

LAURO BRUNO GUIMARÃES E SILVA

**Utilização de Simulação e Análise de Superfície de
Resposta *Fuzzy* na Minimização de Recursos em Projetos
Não Repetitivos**

Dissertação apresentada ao programa de mestrado em engenharia de produção da escola de engenharia da Universidade Federal de Minas Gerais para a obtenção do título de Mestre em Engenharia de Produção.

Linha de Pesquisa: Engenharia Logística e de Manufatura

Orientador: Professor Doutor Luiz Ricardo Pinto

Belo Horizonte, MG

agosto de 09

SUMÁRIO

LISTA DE FIGURAS	III
LISTA DE TABELAS	IV
LISTA DE ABREVIATURAS.....	V
RESUMO.....	VI
<i>ABSTRACT</i>	VII
1 INTRODUÇÃO.....	1
1.1 JUSTIFICATIVAS.....	1
1.2 OBJETIVO GERAL.....	2
1.3 OBJETIVOS ESPECÍFICOS.....	3
1.4 ESTRUTURA DA DISSERTAÇÃO.....	3
2 REVISÃO DE LITERATURA.....	5
2.1 GERENCIAMENTO DE PROJETOS.....	5
2.1.1 Os Projetos e suas Características Comuns.....	5
2.1.2 Representação de Projetos	6
2.1.3 O Gerenciamento de Projetos	6
2.2 SEQUENCIAMENTO DE PROJETOS SOB INCERTEZA	8
2.2.1 Sequenciamento Reativo e Linha de Base.....	9
2.2.2 Sequenciamento de Projeto Estocástico	11
2.2.3 Sequenciamento de Projeto <i>Fuzzy</i>	11
2.2.4 Análise de Sensibilidade	16
2.3 A SIMULAÇÃO PARA OTIMIZAÇÃO.....	18
2.3.1 Abordagens Correntes para Unir Otimização e Simulação	19
2.3.2 As Particularidades de Simulação para Otimização.....	20
2.3.3 Algoritmos Atualmente Utilizados em Simulação para Otimização	22

2.3.4	Algoritmos Encontrados na Literatura.....	23
2.3.5	Algoritmos Encontrados em Softwares Comerciais	27
2.3.6	Sequenciamento de Tarefas Usando Simulação para Otimização	28
3	METODOLOGIA PROPOSTA	30
3.1	CONTEXTUALIZAÇÃO DO PROBLEMA.....	30
3.2	COLETA DE DADOS.....	31
3.2.1	Definindo a Rede do Projeto.....	31
3.2.2	Estimando Durações e Recursos para as Atividades	33
3.3	DEFININDO OS CENÁRIOS.....	37
3.4	MODELANDO A REDE DO PROJETO PARA O SIMULADOR <i>FUZZY</i>	37
3.5	O SIMULADOR <i>FUZZY</i>	40
3.6	GERANDO SUPERFÍCIES DE RESPOSTA	43
3.7	ANÁLISE DAS SUPERFÍCIES DE RESPOSTA	44
3.8	DETERMINAÇÃO DOS RESULTADOS	44
4	RESULTADOS	45
5	CONCLUSÃO	52
6	SUGESTÕES PARA TRABALHOS FUTUROS.....	54
	REFERÊNCIAS BIBLIOGRÁFICAS	55

LISTA DE FIGURAS

Figura 2.1: Relação entre ciclo de vida, incerteza, aprendizagem e alteração de escopo.....	5
Figura 2.2: Modelo de rede usando notação atividade no arco (I) e atividade no nó (II) (Adaptado, Tavares, 2002).	6
Figura 2.3: Interação de grupos de processos em um projeto (PMBok, pág. 68).....	8
Figura 2.4: Número <i>fuzzy</i> triangular (1-1) e trapezoidal (1-2).....	12
Figura 2.5: Número <i>fuzzy</i> sobrepostos.	13
Figura 2.6: <i>Simulação para otimização</i> : Softwares Comerciais (Adaptado, Fu, 2002).	19
Figura 2.7: <i>Simulação para otimização</i> : Programação Estocástica (Adaptado Fu, 2002).	20
Figura 2.8: <i>Simulação para otimização</i> : Perspectiva Prática (Adaptado Fu, 2002).	21
Figura 2.9: <i>Simulação para otimização</i> : Necessidades Futuras (Adaptado Fu, 2002).....	22
Figura 3.1: Diagrama de Rede do Projeto Proposto	32
Figura 3.2: DCA do Projeto Proposto.....	38
Figura 3.3: Fluxograma que representa o modelo computacional.....	43
Figura 4.1: Superfície de Resposta Considerando Duração D1 do Projeto	45
Figura 4.2: Superfície de Resposta Considerando Duração Dm1 do Projeto	47
Figura 4.3: Superfície de Resposta Considerando Duração Dm2 do Projeto	48
Figura 4.4: Superfície de Resposta Considerando Duração D2 do Projeto	49

LISTA DE TABELAS

Tabela 2.1: Diferentes métodos para geração de sequenciamento sob incerteza (Adaptado, Leus, 2005).....	10
Tabela 2.2: Métodos de Otimização Determinística (Adaptado Fu, 2002).....	23
Tabela 2.3: Simulação para otimização: Softwares Comerciais (Adaptado Fu, 2002).....	27
Tabela 3.1: Passo-a-passo do gerenciamento do tempo.....	31
Tabela 3.2: Duração e Recursos das Atividades do Projeto Proposto.....	36
Tabela 4.1: Duração D1 do Projeto em Função dos Recursos Empregados.....	45
Tabela 4.2: Duração Dm1 do Projeto em Função dos Recursos Empregados.....	46
Tabela 4.3: Duração Dm2 do Projeto em Função dos Recursos Empregados.....	47
Tabela 4.4: Duração D2 do Projeto em Função dos Recursos Empregados.....	48

LISTA DE ABREVIATURAS

AoA	– <i>Activity on Arc</i>
AoN	– <i>Activity on Node</i>
CPM	– <i>Critical Path Method</i>
PERT	– <i>Program Evaluation And Review Technique</i>
PMBok	– <i>Project Management Body of Knowledge</i>
PO	– Pesquisa Operacional
PMI	– <i>Project Management Institute</i>
SPO	– <i>Sample Path Optimization</i>
SSD	– Sistema de Suporte a Decisão

Resumo

Uma das principais decisões que o gerente deve tomar no início de um projeto é a determinação da quantidade de recursos de que ele necessitará para conduzir o empreendimento sob sua responsabilidade. A pressão pela redução prazos e pela redução dos custos do projeto, aliada à incerteza inerente ao início da condução dos trabalhos são ingredientes de um problema complexo de decisão. Assim, a determinação da quantidade mínima de recursos necessários para que o projeto seja concluído no menor tempo possível, torna-se um problema complexo quando é considerada a imprecisão na modelagem das durações das atividades de um projeto inédito que por definição não possui dados históricos e, em geral, possui várias atividades em paralelo. A utilização da metodologia proposta neste trabalho, baseada na análise da superfície de resposta *fuzzy*, mostrou-se valiosa na resolução deste problema, inclusive com vantagem em relação à técnica de simulação conhecida como simulação de Monte Carlo.

Palavras-chave: gerenciamento de projetos, simulação por eventos discretos, simulação para otimização, simulação *fuzzy*, superfície de resposta *fuzzy*.

Abstract

One of the main decisions that the manager should take at the beginning of a project is to define the amount of resources it will need to conduct the endeavor under its responsibility. The pressure for cost/time reduction and the uncertainty inherent to initial works are components of a complex decision problem. Therefore, due to the lack of activities duration historical data, this problem becomes harder when several parallel activities are considered at the project network. The methodology proposed in this paper, based on fuzzy response surface analysis, proved to be suitable to solve this kind of problem, even with smaller computational effort than Monte Carlo simulation.

Keywords: project management, discrete event simulation, simulation for optimization, fuzzy simulation, fuzzy response surface.

1 Introdução

1.1 Justificativas

Um fato que exerceu forte influência na ascensão do assunto Gerenciamento de Projetos foi o surgimento da técnica PERT/CPM no final da década de cinquenta. O PERT (do inglês, *Program Evaluation and Review Technique*) surgiu durante a construção de submarinos do programa Polaris da marinha americana, e o CPM (do inglês, *Critical Path Method*) foi desenvolvido pela empresa DuPont. Estas técnicas e suas sucessoras, geralmente conhecidas como PERT/CPM, transformaram-se em um dos pilares da moderna Gerência de Projetos (PRADO, 2004).

Hoje os projetos são fundamentais para as empresas seja no desenvolvimento de novos produtos, serviços e processos ou na melhoria das operações do dia-a-dia. No entanto, segundo o guia PMBoK, guia mundialmente conhecido e responsável pela reunião das melhores práticas do gerenciamento de projetos, os projetos apresentam geralmente resultados diferentes dos esperados.

Para melhorar a qualidade das entregas, reduzir os atrasos e respeitar o orçamento, considerando o ambiente de incerteza no qual os projetos estão inseridos, este trabalho propõe a utilização da experiência do gerente de projeto, aliada a utilização de ferramentas quantitativas, que consideram a incerteza, implementada em sistemas de suporte à decisão (SSD).

Dentro da pesquisa operacional (PO) existem vários modelos, e algoritmos que abordam os problemas do gerenciamento de projetos. Tradicionalmente, a simulação tem sido usada no cálculo da duração do projeto e do custo esperado. No entanto, outras ferramentas da PO também poderiam ser empregadas na construção de um SSD. No capítulo 2, destinado a revisão bibliográfica serão apresentados algumas dessas ferramentas.

Um problema comum encontrado na prática do gerenciamento de projetos é a minimização de recursos e/ou duração do projeto. Comumente, nesses casos, os usuários de softwares de gerenciamento de projeto se valem da “tentativa e erro” para encontrar uma solução viável, usando uma função dos softwares comerciais

conhecida como nivelamento de recursos. Muito tempo pode ser perdido neste processo se a instância do problema for grande. Este fato leva muitas vezes a não utilização de recursos nos cronogramas.

Uma abordagem possível para problemas estocásticos e combinatórios é a simulação para otimização. No entanto, não foi encontrada na literatura nenhuma aplicação dessa técnica em problemas de gerenciamento de projetos. Uma forma de usar essa técnica é a criação de modelos em softwares comerciais de simulação que possuam rotinas de otimização integradas. No entanto, a lentidão também é uma característica verificada nesses softwares já que a velocidade das heurísticas é dependente da determinação dos seguintes parâmetros:

- valores sugeridos (solução inicial viável);
- limites inferiores das variáveis de decisão; e
- limites superiores das variáveis de decisão.

Em manuais de software como o *Optquest for Arena*[®] é inclusive sugerido que o usuário faça retro-alimentações com os resultados obtidos até determinado momento para aumentar a velocidade de processamento. O algoritmo otimizador usado nesse software é o *Scatter Search*, proposto inicialmente por Glover (1977).

Para superar esses problemas, neste trabalho é apresentado um SSD que:

- incorpora as melhores práticas descritas no guia PMBoK;
- considera a experiência do gerente de projetos;
- é capaz de lidar com a imprecisão das estimativas dos dados associados ao prazo;
- não exige procedimento iterativo manual para obtenção de resultados; e
- pode ser adaptado para incorporar conceitos da simulação para otimização.

1.2 Objetivo Geral

O objetivo do trabalho é propor um modelo de simulação por eventos discretos que seja capaz de atender as necessidades de um algoritmo de simulação para otimização e que não necessite de dados históricos para levantar a duração incerta das atividades.

Para alcançar os objetivos relacionados à simulação para otimização será necessário reduzir consideravelmente o tempo de processamento.

Para atender os requisitos de redução do esforço computacional e a ausência de dados históricos será usada a simulação *fuzzy*. Nesse tipo de simulação, que pretende-se unir a teoria dos conjuntos *fuzzy* e ordenação de medidas *fuzzy* em um algoritmo de simulação a eventos discretos, os números *fuzzy* são usados para descrever a incerteza na duração das atividades, refletindo a imprecisão e a subjetividade da estimativa destes valores.

1.3 Objetivos Específicos

Os objetivos específicos são:

- implementar um algoritmo que seja capaz de simular um projeto cuja duração das atividades é representada por números *fuzzy*;
- analisar as superfícies de respostas geradas a partir desta simulação;
- definir a quantidade de recursos mínimos para execução do projeto no menor tempo possível; e
- avaliar a possibilidade de aplicação da simulação *fuzzy* em algoritmos de simulação para otimização de larga escala.

1.4 Estrutura da Dissertação

No capítulo 2 é apresentada a revisão bibliográfica sobre gerenciamento de projetos e ferramentas da pesquisa operacional aplicadas nesse tema. A primeira parte desta revisão consiste de uma breve introdução sobre o gerenciamento de projetos, na qual serão abordados alguns conceitos básicos do tema. Estes conceitos são necessários para um maior entendimento do contexto no qual a ferramenta de PO será aplicada. Na segunda parte, são descritas as linhas de pesquisa atualmente destinadas ao sequenciamento de projetos sob incerteza. Na terceira parte dessa revisão são apresentados conceitos e linhas de pesquisa da simulação para a otimização. E, em seguida, na quarta parte, são apresentadas aplicações de sequenciamento de atividades usando simulação para otimização.

No capítulo 3, o problema estudado será contextualizado e a metodologia proposta para a solução do problema será apresentada. Neste capítulo o projeto proposto é descrito e modelado. Durante esta descrição a metodologia é apresentada. Este capítulo incluirá detalhes do desenvolvimento do simulador *fuzzy*.

No capítulo 4, serão apresentados e discutidos os resultados obtidos a partir da metodologia proposta no capítulo 3. Neste capítulo serão relatadas as dificuldades encontradas, resultados inicialmente esperados e dicas sobre a forma de análise dos resultados. Essa análise de resultados terá como suporte a análise da superfície de resposta *fuzzy*.

No capítulo 5, será apresentada a conclusão geral do trabalho. Nele serão detalhadas as principais vantagens da adoção da metodologia proposta.

