	34
(Q)	17	17	7	1020	900.00	12.00	34	34
(R)	13	4#	4	380	380.00	180.00	43	43
(S)	9	6#	6	270	270.00	60.00	51	51
(T)	19	10#	10	522	522.00	90.00	47	47
(U)	5	1#	1	200	200.00	40.00	57	57
(V)	12	2#	2	350	350.00	100.00	58	58
(X)	11	3#	3	424	424.00	88.00	60	60

Custos > Indiretos: 1260 Diretos: 7555.96 Projeto: 8815.96 Prazo: 63

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	56	7	
ABHIMNQSUVX	56	7	
ABHIOVX	33	30	
ACGHIMNPRTUVX	63	0	==> Crítico
ACGHIMNQSUVX	63	0	==> Crítico
ACGHIOVX	40	23	
ACGJMNPTUVX	63	0	==> Crítico
ACGJMNSUVX	63	0	==> Crítico
ADGHIMNPRTUVX	63	0	==> Crítico
ADGHIMNQSUVX	63	0	==> Crítico
ADGHIOVX	40	23	
ADGJMNPTUVX	63	0	==> Crítico
ADGJMNSUVX	63	0	==> Crítico
AEFKLMNPRTUVX	63	0	==> Crítico
AEFKLMNQSUVX	63	0	==> Crítico
AEGHIMNPRTUVX	63	0	==> Crítico
AEGHIMNQSUVX	63	0	==> Crítico
AEGHIOVX	40	23	
AEGJMNPTUVX	63	0	==> Crítico
AEGJMNSUVX	63	0	==> Crítico

Acelerando Conjunto { F G } em 1 unidade de tempo - (Iteração: 27)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	5	2	250	173.98	3.98	5	5
(D)	5	5	2	72	60.00	4.00	5	5
(E)	12	5	3	198	143.53	8.53	5	5
(F)	21	7	6	430	294.09	79.09	10	10
(G)	8	5#	5	96	96.00	6.00	10	10
(H)	11	5#	5	132	132.00	12.00	15	15
(I)	5	1#	1	290	290.00	100.00	20	20
(J)	7	6	3	84	80.20	0.20	15	15
(K)	8	3#	3	280	280.00	100.00	17	17
(L)	11	1#	1	420	420.00	100.00	20	20
(M)	24	4#	4	1240	1240.00	620.00	21	21
(N)	23	8#	8	380	380.00	120.00	25	25
(O)	13	13	3	570	410.00	16.00	21	21
(P)	14	9#	9	260	260.00	80.00	33	33
(Q)	17	17	7	1020	900.00	12.00	33	33
(R)	13	4#	4	380	380.00	180.00	42	42
(S)	9	6#	6	270	270.00	60.00	50	50
(T)	19	10#	10	522	522.00	90.00	46	46
(U)	5	1#	1	200	200.00	40.00	56	56
(V)	12	2#	2	350	350.00	100.00	57	57
(X)	11	3#	3	424	424.00	88.00	59	59

Custos > Indiretos: 1240 Diretos: 7609.75 Projeto: 8849.75 Prazo: 62

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	56	6	
ABHIMNQSUVX	56	6	
ABHIOVX	33	29	
ACGHIMNPRTUVX	62	0	==> Crítico
ACGHIMNQSUVX	62	0	==> Crítico
ACGHIOVX	39	23	
ACGJMNPTUVX	62	0	==> Crítico
ACGJMNSUVX	62	0	==> Crítico
ADGHIMNPRTUVX	62	0	==> Crítico
ADGHIMNQSUVX	62	0	==> Crítico
ADGHIOVX	39	23	
ADGJMNPTUVX	62	0	==> Crítico
ADGJMNSUVX	62	0	==> Crítico
AEFKLMNPRTUVX	62	0	==> Crítico
AEFKLMNQSUVX	62	0	==> Crítico
AEGHIMNPRTUVX	62	0	==> Crítico
AEGHIMNQSUVX	62	0	==> Crítico
AEGHIOVX	39	23	
AEGJMNPTUVX	62	0	==> Crítico
AEGJMNSUVX	62	0	==> Crítico