No capítulo 6, serão apontados caminhos para desenvolvimento do tema estudado em pesquisas futuras.

Finalmente são apresentadas as referências utilizadas no desenvolvimento desta dissertação.

2 Revisão de Literatura

2.1 Gerenciamento de Projetos

2.1.1 Os Projetos e suas Características Comuns

Projeto é um esforço temporário levado a efeito para criar um produto, serviço ou resultado único (PMI, pág. 5).

Segundo Prado (2004), os projetos apresentam as seguintes características comuns:

- **Ciclo de vida:** durante a sua vida, o projeto progride passando por etapas cujos finais são caracterizados pela entrega de algum produto (*deliverable*);
- **Incerteza:** Em qualquer tipo de projeto existe algum nível de incerteza;
- **Possibilidade de mudanças durante o ciclo de vida do projeto:** devido à incerteza, o projeto sofre alterações em seu escopo em algum momento do ciclo de vida;
- **Aumento de conhecimento com o tempo:** no decorrer do desenvolvimento do projeto, mais se conhece o produto (bem ou serviço) que está sendo produzido e seu processo de produção; e
- **Interfuncionalidade:** projetos geralmente envolvem diversas áreas de uma organização.

A relação qualitativa entre essas características está descrita na figura 2.1.

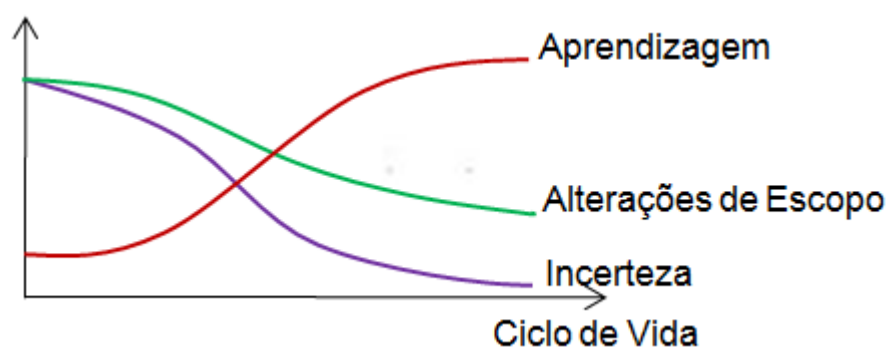


Figura 2.1: Relação entre ciclo de vida, incerteza, aprendizagem e alteração de escopo.

2.1.2 Representação de Projetos

Os dois tipos principais de representação de projetos em redes são o *Activity on Node* (AoN), também conhecido como diagrama de precedência, e o *Activity on Arc* (AoA), também conhecido como diagrama de setas. A figura 2.2 apresenta os dois tipos de representação.

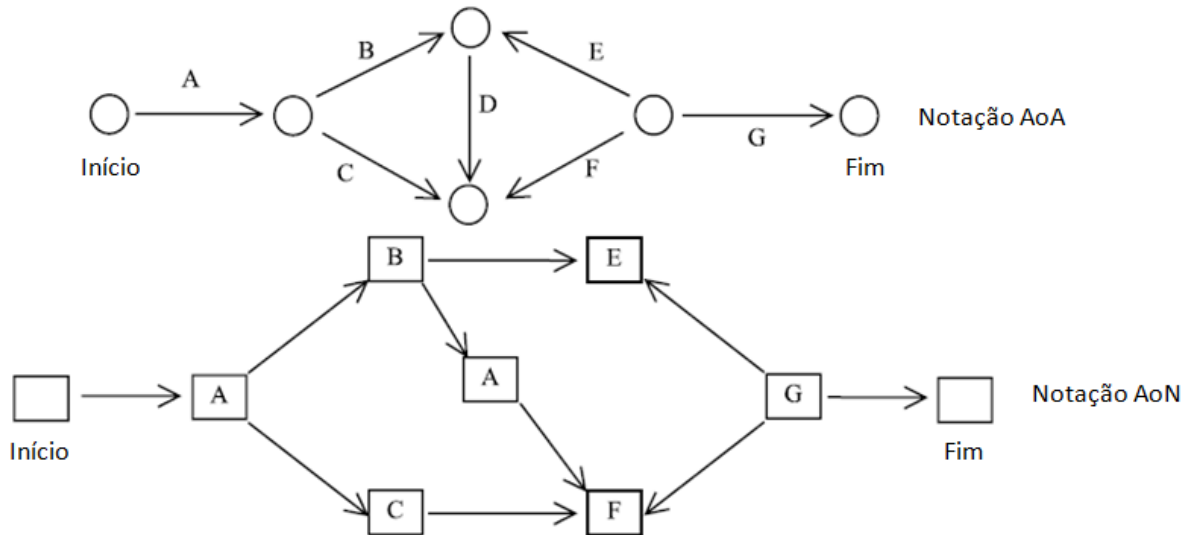


Figura 2.2: Modelo de rede usando notação atividade no arco (I) e atividade no nó (II)
(Adaptado, Tavares, 2002).

2.1.3 O Gerenciamento de Projetos

Segundo o guia PMBoK, o gerenciamento de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus objetivos.

O gerenciamento de projetos é realizado através da aplicação e da integração dos seguintes grupos processos:

- iniciação;
- planejamento;
- execução;
- monitoramento e controle; e

- encerramento.

O grupo de processos de iniciação define e autoriza o projeto ou uma fase dele. Nesse grupo de processos é realizada, por exemplo, a análise de viabilidade. Para que essa análise seja feita, são realizadas definições preliminares de escopo, além de estimativa de prazo e custo.

O grupo de processos de planejamento define e refina os objetivos e planeja a ação necessária para alcançar os objetivos e o escopo para os quais o projeto foi idealizado. Neste grupo de processos são definidas a estratégia de atuação da equipe de projeto e o planejamento das demais fases do trabalho. A partir do planejamento aprovado, pode ser realizado o acompanhamento e o controle do trabalho e de seus resultados. O principal produto dessa etapa é um plano de projeto. Duas das partes desse plano são, por exemplo, o cronograma e a programação dos recursos, foco deste trabalho.

O grupo de processos de execução integra pessoas e outros recursos para realizar o plano do projeto, que foi desenvolvido no grupo de processos do planejamento.

O grupo de processos de monitoramento e controle tem como objetivo a medição regular do progresso do projeto para identificar variações em relação ao plano de gerenciamento do projeto, de forma que possam ser tomadas ações corretivas, quando necessário, para atender aos objetivos do projeto. Essas análises e ações são fundamentais para o bom desempenho do projeto e implicam coletar e distribuir informações para as partes interessadas sobre a situação atual, comparando o realizado com o planejado, além de fazer previsões, estimando o futuro do projeto quanto ao:

- escopo;
- prazo;
- custo;
- qualidade;
- riscos; e
- resultados de negócio.

O grupo de processos de encerramento formaliza a aceitação do produto, serviço ou resultado do projeto. Nesse grupo de processos também é feita uma avaliação do projeto e são armazenadas informações sobre as lições aprendidas no projeto recém executado.

A figura 2.3 exibe a interação, existente nos projetos, entre os grupos de processo mencionados acima. Observando-se essa figura pode-se perceber a superposição dos processos no tempo.

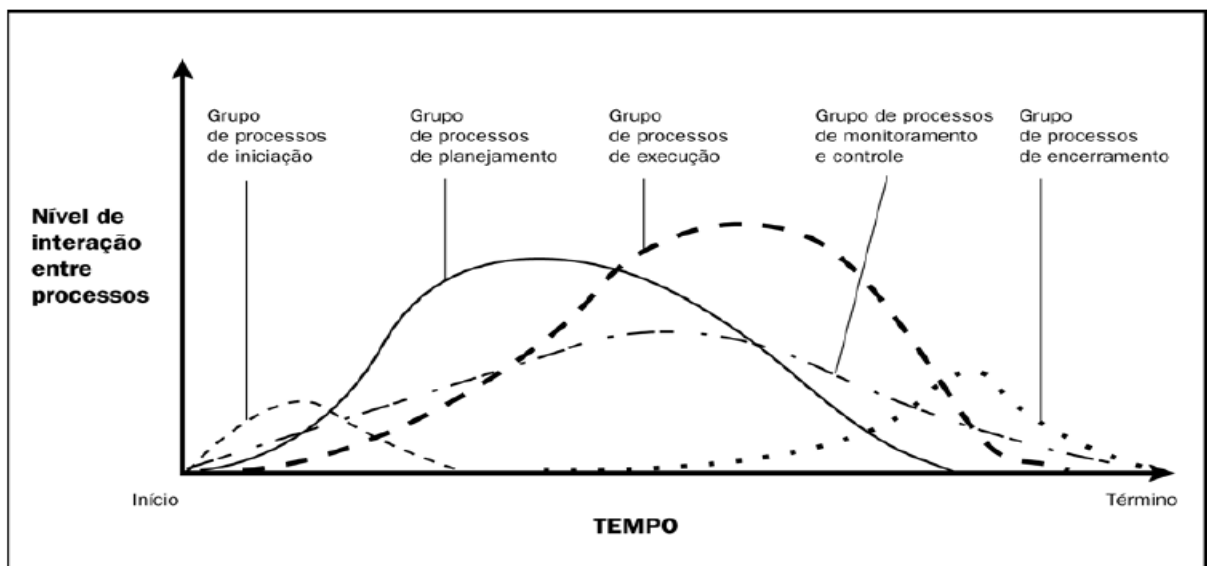


Figura 2.3: Interação de grupos de processos em um projeto (PMBOK, pág. 68).

2.2 Sequenciamento de Projetos sob Incerteza

Segundo Leus (2005) os métodos fundamentais usados no sequenciamento de projeto sob incerteza são:

- sequenciamento reativo;
- sequenciamento de projeto estocástico;
- sequenciamento de projeto *fuzzy*;
- sequenciamento proativo (robusto); e
- sequenciamento via Análise de sensibilidade.

2.2.1 Sequenciamento Reativo e Linha de Base

2.2.1.1 Sequenciamento Reativo

Sequenciamento reativo não tenta lidar com a incerteza no momento da criação da linha de base, em vez disso, ela revisa ou reotimiza a linha de base quando ocorre algum evento esperado. Basicamente, a maior parte do esforço nesse tipo de abordagem reside na “correção” da linha de base (sequenciamento reativo).

Para programar o sequenciamento reativo podem-se usar as seguintes estratégias:

- regra *right shift*;
- ressequenciamento completo; e
- perturbação mínima.

Em um extremo, o esforço reativo se apóia em técnicas muito simples, com o objetivo de restaurar rapidamente a viabilidade do sequenciamento. Esses procedimentos são denominados por Leus (2005) como ações para correção do sequenciamento. Um exemplo típico desse tipo de ação é a regra *right shift*. Segundo essa regra todas as atividades afetadas pelo evento (indisponibilidade de recurso(s) ou inclusão de atividade(s)) serão movidas “para frente” no tempo.

No outro extremo, o sequenciamento reativo pode envolver um sequenciamento completo das atividades restantes a partir de uma solução inviável. Leus (2005) refere a essa abordagem como ressequenciamento completo.

Quando o objetivo é gerar um novo sequenciamento que desvie o mínimo possível do sequenciamento original (*ex post stability*) usa-se a estratégia de perturbação mínima. A implementação dessa estratégia é feita através do uso de algoritmos exatos ou sub-ótimos, em ambos os casos usa-se uma função objetivo de minimização da soma das diferenças absolutas entre a data inicial de cada atividade a ser reagendada e a data inicial original dessas mesmas atividades (El Sakkout e Wallace, 2000).

2.2.1.2 Sequenciamento da Linha de Base

Na tabela 2.1 são relacionadas às estratégias de sequenciamento da linha de base com a forma de lidar com os desvios em relação ao planejamento.

Tabela 2.1: Diferentes métodos para geração de sequenciamento sob incerteza (Adaptado, Leus, 2005).

<i>Sequenciamento da Linha de Base</i>	<i>Durante a Execução do Projeto</i>
(i) Sem Sequenciamento da linha de base	(i) Sequenciamento dinâmico (políticas de Sequenciamento)
(ii) Sequenciamento da linha de base sem antecipação da variabilidade	(ii) Sequenciamento reativo
(iii) Sequenciamento proativo (robusto)	(iii) Ação gerencial
Robustez da qualidade	
Robustez da solução	(iv) Análise de sensibilidade
Flexibilidade	

A primeira coluna da tabela 2.1 apresenta três abordagens básicas diferentes para geração da linha de base. Na primeira abordagem nenhuma linha de base é gerada. Na segunda, a linha de base é gerada usando um método determinístico sem antecipar nenhuma variabilidade nos parâmetros de entrada que devem acontecer durante a execução do projeto. A terceira abordagem visa desenvolver uma linha de base que incorpora um grau de antecipação da variabilidade durante a execução do projeto. Leus (2005) chama esta última abordagem de sequenciamento proativo ou robusto. Este método é alimentado pelas distribuições de probabilidade que regem a duração das atividades e/ou pelas informações sobre o método do sequenciamento reativo que serão obtidas durante a execução do projeto.

No sequenciamento proativo (robusto), o objetivo é minimizar o desvio entre a linha de base e o sequenciamento final (*ex ante stability*) a estratégia é dita robusta. O termo *robustez da qualidade* é frequentemente usado para designar a sensibilidade do desempenho do sequenciamento em termos do valor da função objetivo, enquanto o termo *robustez da solução* se refere à insensibilidade das datas iniciais

das atividades a mudanças nos dados de entrada. A robustez está fortemente relacionada à flexibilidade. Um sequenciamento é considerado flexível se ele pode ser facilmente reparado, ou seja, mudado para uma nova solução de alta qualidade.

A segunda coluna da tabela 2.1 diferencia ações que são tomadas durante a execução do projeto sobre como reagir às “rupturas” no cronograma e quando iniciar novas atividades. No primeiro caso (i) nenhuma linha de base é gerada, mas antes de iniciar um projeto, uma política de sequenciamento é escolhida. Essa política irá determinar como agir durante a execução do projeto. Em (ii), como existe uma linha de base pode ser feito um ressequenciamento nos moldes apresentados no sequenciamento reativo. Em (iii), ao invés de usar as políticas atuais de sequenciamento, a gerência do projeto toma decisões de acordo com o andamento do projeto.

Além desses métodos de construção do sequenciamento final, técnicas foram desenvolvidas para ajudar o gestor do projeto fornecendo informações sobre desvios permissíveis nos parâmetros do projeto. Essas informações ajudam o gestor a determinar quais partes do projeto necessitam de sua maior atenção. Um exemplo dessa abordagem é a análise de sensibilidade (iv).

2.2.2 Sequenciamento de Projeto Estocástico

A literatura para sequenciamento de projeto estocástico tem concentrado seus esforços no problema de sequenciamento de projeto estocástico com recursos limitados. Além desse problema, outros também veem sendo estudados como os problemas de interrupção de atividades estocásticas, *trade-off* tempo/custo e multi-modos estocástico.

2.2.3 Sequenciamento de Projeto Fuzzy

Os defensores do método da duração *fuzzy* das atividades argumentam que as distribuições de probabilidade para as durações da atividade são desconhecidas devido à ausência de dados históricos. Neste caso, no qual a incerteza é substituída pela imprecisão, a literatura recomenda a utilização de números *fuzzy* para modelar

a duração das atividades em vez de números estocásticos. Assim, ao invés de distribuições de probabilidade, são utilizadas funções de pertinência baseada em teoria da possibilidade.

No contexto da simulação, Zhang et al (2005) apresenta como unir a teoria dos conjuntos *fuzzy* e ordenação de medidas *fuzzy* em um algoritmo de simulação a eventos discretos. Este algoritmo tem o objetivo de modelar à incerteza inerente a duração das atividades em projetos reais, especialmente quando amostras de dados são insuficientes ou inexistentes.

Os números *fuzzy* são usados para descrever a incerteza na duração das atividades, refletindo a imprecisão e a subjetividade da estimativa destes valores. Em seu artigo Zhang utiliza os números na forma triangular e trapezoidal conforme mostrado na figura 2.4.

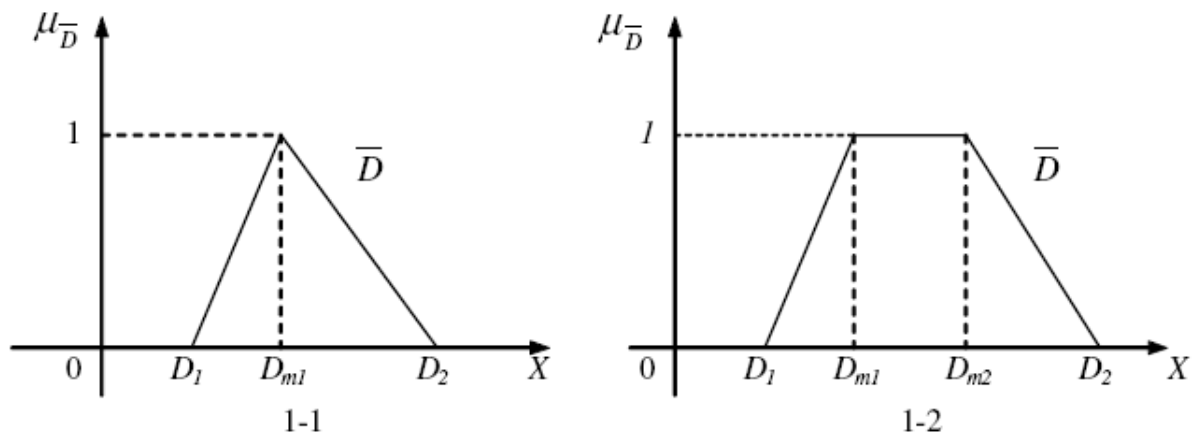


Figura 2.4: Número *fuzzy* triangular (1-1) e trapezoidal (1-2).

Assim a duração das atividades do projeto é definida por sua função de pertinência $\mu_{\bar{D}}(x)$, associada a um valor numérico no intervalo $[0,1]$, isto é:

$$\bar{D} = \{(x, \mu_{\bar{D}}(x) | x \in \bar{D}, \mu_{\bar{D}}(x) \in [0,1])\} \quad (1)$$

Quando a duração da atividade \bar{D} é descrita usando termos linguísticos como, provavelmente entre D_{m1} e D_{m2} , mas definitivamente não menor que D_1 e não maior que D_2 , os números *fuzzy* trapezoidais descrevem bem sua duração. Assim, a função de pertinência para números da forma trapezoidal tem a seguinte forma:

$$\mu_{\bar{D}}(x) = \begin{cases} \frac{x - D_1}{D_{m1} - D_1} & D_1 \leq x \leq D_{m1} & e & D_1 \neq D_{m1} \\ 1, & D_{m1} \leq x \leq D_{m2} \\ \frac{x - D_2}{D_{m2} - D_2} & D_{m2} \leq x \leq D_2 & e & D_{m2} \neq D_2 \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

Analogamente, os números da forma triangular também podem ser usados, já que estes representam um caso particular do trapezoidal no qual $D_{m1} = D_{m2}$.

Para implementar a simulação *fuzzy*, baseada na verificação de atividades (*activity scanning*), Zhang precisava definir qual atividade deveria ser iniciada, ou seja, qual atividade deveria ser executada primeira para avançar o relógio do simulador. Neste momento, ele se deparou com o problema de ordenar números *fuzzy* sobrepostos. Exemplos de números *fuzzy* sobrepostos podem ser observados na figura 2.5.

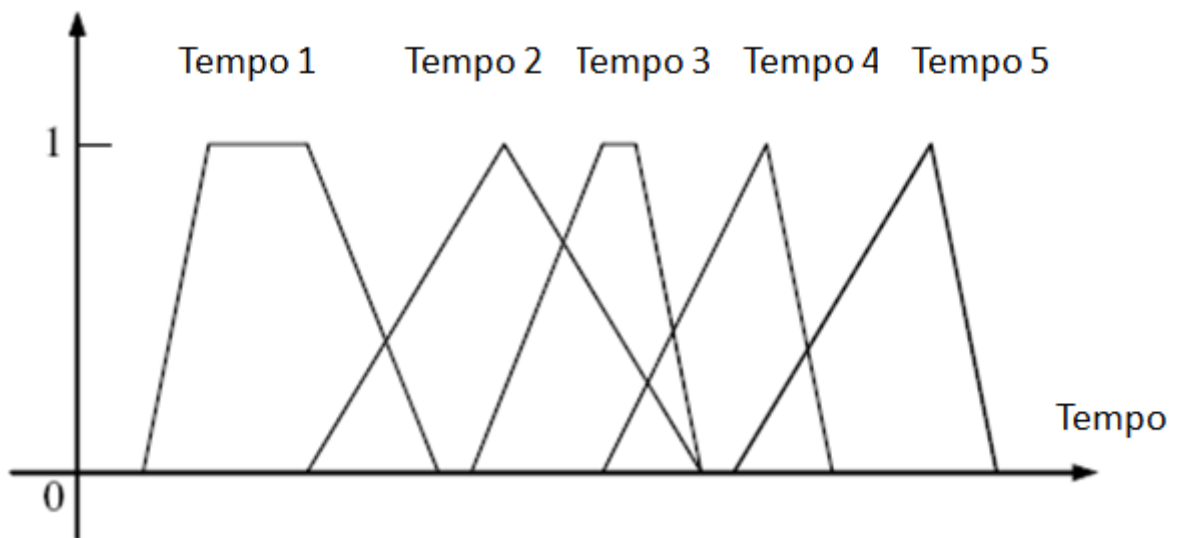


Figura 2.5: Número *fuzzy* sobrepostos.

Para resolver este problema ele usou a metodologia de ordenação proposta por Tran e Duzkstein (2002). Esta metodologia é baseada na comparação dos seguintes alvos pré-determinados:

- crisp máximo (Max); e
- crisp mínimo (Min).

A idéia é que um número *fuzzy* é maior se sua distância para o crisp máximo D_{max} é a menor, mas sua distância para o crisp mínimo D_{min} é a maior. Os valores Max e Min são escolhidos segundo a operação abaixo:

$$\begin{aligned} Min &\leq \inf \left(\bigcup_{i=1}^I s(A(i)) \right) \\ Max &\leq \sup \left(\bigcup_{i=1}^I s(A(i)) \right) \end{aligned} \quad (3)$$

Onde $s(A(i))$ é o suporte do número *fuzzy* $A(i)$ a ser ordenado. Baseado na metodologia de ordenação, a fórmula para calcular D_{max} e D_{min} para um número *fuzzy* $A(a_1, a_2, a_3, a_4)$ é apresentada a seguir:

Quando a função peso $f(\alpha) = \alpha$:

$$\begin{aligned} D^2(A, M) &= \left(\frac{a_2 + a_3}{2} - M \right)^2 + \frac{1}{3} \left(\frac{a_2 + a_3}{2} - M \right) \times [(a_4 - a_3) - (a_2 - a_1)] + \frac{2}{3} \left(\frac{a_3 + a_2}{2} \right)^2 \\ &\quad + \frac{1}{9} \left(\frac{a_3 - a_2}{2} \right) [(a_4 - a_3) + (a_2 - a_1)] + \frac{1}{18} [(a_4 - a_3)^2 + (a_2 - a_1)^2] \\ &\quad - \frac{1}{18} [(a_4 - a_3)(a_2 - a_1)] \end{aligned} \quad (4)$$

Quando a função peso $f(\alpha) = 1$:

$$\begin{aligned} D^2(A, M) &= \left(\frac{a_2 + a_3}{2} - M \right)^2 + \frac{1}{2} \left(\frac{a_2 + a_3}{2} - M \right) \times [(a_4 - a_3) - (a_2 - a_1)] + \frac{1}{3} \left(\frac{a_3 + a_2}{2} \right)^2 \\ &\quad + \frac{1}{6} \left(\frac{a_3 - a_2}{2} \right) [(a_4 - a_3) + (a_2 - a_1)] + \frac{1}{9} [(a_4 - a_3)^2 + (a_2 - a_1)^2] \\ &\quad - \frac{1}{9} [(a_4 - a_3)(a_2 - a_1)] \end{aligned} \quad (5)$$

Onde:

- a_1, a_2, a_3 e a_4 representam os quatro parâmetros descritos no número *fuzzy* trapezoidal A ;
- M é tanto Max quanto Min. Logo, $D_{min} = \sqrt{D^2(A, Min)}$ e $D_{max} = \sqrt{D^2(A, Max)}$; e

- $f(\alpha)$ é uma função contínua crescente adotada na fórmula original de Tran e Duzkstein (2002), servindo como uma função peso para o cálculo soma dos pesos das distâncias.

Os valores α em $f(\alpha)$ pertencem ao intervalo 0 e 1. O valor de $f(\alpha)$ é normalmente definido igual a 1 (constante) ou α , dependendo da característica do tomador de decisão. Geralmente, para tomadores de decisão conservadores ou aversos ao risco o valor da função peso $f(\alpha) = \alpha$, já para tomadores de decisão dispostos a correr riscos, o valor da função peso $f(\alpha)$ é igual a 1.

Depois de definida esta regra para comparação entre números *fuzzy* sobrepostos, Zhang realizou análises entre a simulação *fuzzy* e a simulação de Monte Carlo em uma série de experimentos mostrados nas figuras 2.6 e 2.7. Nestas análises foi observado que além dos resultados obtidos por meio da simulação *fuzzy* possuem toda a informação estatística produzida pela simulação de Monte Carlo, eles ainda são obtidos mais rapidamente, uma vez que é necessário apenas um ciclo da simulação para se obter as saídas do modelo.

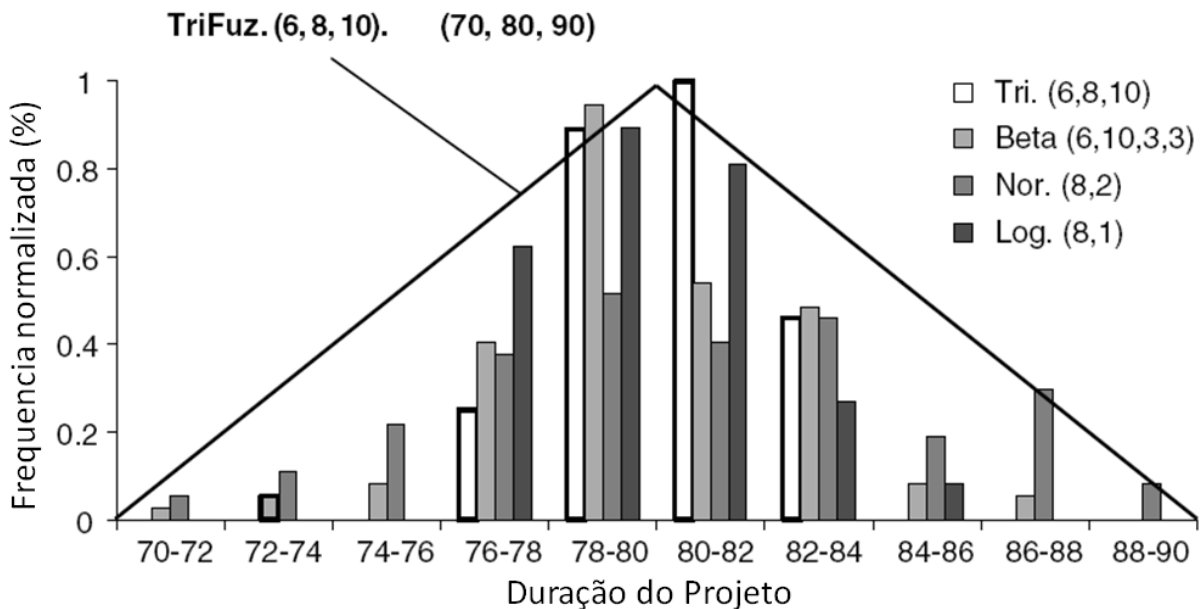


Figura 2.6: Comparação entre os resultados da simulação obtidos a partir de números *fuzzy* triangulares e resultados obtidos a partir de distribuições de probabilidade.