Acelerando Conjunto { C D E } em 2 unidades de tempo - (Iteração: 28)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	5	3	300	178.95	18.95	0	0
(B)	4	4	1	146	125.00	7.00	5	5
(C)	10	3	2	250	199.43	29.43	5	5
(D)	5	3	2	72	64.41	4.41	5	5
(E)	12	3#	3	198	198.00	63.00	5	5
(F)	21	7	6	430	294.09	79.09	8	8
(G)	8	5#	5	96	96.00	6.00	8	8
(H)	11	5#	5	132	132.00	12.00	13	13
(I)	5	1#	1	290	290.00	100.00	18	18
(J)	7	6	3	84	80.20	0.20	13	13
(K)	8	3#	3	280	280.00	100.00	15	15
(L)	11	1#	1	420	420.00	100.00	18	18
(M)	24	4#	4	1240	1240.00	620.00	19	19
(N)	23	8#	8	380	380.00	120.00	23	23
(O)	13	13	3	570	410.00	16.00	19	19
(P)	14	9#	9	260	260.00	80.00	31	31
(Q)	17	17	7	1020	900.00	12.00	31	31
(R)	13	4#	4	380	380.00	180.00	40	40
(S)	9	6#	6	270	270.00	60.00	48	48
(T)	19	10#	10	522	522.00	90.00	44	44
(U)	5	1#	1	200	200.00	40.00	54	54
(V)	12	2#	2	350	350.00	100.00	55	55
(X)	11	3#	3	424	424.00	88.00	57	57

Custos > Indiretos: 1200 Diretos: 7694.08 Projeto: 8894.08 Prazo: 60

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	56	4	
ABHIMNQSUVX	56	4	
ABHIOVX	33	27	
ACGHIMNPRTUVX	60	0	==> Crítico
ACGHIMNQSUVX	60	0	==> Crítico
ACGHIOVX	37	23	
ACGJMNPTUVX	60	0	==> Crítico
ACGJMNSUVX	60	0	==> Crítico
ADGHIMNPRTUVX	60	0	==> Crítico
ADGHIMNQSUVX	60	0	==> Crítico
ADGHIOVX	37	23	
ADGJMNPTUVX	60	0	==> Crítico
ADGJMNSUVX	60	0	==> Crítico
AEFKLMNPRTUVX	60	0	==> Crítico
AEFKLMNQSUVX	60	0	==> Crítico
AEGHIMNPRTUVX	60	0	==> Crítico
AEGHIMNQSUVX	60	0	==> Crítico
AEGHIOVX	37	23	
AEGJMNPTUVX	60	0	==> Crítico
AEGJMNSUVX	60	0	==> Crítico

Acelerando Conjunto { A } em 2 unidades de tempo - (Iteração: 29)

Atv	D_Nor	D_Atu	D_Acel	C_Acel	C_Atu	C_Marg	Cedo	Tarde
(A)	10	3#	3	300	300.00	140.00	0	0
(B)	4	4	1	146	125.00	7.00	3	3
(C)	10	3	2	250	199.43	29.43	3	3
(D)	5	3	2	72	64.41	4.41	3	3
(E)	12	3#	3	198	198.00	63.00	3	3
(F)	21	7	6	430	294.09	79.09	6	6
(G)	8	5#	5	96	96.00	6.00	6	6
(H)	11	5#	5	132	132.00	12.00	11	11
(I)	5	1#	1	290	290.00	100.00	16	16
(J)	7	6	3	84	80.20	0.20	11	11
(K)	8	3#	3	280	280.00	100.00	13	13
(L)	11	1#	1	420	420.00	100.00	16	16
(M)	24	4#	4	1240	1240.00	620.00	17	17
(N)	23	8#	8	380	380.00	120.00	21	21
(O)	13	13	3	570	410.00	16.00	17	17
(P)	14	9#	9	260	260.00	80.00	29	29
(Q)	17	17	7	1020	900.00	12.00	29	29
(R)	13	4#	4	380	380.00	180.00	38	38
(S)	9	6#	6	270	270.00	60.00	46	46
(T)	19	10#	10	522	522.00	90.00	42	42
(U)	5	1#	1	200	200.00	40.00	52	52
(V)	12	2#	2	350	350.00	100.00	53	53
(X)	11	3#	3	424	424.00	88.00	55	55

Custos > Indiretos: 1160 Diretos: 7815.14 Projeto: 8975.14 Prazo: 58

Caminhos	Duracoes	Folgas	
ABHIMNPRTUVX	54	4	
ABHIMNQSUVX	54	4	
ABHIOVX	31	27	
ACGHIMNPRTUVX	58	0	==> Crítico
ACGHIMNQSUVX	58	0	==> Crítico
ACGHIOVX	35	23	
ACGJMNPTUVX	58	0	==> Crítico
ACGJMNSUVX	58	0	==> Crítico
ADGHIMNPRTUVX	58	0	==> Crítico
ADGHIMNQSUVX	58	0	==> Crítico
ADGHIOVX	35	23	
ADGJMNPTUVX	58	0	==> Crítico
ADGJMNSUVX	58	0	==> Crítico
AEFKLMNPRTUVX	58	0	==> Crítico
AEFKLMNQSUVX	58	0	==> Crítico
AEGHIMNPRTUVX	58	0	==> Crítico
AEGHIMNQSUVX	58	0	==> Crítico
AEGHIOVX	35	23	
AEGJMNPTUVX	58	0	==> Crítico
AEGJMNSUVX	58	0	==> Crítico