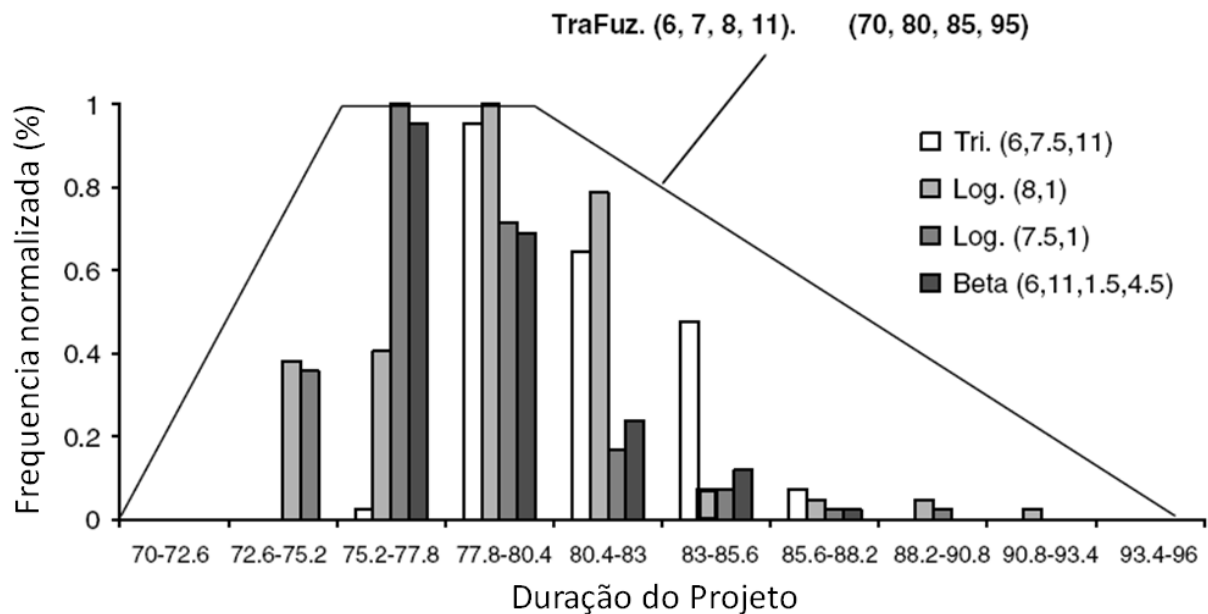


Figura 2.7: Comparação entre os resultados da simulação obtidos a partir de números *fuzzy* trapezoidais e resultados obtidos a partir de distribuições de probabilidade.

2.2.4 Análise de Sensibilidade

A análise de sensibilidade lida com questões do tipo “E se?” (ou análise de cenários) que surgem quando são propostas alterações nos parâmetros. Recentemente, tem-se estudado problemas de sequenciamento de máquina nos quais se busca responder as seguintes questões:

- Quais são os limites para mudança nos parâmetros tal que a solução se mantém ótima?
- Dada uma alteração de parâmetro específica, qual é o novo custo ótimo?
- Dada uma alteração de parâmetro específica, qual é a nova solução ótima?
- Quando uma linha de base mantém-se ótima?
- Quando o valor de uma função objetivo se mantém ótima?
- Quais são os tipos de análise de sensibilidade úteis para avaliar a robustez de soluções ótimas?

- Quais tipos de análises de sensibilidade podem ser executados sem usar os detalhes completos da solução?

Apesar de ser uma análise promissora, segundo Leus (2005), o uso de análise de sensibilidade no contexto do gerenciamento de projetos ainda precisa ser explorado. Dentre essas poucas publicações, destaca-se a aplicação da análise de sensibilidade a variabilidade da duração média feita por Elmaghrabv et al (1999).

2.3 A Simulação para Otimização

Características como, animação, facilidade na construção do modelo e de reutilização, assim como a geração de relatórios têm sido os fatores considerados pelos usuários na compra dos softwares de simulação. Na última década, entretanto, o surgimento de certas áreas de aplicação, como a computação financeira (Consiglio et al, 1999) , a velocidade e qualidade dos cálculos no simulador têm ocupado um papel central na percepção de valor do usuário. Por esse motivo, rotinas de otimização tornaram-se um fator de diferenciação no mercado de softwares de simulação.

A seguir são apresentadas algumas evidências práticas da união entre simulação e otimização. (Fu, 2002):

- atualmente todo pacote de simulação possui algum tipo de otimização;
- recentemente, foram incluídas novas seções referentes a esse assunto em livros didáticos (seção 12.4 em Banks et al. (2000) e seção 12.6 em Law and Kelton (2000)); e
- o termo “*simulation optimization*” está mais conhecido. Ele foi recentemente incluído na *Encyclopedia of Operations Research and Management Science* (Gass e Harris, 2000).

Dois caminhos têm sido tomados para aumentar a qualidade e a eficiência dos cálculos. O primeiro, pesquisado por vários autores, consiste em se aproveitar da estrutura do problema para aumentar a eficiência. O segundo, solução encontrada pelos desenvolvedores de softwares, consiste em incluir algoritmos mais gerais (como métodos genéticos e técnicas do tipo *simulated annealing* com proposto por Alkhamis et al(2002)) capazes de resolver de forma robusta uma grande variedade de problemas. Apesar da segunda opção não fornecer soluções ótimas, a escolha por esses algoritmos, realizada pelos desenvolvedores de softwares, é compreensível.

Essas diferentes abordagens levaram Michael C. Fu em 2002 a escrever o artigo intitulado, *Optimization for Simulation: Theory vs. Practice*. Nele, o autor defende uma abordagem conjunta para as duas partes envolvidas. A partir dessa estrutura,

proposta por Fu (2002), adaptações de antigas técnicas ou mesmo o desenvolvimento de novas técnicas têm contribuído para o desenvolvimento nessa área de pesquisa.

O problema proposto pela otimização para simulação, como em qualquer problema de otimização, é encontrar o melhor conjunto de parâmetros que minimize ou maximize uma função objetivo sujeita a restrições. A função objetivo possui valores que devem ser estimados pela saída de uma simulação estocástica, conforme a equação abaixo:

$$\min_{\theta \in \Theta} J(\theta) = E[L(\theta, \omega)] \quad (6)$$

Sendo, θ o vetor que representa as variáveis de decisão, $J(\theta)$ a função objetivo, Θ o conjunto de restrições, ω representa um *sample path* (replicação da simulação), $L(\theta, \omega)$ a medida de desempenho avaliada e $E[L(\theta, \omega)]$ a esperança dessa medida.

2.3.1 Abordagens Correntes para Unir Otimização e Simulação

A união entre simulação e otimização pode ser realizada de várias formas. Nesta seção serão apresentadas as duas maneiras opostas de como esta união pode ser realizada.

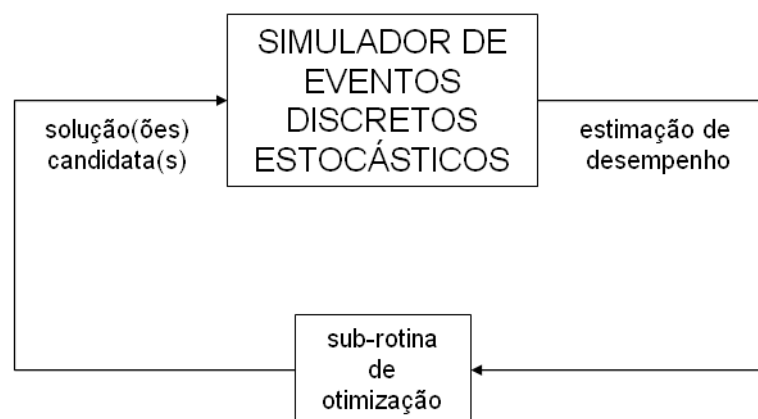


Figura 2.8: *Simulação para otimização: Softwares Comerciais* (Adaptado, Fu, 2002).

Atualmente, nos softwares comerciais, originariamente usados para simulação, não há uma relação de igualdade entre a otimização e a simulação. Neles as rotinas de otimização são apêndices do motor de simulação, conforme mostra a figura 2.8.

Em contraste, desenvolvimentos recentes realizados em programação estocástica apresentam outra forma de abordagem para a união dessas duas técnicas, conforme mostrado na figura 2.9.

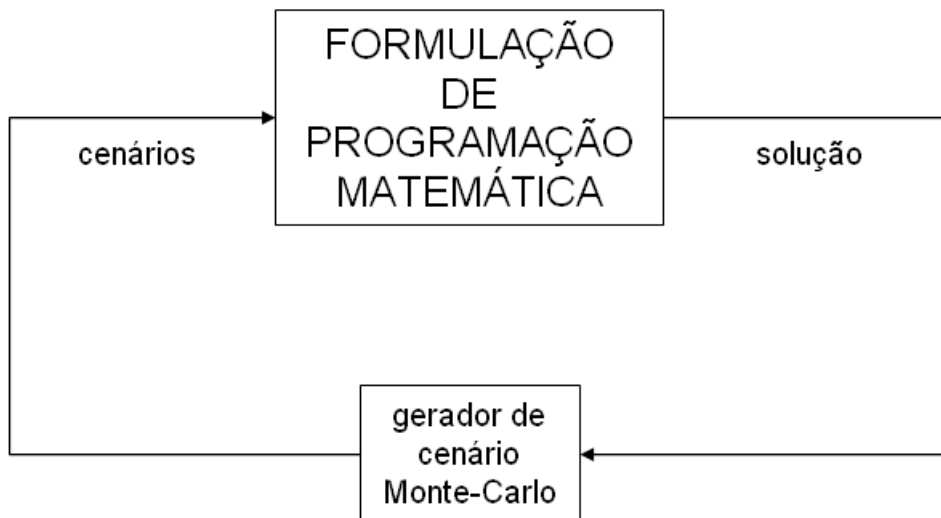


Figura 2.9: Simulação para otimização: Programação Estocástica (Adaptado Fu, 2002).

Nesse caso, a simulação de Monte Carlo é um apêndice que é usado para gerar cenários para as formulações de programação matemática.

2.3.2 As Particularidades de Simulação para Otimização

Segundo Fu (2002), o processo empregado atualmente pode ser dividido em dois passos:

- geração de soluções candidatas; e
- avaliação das soluções.

No ponto de vista da prática, simulação para otimização trabalha mais intensamente com algoritmos que se concentram no primeiro passo. Dessa forma, o modelo de simulação é tratado essencialmente como mais uma função geradora que possui algumas análises estatísticas. O processo de obtenção da solução é mostrado na figura 2.10.

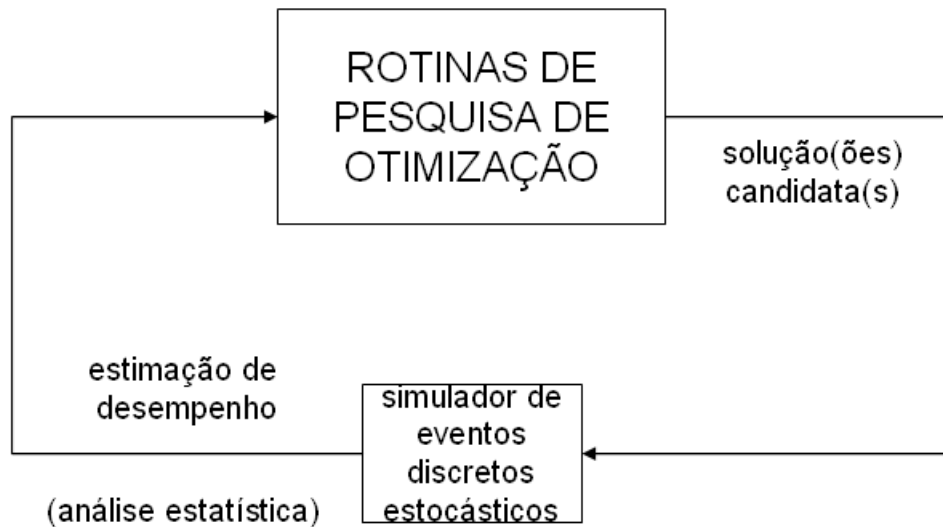


Figura 2.10: Simulação para otimização: Perspectiva Prática (Adaptado Fu, 2002).

No entanto, este processo não trata de forma adequada o universo de simulação para otimização, porque concentra muito na busca ao invés de concentrar na busca e na avaliação das soluções. A principal característica de simulação para otimização, que não é relevante para o universo determinístico, é o *trade-off* entre quantidade de esforço computacional necessário para encontrar mais soluções candidatas *versus* obter uma estimativa melhor das soluções atuais, isto é, encontrar $\arg \min_{\theta \in \Theta} J(\theta)$ *versus* estimar $J(\theta)$.

Outra dificuldade particular de se tratar problemas de simulação para otimização é que as medidas de desempenho de saída são estimadas por meio de simulação estocástica o que despense muito esforço computacional. Em outras palavras, uma replicação de simulação simples pode levar o mesmo tempo que a solução de um problema típico de programação linear (Fu, 2002).

Sendo assim, deve-se concentrar na comparação do desempenho relativo ao invés de estimar o desempenho absoluto. Em outras palavras, o objetivo primordial do processo de busca é ordenar.

Para atender a necessidade de comparação constante das soluções, o processo deveria ser modificado conforme descrito abaixo e mostrado na figura 2.11:

1. Realizar passos iterativos e integrados de pesquisa e comparação até a escolha da solução ótima; e
2. Estimar o valor da função objetivo.

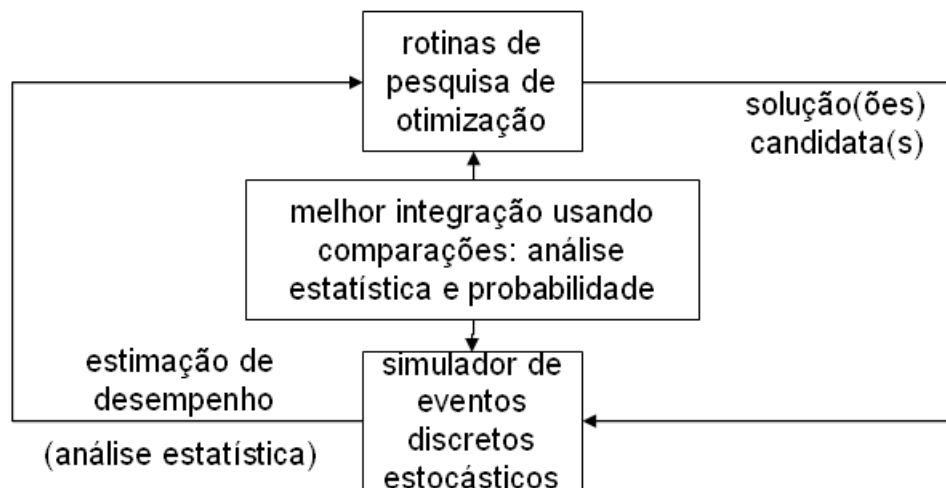


Figura 2.11: *Simulação para otimização: Necessidades Futuras* (Adaptado Fu, 2002).

2.3.3 Algoritmos Atualmente Utilizados em Simulação para Otimização

Atualmente são utilizados alguns algoritmos que tem o objetivo de integrar melhor as duas técnicas. A tabela 2.2 mostra os métodos da otimização determinística que vem sendo adotados em simulação para otimização.

Os métodos, *simulated annealing*, algoritmos genéticos, *scatter search*, busca tabu e redes neurais dominam as aplicações comerciais enquanto os demais métodos apresentados na tabela 2.2 são encontrados em aplicações teóricas.

Tabela 2.2: Métodos de Otimização Determinística (Adaptado Fu, 2002).

<i>Método</i>	<i>Características Principais</i>
<i>Gradient search</i>	Move localmente na direção mais promissora (direção apontada pelo gradiente)
<i>Random search</i>	Move aleatoriamente para um novo ponto. Nenhuma informação é usada na pesquisa.
<i>Simulated annealing</i>	Às vezes, move para uma direção pior para evitar um extremo local.
Algoritmos genéticos e <i>scatter search</i>	Gera novos membros usando operações em atributos dos membros correntes.
Busca tabu	Usa memória (histórico de pesquisa) para evitar movimentos tabu.
Redes neurais	Função de aproximação (não-linear)
Programação matemática	Poderoso arsenal de softwares já testados rigorosamente

2.3.4 Algoritmos Encontrados na Literatura

2.3.4.1 Ranking & Selection e Ordinal Optimization

Conforme apresentado por Law e Kelton (2000), este método usa análise estatística para ordenar o conjunto de alternativas. O método realizada uma avaliação exaustiva do conjunto. Essa avaliação fornece subsídios para a ordenação do conjunto de alternativas.

Fazendo-se uma comparação entre este procedimento e aqueles utilizados pela otimização, observa-se uma diferença crucial. Em vez de realizar uma busca por soluções cada vez melhores em termos absolutos, como fazem os procedimentos de otimização, este procedimento consiste em ordenar as soluções.

Conceitos de probabilidade e estatística são fundamentais neste tipo de procedimento, já que a aleatoriedade está presente em modelos de simulação para otimização.

Dois conceitos importantes, trazidos da probabilidade e estatística, presentes neste procedimento são inseridos pelo usuário. A saber:

- zona de indiferença (nível de precisão); e
- nível de confiança (probabilidade de seleção correta).

Segundo Ho et al. (1992, 2000), a principal idéia por trás deste algoritmo, também chamado *ordinal optimization approach*, é que, é muito mais fácil estimar a ordem relativa do que precisar o valor.

A alocação do esforço computacional para Ordinal Optimization e para Simulação a eventos discretos foi avaliada por Chen et al (2000).

2.3.4.2 Técnicas de Redução de Variância

Para aplicações em simulação para otimização o tempo computacional é algo extremamente precioso. Em simulação, é sabido que técnicas de redução de variância podem substancialmente reduzir o número de replicações, ou seja, o esforço computacional. Por isso, técnicas como amostragem descritiva, proposta por Saliby (1997) devem ser usadas para alcançar este objetivo.

A amostragem descritiva baseia-se na seleção dos valores de entrada necessários e na posterior escolha aleatória da sequência desses valores selecionados. Conforme demonstrado por Saliby (1997), com a utilização dessa técnica pode-se reduzir parte da variabilidade entre replicações, podendo assim, reduzir a necessidade de replicações .

Law e Kelton (2000) descrevem outras técnicas para redução de variância como, por exemplo, *commom random numbers*, variáveis antitéticas e variáveis de controle.

2.3.4.3 Aproximação Estocástica

O algoritmo imita a pesquisa por gradiente da otimização determinística, mas de maneira estatisticamente rigorosa levando em consideração o universo estocástico, no qual o problema a ser resolvido está inserido.

A forma geral do algoritmo assume a seguinte forma iterativa:

$$\theta_{n+1} = \Pi_{\Theta}(\theta_n - a_n \hat{\nabla} J(\theta_n)) \quad (7)$$

Na qual Π_{Θ} denota algum passo para trás que deve ser realizado quando a uma iteração leva para o ponto fora do conjunto delimitado pelas restrições (no pior caso o retorno é para o ponto anterior), a_n é o multiplicador do tamanho do passo, e $\hat{\nabla} J$ é um estimador do gradiente da função objetivo com as respectivas variáveis de decisão.

Essa técnica foi proposta há quase meio século. Uma aplicação relativamente recente aplicada à simulação para otimização foi feita por Fu (1997).

2.3.4.4 Metodologia da Superfície de Resposta

O objetivo da metodologia da superfície de resposta é obter uma relação funcional aproximada entre as variáveis de entrada e a função objetivo de saída. Quando isso é feito em todo o domínio de interesse, o resultado é conhecido como metamodelo (Barton, 1998). Ele pode ser obtido de várias formas. Os dois caminhos mais comuns têm sido:

- regressão; e
- redes neurais.

Uma vez obtido o metamodelo, poderiam ser aplicados algoritmos otimizadores para obter um valor de máximo ou de mínimo da superfície de resposta. No caso de superfícies lineares ou lineares por partes, por exemplo, poderia ser usado o método

simplex e/ou *branch-and-bound*. Humphrey et al (2000) propôs um algoritmo simplex revisado para otimização de superfície de respostas geradas a partir de simulações.

Quando o objetivo da criação do metamodelo é a aplicação posterior de um modelo determinístico (como é o caso), uma forma de utilização sequencial dessa técnica é indicada.

2.3.4.5 *Random Search Methods*

A parte central deste algoritmo consiste em definir uma estrutura de vizinhança apropriada, que deve ser conectada no sentido matemático. Nesses algoritmos, a partir de um dado ponto são feitos movimentos iterativamente de um ponto para outro de sua vizinhança. Apesar de ser um algoritmo simples, dificuldades de implementação podem ser encontradas para amostrar aleatoriamente a vizinhança com uma distribuição apropriada (Banks *et al* 2000, pág. 495).

A principal vantagem deste método é a sua generalidade e a existência de provas de convergência teórica.

Uma versão simples deste algoritmo foi proposta por Andradóttir (1996). E mais tarde, o mesmo Andradóttir (2002) propõe um algoritmo para aceleração da convergência desse tipo de técnica.

2.3.4.6 *Sample Path Optimization (SPO)*

O SPO, ou otimização das replicações da simulação, busca explorar o poder das ferramentas existentes da otimização determinística para variáveis contínuas.

Seja $\omega = \{\omega_1, \omega_2, \dots, \omega_n, \dots\}$ o conjunto de todos os possíveis *sample paths*, ou replicações da simulação, para $L(\theta, \omega_i)$. Defina a média da amostra com base nos n primeiros *sample paths*:

$$\bar{L}_n(\theta) = \frac{1}{n} L(\theta, \omega_i) \quad (8)$$

Se cada $L(\theta, \omega_i)$ são estimativas não viesadas de $J(\theta)$, então pela lei dos grandes números, tem-se que com probabilidade igual a 1 (um),

$$\bar{L}_n(\theta) \rightarrow J(\theta) \quad (9)$$

SPO otimiza, para um valor de n suficientemente grande, a função determinística \bar{L}_n que aproxima $J(\theta)$.

Conforme Robinson (1996), a maior vantagem deste método é a possibilidade de utilização *a posteriori* do poder das ferramentas desenvolvidas pela otimização determinística para variáveis contínuas.

2.3.5 Algoritmos Encontrados em Softwares Comerciais

A tabela 2.3 apresenta os principais softwares comerciais que possuem métodos de simulação para otimização.

Tabela 2.3: *Simulação para otimização: Softwares Comerciais (Adaptado Fu, 2002)*

<i>Pacote de Otimização</i> <i>(Plataforma de Simulação)</i>	<i>Produtor</i> <i>(URL)</i>	<i>Estratégia de Pesquisa Inicial</i>
AutoStat (AutoMod)	AutoSimulations, Inc. (www.autosim.com)	Algoritmos genéticos e evolucionários.
OptQuest (Arena)	Optimization Technologies, Inc. (www.opttek.com)	<i>Scatter search</i> , busca tabu, e redes neurais
OPTIMIZ (SIMUL8)	Visual Thinking International Ltda. (www.simul8.com)	Redes neurais
SimRunner (ProModel)	PROMODEL Corp. (www.promodel.com)	Algoritmos genéticos e evolucionários.
Optimizer (WITNESS)	Lanner Group, Inc. (www.lanner.com/corporate)	<i>Simulated annealing</i> , busca tabu

Esses métodos usados nos software comerciais não serão detalhados neste trabalho, porque apesar de fornecerem soluções em tempo computacional adequado não possuem garantias de convergência ou otimalidade.

A necessidade crescente de integração das ferramentas de simulação e otimização tem forçado os desenvolvedores de softwares comerciais a implementar rotinas de otimização em seus softwares de simulação. No entanto, essas rotinas atualmente são basicamente heurísticas que não possuem provas de convergência ou garantias de otimalidade.

Existem algoritmos que vêm sendo estudados na literatura que apresentam garantias de convergência e otimalidade, mas são em geral lentos. Os algoritmos mais rápidos são desenvolvidos caso a caso e dependem da estrutura do problema.

Esse descompasso entre teoria e prática, descrito por Fu (2002) e comentado por Glynn (2002) e Kelly (2002) deve ser suplantado pela construção de softwares amigáveis que possuam algoritmos que sejam capazes de resolver rapidamente e com qualidade os problemas propostos pelo mundo real.

As soluções possíveis para este problema são a construção de suplementos para softwares de simulação/otimização já existentes, ou o desenvolvimento de uma nova plataforma que seja capaz de resolver problemas especificamente do tipo simulação para otimização. Apesar da construção de suplementos parecer uma idéia tentadora, os problemas associados à estrutura do software não seriam resolvidos (ver figura 2.8 e 2.9).

2.3.6 Sequenciamento de Tarefas Usando Simulação para Otimização

A simulação para otimização aplicada ao gerenciamento de projetos ainda é incipiente. A única aplicação encontrada que pode ser aplicada em larga escala é de um algoritmo desenvolvido por Lai (2006). Ele desenvolve um novo método *Surrogate Search* para determinar desenhos de sistemas para problemas de grande porte que contém variáveis de decisão combinatórias. Nesse trabalho Lai apresenta

dois exemplos de aplicação. No primeiro ele resolve o problema da designação de operador e no segundo ele trata do sequenciamento de atividades.

3 Metodologia Proposta

3.1 Contextualização do Problema

Dentro do contexto de incerteza e ausência de dados históricos, no qual grande parte dos projetos está inserida, este trabalho tem o objetivo de determinar a quantidade mínima de recursos necessários para que a duração do projeto seja a menor possível.

A metodologia proposta segue os seguintes passos para obtenção dos resultados:

1. Coletar de dados de forma semelhante às melhores práticas descritas no guia PMBoK;
2. Definir os cenários;
3. Modelar a rede do projeto através de um Diagrama de Ciclo de Atividade (DCA), ou qualquer outra forma de modelagem que expresse a lógica do sistema em estudo;
4. Entrar com os modelos propostos no simulador *fuzzy*;
5. Gerar superfícies de resposta;
6. Analisar as superfícies de resposta; e
7. Determinar o valor que minimiza a duração do projeto e os recursos necessários.

Nas próximas seções serão apresentados os passos dessa metodologia, exemplificados por dados de um projeto proposto. A escolha dos dados foi feita para deixar o problema suficientemente simples para que a sua aplicação e análise fosse de simples compreensão. Além do mais, um modelo maior não agregaria mais valor a este trabalho, já que componentes de problemas complexos como o paralelismo já são contemplados nesta instância experimental.

3.2 Coleta de Dados

3.2.1 Definindo a Rede do Projeto

Para levantar os dados serão utilizadas as melhores práticas descritas no guia PMBoK. Para gerenciar o tempo em um projeto o guia apresenta os passos descritos na tabela 3.1.

Tabela 3.1: Passo-a-passo do gerenciamento do tempo.

<i>Item</i>	<i>Passo</i>	<i>Descrição</i>
1	Definição da atividade	Identificação das atividades específicas do cronograma que precisam ser realizadas para produzir as várias entregas do projeto.
2	Relação de sucessão entre as atividades	Identificação e documentação das dependências entre as atividades do cronograma.
3	Estimativa de recursos da atividade	Estimativa do tipo e das quantidades de recursos necessários para realizar cada atividade do cronograma.
4	Estimativa de duração da atividade	Estimativa do número de unidades de tempo que serão necessários para terminar as atividades individuais do cronograma em função dos recursos necessários.
5	Desenvolvimento do cronograma	Sequenciamento propriamente dito. Nele são realizadas análises dos recursos necessários, são consideradas as restrições do cronograma, durações e a relação de precedência das atividades do projeto.
6	Controle do cronograma	Controle das mudanças no cronograma do projeto.

A partir dos passos 1 e 2 da tabela 3.1 é possível construir uma rede do projeto. Para analisar a metodologia proposta neste trabalho será usada a rede mostrada na figura 3.1. O projeto proposto servirá única e exclusivamente para exemplificar o uso da metodologia. Esta metodologia pode ser aplicada a qualquer projeto cujas atividades tenham durações probabilísticas, utilizem recursos e não existam dados históricos suficientes para realização de ajustes de distribuições de probabilidades.

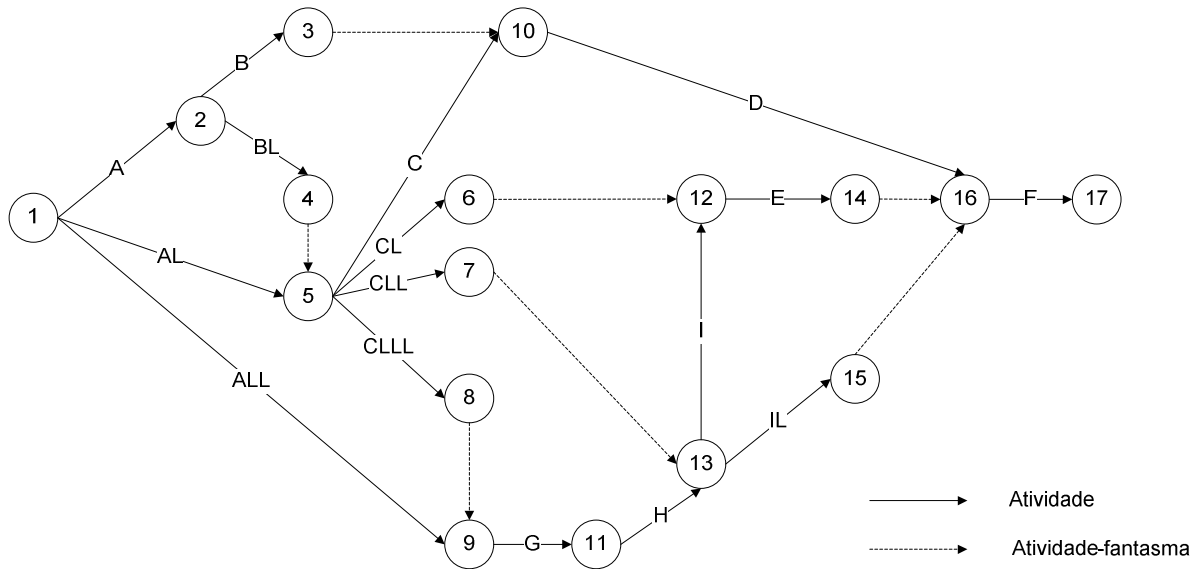


Figura 3.1: Diagrama de Rede do Projeto Proposto

Neste diagrama as atividades estão representadas nos arcos (AoA) e os nós são os eventos que caracterizam o início ou o fim de uma ou mais atividades. Por exemplo, o evento 1 caracteriza o início das atividades A, AL e ALL, enquanto o evento 7 caracteriza o fim da atividade CLL.

No diagrama de rede foi necessária a utilização de seis atividades fantasmas. Essas atividades, que não consomem nem recurso nem tempo, são usadas para representar a relação de dependência entre atividades de um determinado projeto. Por exemplo, as atividades I e IL só podem ser iniciadas se as atividades CLL e H já tiverem sido concluídas.

A primeira vista, pode-se ter a impressão de que foram utilizadas atividades fantasmas em excesso. No entanto, para facilitar a etapa de criação do modelo de simulação foram definidas atividades fantasmas sempre que o início de uma ou mais atividades dependiam da conclusão de mais de uma atividade. Por exemplo, para

iniciar, a atividade F necessita que três (n) atividades sejam finalizadas. Neste caso, duas (n-1) atividades foram definidas como fantasma. Criando a rede dessa forma, a identificação dos pontos nos quais as entidades deverão ser unidas no modelo de simulação é facilitada (isso ficará mais claro na explicação do DCA).

3.2.2 Estimando Durações e Recursos para as Atividades

Na tabela 3.2, os passos 3 e 4 são raramente analisados individualmente dada à dificuldade em estimar a duração de uma atividade sem ter em mente a quantidade de recursos necessários para execução naquele intervalo de tempo.

Para a estimativa da duração da atividade, o guia PMBoK descreve a estimativa de três pontos como melhor prática. Segundo essa prática, a exatidão da estimativa de duração da atividade pode ser aumentada considerando sua variabilidade. Assim, a estimativa de três pontos baseia-se na determinação de três valores estimados:

- mais provável – A duração da atividade é estimada em função de um cenário realista, no qual são consideradas a produtividade esperada dos recursos e a possibilidade de interrupções devido a eventos internos e externos ao projeto;
- otimista – A duração da atividade é estimada em função do cenário mais favorável que poderia ocorrer; e
- pessimista – A duração da atividade baseia-se no pior cenário que poderia ocorrer.

Como orientação, o guia PMBoK sugere o uso de uma média ponderada das durações estimadas para fornecer uma estimativa mais exata da duração da atividade. Assim a duração de uma atividade é calculada seguindo-se a fórmula:

$$Duração\ da\ atividade = \frac{Pessimista + 4 (Mais\ Provável) + Otimista}{6} \quad (10)$$

Assim, a partir dessa fórmula que é baseada na duração por três pontos é definido um valor médio de duração que se aproxima mais do valor mais provável do que dos valores otimistas e pessimistas.

No entanto, dessa forma, ainda seria usado apenas um valor e, por isso, não seria contemplada a variabilidade da duração das atividades. Por este motivo, a partir deste ponto, a metodologia dessa dissertação segue caminhos diferentes dos ditados pelo guia PMBoK para desenvolvimento do cronograma. Estes caminhos podem ser vistos como a base desta dissertação.

A partir desse ponto poderiam ser usadas as seguintes abordagens que se baseiam na simulação:

- simulação de Monte Carlo; e
- simulação *Fuzzy*.

Pela abordagem via teoria da probabilidade deveriam ser coletados dados históricos sobre a duração das atividades e, a partir daí, realizar testes de aderência que seriam usados para identificar a distribuição de probabilidade mais adequada para representar a duração da atividade. Posteriormente, esses dados poderiam ser incluídos em um simulador baseado na simulação de Monte Carlo, como o *Arena*[®]. Em seguida, poderia ser usado um otimizador, como, por exemplo, o *Optquest for Arena*[®], para determinar a quantidade de mínima de recursos que resulta no menor tempo de projeto. No entanto, essa abordagem apresenta os seguintes inconvenientes:

Na coleta de dados:

- ausência de dados históricos em projetos inéditos; e
- arbitrariedade na definição da distribuição de probabilidade, uma vez que não existem dados históricos que garantam a aderência a nenhuma distribuição de probabilidade.

No simulador:

- esforço computacional desnecessário para tratar dados imprecisos se comparado a simulação *fuzzy*; e
- necessidade de replicações e corridas para retornar a saída do simulador para garantir convergência.

No otimizador:

- necessidade de repetir as replicações para cada combinação de recursos determinados pela heurística *Scatter Search* (no caso do *Optquest for Arena*[®]); e
- necessidade de realimentar o sistema com os resultados (quantidades de recursos), para acelerar a obtenção de respostas cada vez melhores (sugestão do manual do *Optquest for Arena*[®]).

Como será mostrado neste trabalho, a abordagem mais adequada para essa situação é a simulação *fuzzy*. Segundo Zhang et al (2005), os principais motivos para essa afirmação são os seguintes:

- não necessita de dados históricos;
- facilidade da verbalização da duração das atividades;
- processo de simulação simplificado dada à imprecisão dos dados coletados;
- simulador rápido e preciso.

Outro fator não apresentado por Zhang et al (2005), mas que é detalhado nesta dissertação é a possibilidade de união com a recente abordagem da simulação para otimização.

Feita a escolha de abordagem, modela-se a natureza imprecisa da duração das atividades dos projetos por números *fuzzy* trapezoidais e/ou triangulares. Esses números serão descritos conforme apresentado por Zhang et al (2005). Assim, a principal diferença entre números *fuzzy* trapezoidais e triangulares é que estes têm as medidas centrais $Dm1$ e $Dm2$ com valores iguais.

Na tabela 3.2 são apresentados as durações e os recursos necessários para execução do projeto proposto, mostrado anteriormente na figura 3.1. Nessa tabela, para cada atividade do projeto são mostradas as durações *fuzzy*, modelada por meio de números *fuzzy* trapezoidais, representados pelo conjunto ($D1$, $Dm1$, $Dm2$ e $D2$), e as quantidades de recursos X e Y associadas. Esses recursos podem ser interpretados como trabalhadores com qualificações diferentes, mas complementares, de modo que para que qualquer atividade seja concluída é

necessária a participação desses dois trabalhadores. Essa suposição de apenas dois trabalhadores pode ser um tanto irreal, no entanto para fins de explicação da metodologia parece ser a abordagem mais adequada. Apesar de essa instância apresentar apenas dois tipos de recursos a metodologia, apresentada aqui, é capaz de lidar com instâncias maiores.

Tabela 3.2: Duração e Recursos das Atividades do Projeto Proposto.

Atividade	Duração Fuzzy				Necessidade de Recurso X (unid.)	Necessidade de Recurso Y (unid.)
	D1	Dm1	Dm2	D2		
A	1,0	1,2	1,3	1,5	1	1
AL	1,1	1,3	1,4	1,5	1	1
ALL	1,9	2,8	3,7	4,6	1	1
B	1,3	1,6	1,9	2,0	1	1
BL	0,5	1,0	2,0	3,0	1	1
C	1,5	2,0	2,5	3,0	1	1
CL	1,0	3,0	5,0	7,0	1	1
CLL	1,1	3,2	5,3	7,4	1	1
CLLL	0,6	0,9	1,1	1,5	1	1
D	1,2	1,4	1,6	1,8	1	1
E	2,0	4,0	6,0	8,0	1	1
F	2,9	3,9	4,9	5,9	1	1
G	5,0	5,1	5,2	5,3	1	1
H	3,0	3,1	3,2	3,3	1	1
I	4,0	4,1	4,2	4,8	1	1
IL	3,2	3,3	3,4	3,6	1	1

A escolha das durações mostradas na tabela 2.1 foi proposital. Como se pode perceber os valores são próximos. Essa proximidade é um fator complicador para o tipo de simulação escolhido uma vez que são necessárias comparações entre números *fuzzy* que podem se sobrepor. Para vencer esse obstáculo, foi utilizado o método de comparação de números *fuzzy* triangulares e trapezoidais (Tran e Duzkstein, 2002).

A escolha de dois tipos de recurso, bem como suas quantidades também foi proposital e feita por questões didáticas. A utilização de dois tipos de recursos proporcionará a análise gráfica da superfície de resposta. A definição das necessidades de recursos com valores iguais proporcionará uma simetria que será útil na análise das superfícies de resposta. Vale lembrar que, poderiam ser usados

outros dados, no entanto esses valores não agregariam valor ao trabalho, uma vez que dificultariam a compreensão da metodologia proposta.

3.3 Definindo os Cenários

Em função do número de atividades paralelas possíveis em um mesmo intervalo de tempo no projeto e da necessidade de recursos X e Y dessas atividades, definiu-se a área de busca da melhor solução. Como o projeto proposto é composto por 16 (dezesseis) atividades, deve-se escolher um valor menor que 16 para a quantidade de recursos. Dessa forma, a área de busca será dada pela combinação das quantidades de recurso associadas a cada tipo de recurso (X e Y).

Assim, os recursos X e Y foram escolhidos para variar no intervalo [1,10], e conseqüentemente, a área de busca é formada pelas 100 combinações (X, Y).

3.4 Modelando a Rede do Projeto para o Simulador *Fuzzy*

Nesta etapa da metodologia, deve-se transformar a rede mostrada na figura 3.1 em um DCA para que o projeto seja inserido no simulador *fuzzy*. O simulador *fuzzy* implementado neste trabalho foi baseado no Método das Três Fases proposto por Tocher (1963) e por isso a modelagem via DCA é a mais conveniente, já que a mesma foi proposta pelo mesmo autor e segue os mesmos princípios do método, facilitando assim a implementação computacional do modelo. Por essas e outras razões que serão explicadas a seguir, foram propostas algumas alterações na forma de modelagem via DCA.

O DCA (com as alterações conceituais propostas) para o projeto é apresentado na figura 3.2. Como pode ser observado, foram intencionalmente ocultadas informações no DCA. Nele:

- não existe legenda de entidades;
- não existem regras para desvio condicional;
- as durações das atividades não são representadas; e
- não são representados os recursos para a realização de cada atividade.

No diagrama não é mostrada uma legenda de entidades, dado que existe apenas uma entidade. Essa é única no início do projeto e vai sendo partida (artifício incluído

O diagrama da figura 3.2 pode ser lido da seguinte forma: no nascedouro, a entidade “projeto” é iniciada e essa entidade passa em seguida para a atividade “Início Dummy” que possui duração zero e não consome nenhum recurso. Após o término dessa atividade, as filas subseqüentes “Aguarda A”, “Aguarda AL” e “Aguarda ALL” recebem cópias da entidade “projeto” (são geradas cópias da entidade). Caso existam recursos suficientes (X e Y) as atividades A, AL e ALL podem então ser iniciadas. Após o término dessas atividades as filas subseqüentes podem receber as entidades (cópias da entidade “projeto”). Assim, nessa sequência “fila-atividade”, o projeto é simulado respeitando se a relação de precedência, a duração *fuzzy* e a necessidade de recursos das atividades.

O parágrafo anterior mostra como as filas são “carregadas” com as entidades e quando uma atividade pode ser iniciada (existência de entidades nas filas anteriores a atividade). No entanto, ainda não foi apresentado como definir o(s) caminho(s) crítico(s) do projeto.

Depois que o projeto foi modelado e simulado no simulador *fuzzy* usando a lógica apresentada no DCA, a única dificuldade para definir o caminho crítico passa a ser a identificação do caminho de folga zero. Para calcular a folga deve-se:

- Armazenar as datas de início e término de cada atividade; e
- Calcular a folga da atividade.

De modo geral, as durações das esperas em fila são uma medida da folga da atividade, ou seja, quanto determinada atividade pode atrasar sem que atrase o projeto como um todo. Essa folga pode ser calculada pela diferença entre a data em que a atividade saiu da fila (ou data de início da atividade sucessora) menos a data que ela chegou na fila (ou data de término da atividade predecessora).

Assim, para identificar o(s) caminho(s) crítico(s) basta medir e ordenar os tempos em fila das entidades da rede. A exceção das atividades *dummy*, para os menores tempos em fila tem-se um indicativo de um caminho crítico.

Em resumo, a lógica é: as entidades que tem maior tempo em fila ficaram esperando a chegada das entidades que estão no caminho crítico, e por isso estas terão tempo em fila nulo.

3.5 O Simulador *Fuzzy*

A implementação do simulador *fuzzy* foi feita em linguagem C++ seguindo o paradigma da orientação a objeto e contemplando as peculiaridades desse tipo de simulação descritas por Zhang et al (2005). Segundo este autor, o simulador *fuzzy* possui as seguintes características principais:

- são inseridas durações *fuzzy* para as atividades;
- não existe geração de números pseudo-aleatórios;
- com apenas um ciclo de simulação já são obtidos os resultados; e
- algoritmo para comparação de números *fuzzy* triangulares e trapezoidais, conforme proposto por Tran e Duzkstein (2002).

Além de incorporar essas características típicas da simulação *fuzzy*, o simulador construído neste trabalho baseia-se no método das três fases.

Conforme descrito por Pinto (2003), esse método é uma das formas de estruturação de programas de simulação a eventos discretos. Seus principais conceitos são:

- entidades;
- fontes/sumidouros;
- filas;
- eventos;
- tipos de eventos;
- relógio de simulação; e
- fases da simulação (A, B e C).

Uma entidade pode ser entendida, no caso da simulação *fuzzy* como um recurso de projeto, ou como o próprio fluxo de trabalho.

Fontes são geradores de entidades e sumidouros são eliminadores de entidades. No caso do simulador *fuzzy* para projetos, a fonte representa o início e o sumidouro o fim do projeto.

Filas são repositórios de entidades.

Um evento pode ser entendido como uma mudança no estado do sistema em processo de simulação. Toda atividade está delimitada por dois eventos, o primeiro que caracteriza seu início e o segundo seu término.

Existem dois tipos de eventos associados ao método das três fases:

- eventos do tipo C, que caracterizam o início da atividade; e
- eventos do tipo B, que caracterizam o fim da atividade.

Todo evento C é também chamado de condicional, porque para uma atividade ter início uma ou mais condições devem ser satisfeitas. A cada atividade está associado um único evento C, que está condicionado à existência de entidades nas filas e/ou fontes que precedem a atividade. Assim, no simulador, atividades só podem ser iniciadas caso existam recursos disponíveis para a execução da atividade.

Os eventos B têm sua ocorrência predeterminada. Isso ocorre porque no método das três fases a duração de cada atividade é determinada antes de seu início. No simulador *fuzzy* os eventos B são definidos executando-se a soma *fuzzy* entre tempo corrente de simulação a duração *fuzzy* da atividade. Por exemplo, a atividade AL foi iniciada em (3; 5; 6; 9), sua duração é (4; 5; 6; 7), logo seu término ocorrerá em (7; 10; 12; 16).

Para cada entidade participante da atividade é definido um evento B distinto. Assim, a cada atividade está associado um único evento C e tantos eventos B quantas forem às entidades que dela participam. O evento B libera as entidades que participaram da atividade a ele relacionada para as filas subsequentes àquela atividade.

O relógio da simulação corresponde ao instante corrente. O relógio sempre avança para o instante em que ocorrerá o evento mais próximo, o que implica em um avanço a intervalos quase sempre variáveis. Sendo assim, o programa de simulação deve dispor de recursos para descobrir quando ocorrerá o próximo evento e avançar o relógio da simulação até ele. No caso do simulador *fuzzy*, a identificação para que o avanço do relógio de simulação possa ser efetuado é função da rotina de comparação de números *fuzzy* proposta por Tran e Duzkstein (2002).

Baseado da descrição de Pinto (2003), as três fases que caracterizam o método são:

- fase A – Corresponde ao avanço do relógio da simulação. Nesta fase o relógio é avançado para o instante quando se dará o término da próxima atividade. Para tanto, o programa consulta a lista de eventos B e encontra, via comparação *fuzzy*, aquele que ocorrerá em primeiro lugar;
- fase B – Termina a atividade e executa todos os eventos B programados para aquele momento, libera os recursos e permite que a atividade sucessora possa ser iniciada. Isso é, adiciona as entidades participantes da atividade nas filas subsequentes; e
- fase C – Executa a verificação das condições de início de todas as atividades do sistema. Inicia as atividades cujas condições foram satisfeitas, calculando suas durações e programando os eventos B associados a elas. O início de cada atividade é caracterizado pela saída das entidades das filas que a precedem. Uma atividade é iniciada quantas vezes sua condição de início for satisfeita.

Para formulação do modelo computacional, baseado no método das três fases, foi construído o algoritmo representado no fluxograma da figura 3.3.

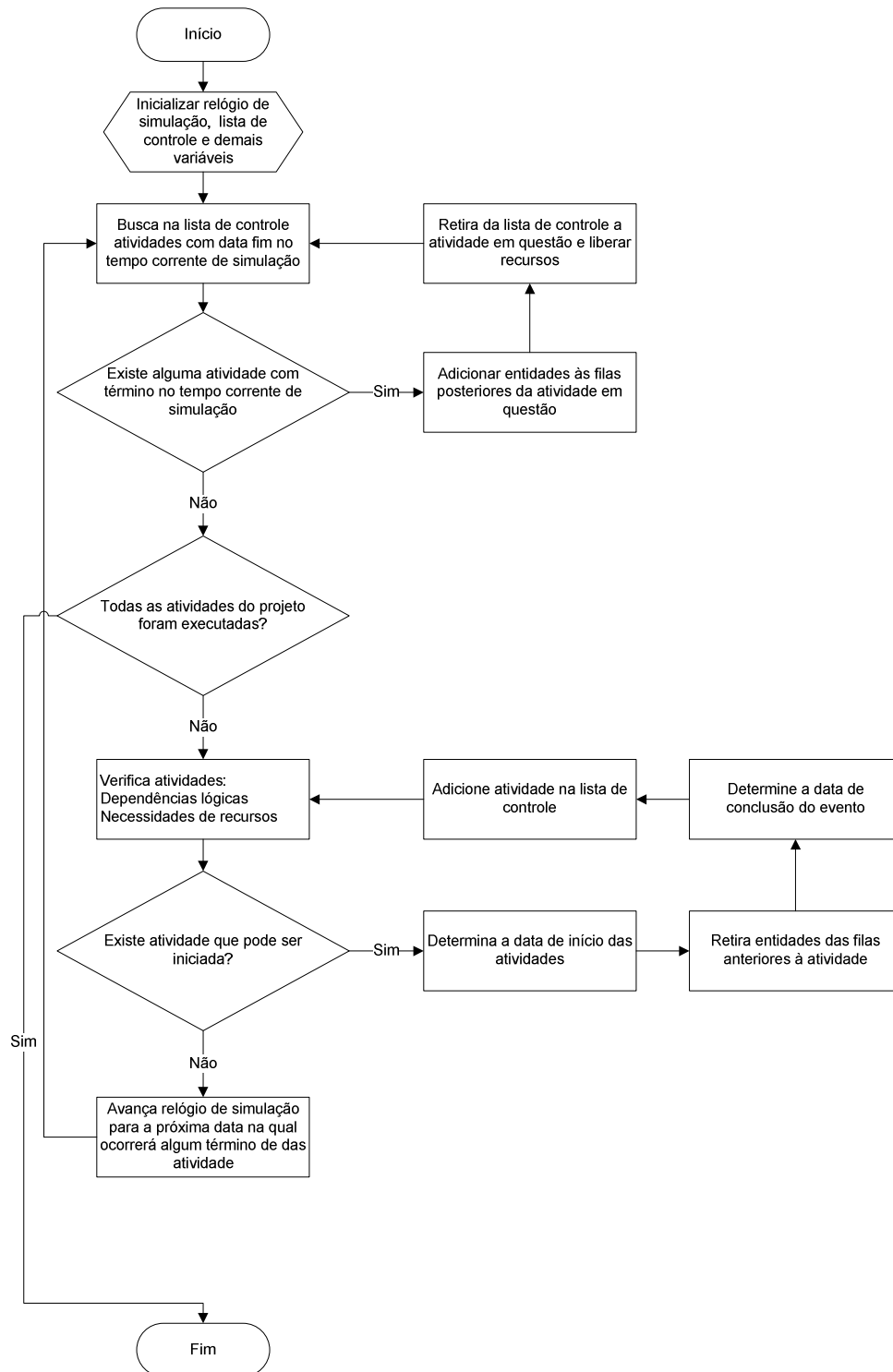


Figura 3.3: Fluxograma que representa o modelo computacional.

3.6 Gerando Superfícies de Resposta

Depois de definido os cenários a serem analisados, ou seja, a combinação de quantidades de cada tipo de recurso (X, Y), o próximo passo é gerar os resultados da simulação que, para a aplicação proposta, é o tempo de execução de projeto.

Deve-se lembrar neste momento que a saída de um simulador *fuzzy* é um número *fuzzy* e que a superfície de resposta é mais complexa que um mero gráfico de três dimensões. Na verdade são geradas quatro superfícies, cada uma delas correspondendo a um dos valores D1, Dm1, Dm2, D2. Essas superfícies formam três camadas capazes de representar as durações dos projetos modelados a partir de números *fuzzy* trapezoidais e/ou triangulares.

Para facilitar a visualização dessas superfícies no capítulo 4, destinado a análise dos resultados, serão gerados gráficos 3D para cada componente do número *fuzzy* (D1, Dm1, Dm2, D2).

3.7 Análise das Superfícies de Resposta

Para analisar as superfícies de respostas geradas a partir de números *fuzzy* devem-se tomar alguns cuidados. A análise isolada de uma superfície pode render conclusões equivocadas. No capítulo 4 serão apresentadas dicas de como evitar esse problema e como realizar uma análise adequada.

3.8 Determinação dos Resultados

Considerando que o problema apresentado neste trabalho é relativamente pequeno e que são considerados apenas dois tipos de recursos X e Y, a determinação dos resultados é obtida diretamente dos gráficos da superfície de resposta. Para situações nas quais exista mais de dois tipos de recursos uma solução gráfica fica inviável. Para encontrar combinação de mínima de recursos que resulta na duração mínima de projeto, basta usar a rotina de comparação de números *fuzzy* proposta por Tran e Duzkstein (2002).

Assim, considerando o problema com dois tipos de recurso, a dificuldade será a identificação do vale mais profundo (resposta da simulação) e seus respectivos valores de X e Y.

4 Resultados

Para facilitar a visualização dessas superfícies foram gerados gráficos de três dimensões para cada componente do número *fuzzy* (D1, Dm1, Dm2, D2). Os resultados são mostrados nas figuras 4.1, 4.2, 4.3 e 4.4 e nas tabelas 4.1, 4.2, 4.3 e 4.4.

Tabela 4.1: Duração D1 do Projeto em Função dos Recursos Empregados

D1		Recursos Y									
		1	2	3	4	5	6	7	8	9	10
Recursos X	1	31,30	31,30	31,30	31,30	31,30	31,30	31,30	31,30	31,30	31,30
	2	31,30	22,60	22,60	22,60	22,60	22,60	22,60	22,60	22,60	22,60
	3	31,30	22,60	20,20	20,20	20,20	20,20	20,20	20,20	20,20	20,20
	4	31,30	22,60	20,20	19,40	19,40	19,40	19,40	19,40	19,40	19,40
	5	31,30	22,60	20,20	19,40	19,80	19,80	19,80	19,80	19,80	19,80
	6	31,30	22,60	20,20	19,40	19,80	19,00	19,00	19,00	19,00	19,00
	7	31,30	22,60	20,20	19,40	19,80	19,00	19,00	19,00	19,00	19,00
	8	31,30	22,60	20,20	19,40	19,80	19,00	19,00	19,00	19,00	19,00
	9	31,30	22,60	20,20	19,40	19,80	19,00	19,00	19,00	19,00	19,00
	10	31,30	22,60	20,20	19,40	19,80	19,00	19,00	19,00	19,00	19,00

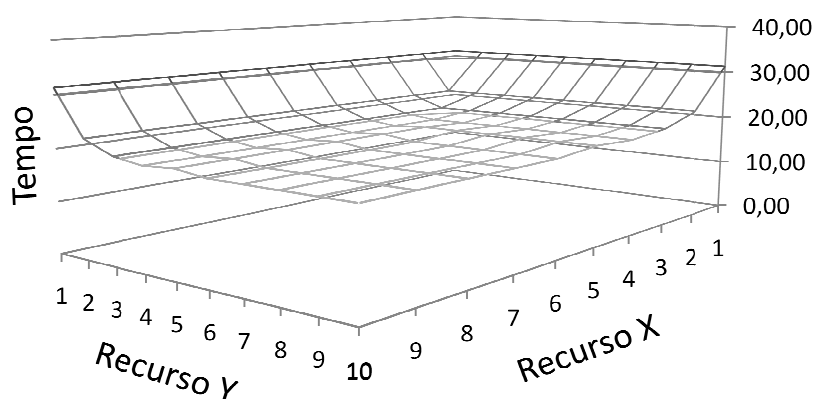


Figura 4.1: Superfície de Resposta Considerando Duração D1 do Projeto

A tabela 4.1 e a figura 4.1 apresentadas, descrevem o comportamento da duração otimista do projeto em função da quantidade de recursos a ele associadas. Assim, caso o projeto seja executado com nove recursos do tipo X e dois recursos do tipo Y

a duração do projeto será, em uma expectativa otimista, de 22,60 unidades de tempo.

O comportamento da superfície de resposta é explicado através dos valores que foram inseridos no simulador. Como todas as atividades necessitam de um recurso de cada tipo para serem executadas, conforme mostrado na tabela 3.2, a matriz de resposta guarda certa simetria, na qual o tempo de duração do projeto cai à medida de são incluídos simultaneamente quantidades unitárias nos dois tipos recursos. No entanto, se a quantidade de um tipo de recurso é acrescida, mas a do outro não, a duração do projeto se mantém inalterada.

Apesar de apresentar de maneira geral o comportamento descrito acima, a simulação *fuzzy* guarda certa curiosidade. Analisando o tempo de duração de um projeto desse tipo para quantidade de recursos $X=Y=4$ e a adição de uma unidade a cada tipo de recurso, ou seja, $X=Y=5$, a duração do projeto parece aumentar de 19,40 para 19,80 unidades de tempo. Este fato também pode ser verificado através de uma pequena ondulação na superfície entre os valores 4 e 6 da figura 4.1.

Tabela 4.2: Duração Dm1 do Projeto em Função dos Recursos Empregados

Dm1		Recursos Y										
		1	2	3	4	5	6	7	8	9	10	
Recursos X	1	41,90	41,90	41,90	41,90	41,90	41,90	41,90	41,90	41,90	41,90	41,90
	2	41,90	29,40	29,40	29,40	29,40	29,40	29,40	29,40	29,40	29,40	29,40
	3	41,90	29,40	26,00	26,00	26,00	26,00	26,00	26,00	26,00	26,00	26,00
	4	41,90	29,40	26,00	23,90	23,90	23,90	23,90	23,90	23,90	23,90	23,90
	5	41,90	29,40	26,00	23,90	23,90	23,90	23,90	23,90	23,90	23,90	23,90
	6	41,90	29,40	26,00	23,90	23,90	23,30	23,30	23,30	23,30	23,30	23,30
	7	41,90	29,40	26,00	23,90	23,90	23,30	23,30	23,30	23,30	23,30	23,30
	8	41,90	29,40	26,00	23,90	23,90	23,30	23,30	23,30	23,30	23,30	23,30
	9	41,90	29,40	26,00	23,90	23,90	23,30	23,30	23,30	23,30	23,30	23,30
	10	41,90	29,40	26,00	23,90	23,90	23,30	23,30	23,30	23,30	23,30	23,30

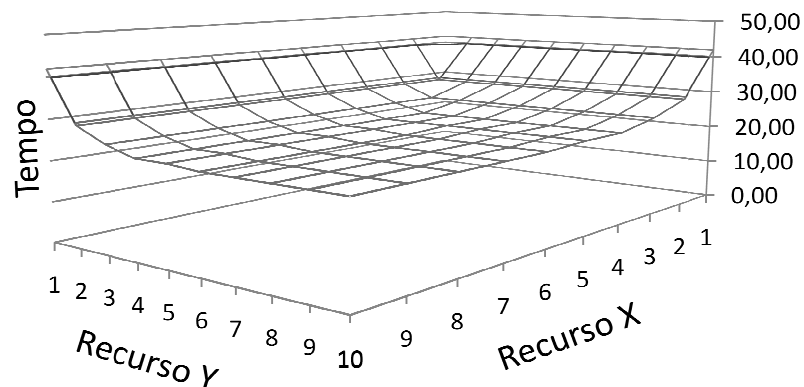


Figura 4.2: Superfície de Resposta Considerando Duração Dm1 do Projeto

Analisando a duração Dm1, na tabela 4.2 e na figura 4.2, não é verificado nenhum tipo de comportamento aparentemente anômalo como aquele verificado na duração otimista D1. Ou seja, a duração do projeto tende a cair à medida que são adicionadas quantidades simultâneas dos dois tipos de recursos até que é atingido um ponto de saturação, no qual a adição de recursos adicionais não resulta em redução da duração do projeto. Considerando-se apenas a duração Dm1, quantidades de recursos acima de seis unidades não resultam em redução da duração do projeto.

Tabela 4.3: Duração Dm2 do Projeto em Função dos Recursos Empregados

Dm2		Recursos Y									
		1	2	3	4	5	6	7	8	9	10
Recursos X	1	52,70	52,70	52,70	52,70	52,70	52,70	52,70	52,70	52,70	52,70
	2	52,70	36,50	36,50	36,50	36,50	36,50	36,50	36,50	36,50	36,50
	3	52,70	36,50	31,70	31,70	31,70	31,70	31,70	31,70	31,70	31,70
	4	52,70	36,50	31,70	28,30	28,30	28,30	28,30	28,30	28,30	28,30
	5	52,70	36,50	31,70	28,30	27,80	27,80	27,80	27,80	27,80	27,80
	6	52,70	36,50	31,70	28,30	27,80	27,90	27,90	27,90	27,90	27,90
	7	52,70	36,50	31,70	28,30	27,80	27,90	27,90	27,90	27,90	27,90
	8	52,70	36,50	31,70	28,30	27,80	27,90	27,90	27,90	27,90	27,90
	9	52,70	36,50	31,70	28,30	27,80	27,90	27,90	27,90	27,90	27,90
	10	52,70	36,50	31,70	28,30	27,80	27,90	27,90	27,90	27,90	27,90

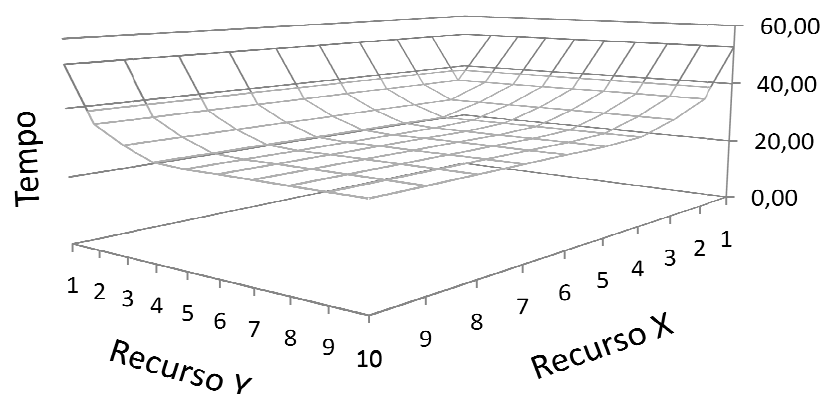


Figura 4.3: Superfície de Resposta Considerando Duração Dm2 do Projeto

Analisando as durações Dm2, na tabela 4.3 e na figura 4.3 pode-se observar, como em Dm1, um comportamento aparentemente anômalo, no qual a duração do projeto aumenta com a adição de recursos. Além disso, neste caso o problema parece ser mais grave dado que a partir de seis recursos de cada tipo (X e Y) a saturação se dá a um nível 0,1 unidades de tempo mais alto que quando se tem $X=Y=5$.

Tabela 4.4: Duração D2 do Projeto em Função dos Recursos Empregados

D2		Recursos Y										
		1	2	3	4	5	6	7	8	9	10	
Recursos X	1	64,20	64,20	64,20	64,20	64,20	64,20	64,20	64,20	64,20	64,20	64,20
	2	64,20	44,10	44,10	44,10	44,10	44,10	44,10	44,10	44,10	44,10	44,10
	3	64,20	44,10	37,80	37,80	37,80	37,80	37,80	37,80	37,80	37,80	37,80
	4	64,20	44,10	37,80	33,40	33,40	33,40	33,40	33,40	33,40	33,40	33,40
	5	64,20	44,10	37,80	33,40	32,30	32,30	32,30	32,30	32,30	32,30	32,30
	6	64,20	44,10	37,80	33,40	32,30	33,30	33,30	33,30	33,30	33,30	33,30
	7	64,20	44,10	37,80	33,40	32,30	33,30	33,30	33,30	33,30	33,30	33,30
	8	64,20	44,10	37,80	33,40	32,30	33,30	33,30	33,30	33,30	33,30	33,30
	9	64,20	44,10	37,80	33,40	32,30	33,30	33,30	33,30	33,30	33,30	33,30
	10	64,20	44,10	37,80	33,40	32,30	33,30	33,30	33,30	33,30	33,30	33,30

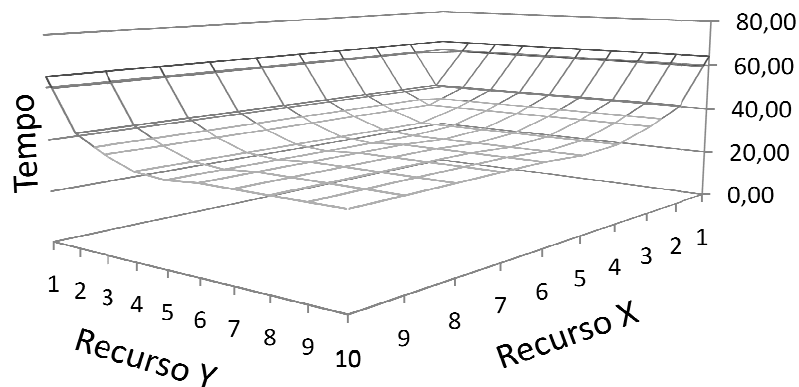


Figura 4.4: Superfície de Resposta Considerando Duração D2 do Projeto

Analisando a tabela 4.4 e a figura 4.4 que representam a duração de projeto pessimista outro comportamento aparentemente anômalo também ocorre. E dessa vez a magnitude é ainda maior. A diferença entre as durações de projeto com $X=Y=5$ e $X=Y=6$ é 10 vezes maior que o encontrado para durações do tipo Dm2.

A explicação para este comportamento aparente anômalo está na própria definição do número *fuzzy*. Quando a duração das atividades do projeto é modelada pelo número *fuzzy* não se está preocupado apenas com a duração otimista, pessimista, ou mais provável. A grande vantagem é a representação dos três (ou quatro, no caso do número trapezoidal) em um único número *fuzzy*. Por isso, para analisar os resultados acima, deve-se ter em mente que não se podem observar as medidas separadamente.

Analisando, por exemplo, a primeira anomalia aparente nos dados gerados para D1 para quantidades de recurso X e Y iguais a quatro e em seguida iguais a cinco, aparentemente tem-se um aumento da duração do projeto com o crescimento da quantidade de recursos. No entanto, ao se analisar o número *fuzzy* completamente, ou seja:

- para recursos $X = 4$ e $Y = 4$, tem-se tempo de duração igual a (19,4; 23,9; 28,3; 33,4);
- para recursos $X = 5$ e $Y = 5$, tem-se tempo de duração igual a (19,8; 23,9; 27,8; 32,3).

Para comparar números *fuzzy* trapezoidais será usado o método proposto por Tran e Duzkstein (2002). Esse método é baseado na comparação dos seguintes alvos pré-determinados:

- crisp máximo (Max); e
- crisp mínimo (Min).

A idéia é que um número *fuzzy* é maior se sua distância para o crisp máximo D_{max} é a menor, mas sua distância para o crisp mínimo D_{min} é a maior. Os valores Max e Min são escolhidos segundo a operação abaixo:

$$\begin{aligned} Min &\leq \inf \left(\cup_{i=1}^l s(A(i)) \right) \\ Max &\leq \sup \left(\cup_{i=1}^l s(A(i)) \right) \end{aligned} \quad (3)$$

Onde $s(A(i))$ é o suporte do número *fuzzy* $A(i)$ a ser ordenado. Baseado na metodologia de ordenação, a fórmula para calcular D_{max} e D_{min} para um número *fuzzy* $A(a_1, a_2, a_3, a_4)$ é apresentada a seguir:

Quando a função peso $f(\alpha) = \alpha$ (tomadores de decisão aversos ao risco):

$$\begin{aligned} D^2(A, M) &= \left(\frac{a_2+a_3}{2} - M \right)^2 + \frac{1}{3} \left(\frac{a_2+a_3}{2} - M \right) \times [(a_4 - a_3) - (a_2 - a_1)] + \\ &\frac{2}{3} \left(\frac{a_3+a_2}{2} \right)^2 + \frac{1}{9} \left(\frac{a_3-a_2}{2} \right) [(a_4 - a_3) + (a_2 - a_1)] + \frac{1}{18} [(a_4 - a_3)^2 + (a_2 - a_1)^2] - \\ &\left(\frac{1}{18} [(a_4 - a_3)(a_2 - a_1)] \right) \end{aligned} \quad (4)$$

Na qual:

- a_1, a_2, a_3 e a_4 representam os quatro parâmetros descritos no número *fuzzy* trapezoidal A ;
- M é tanto Max quanto Min. Logo, $D_{min} = \sqrt{D^2(A, Min)}$ e $D_{max} = \sqrt{D^2(A, Max)}$; e
- $f(\alpha)$ é uma função contínua crescente adotada na fórmula original de Tran e Duzkstein (2002), servindo como uma função peso para o cálculo soma dos pesos das distâncias.

Comparando as durações de projeto para $X=Y=4$ e $X=Y=5$ nota-se que $(19,4; 23,9; 28,3; 33,4) > (19,8; 23,9; 27,8; 32,3)$. E, além disso, analisando os demais dados, observa-se que a quantidade de recursos que resulta na menor duração de projeto acontece quando $X^*=Y^*=5$. Pode-se dizer que utilizando $X=Y=5$ a duração mínima

do projeto será de 19,8 unidades de tempo, a duração máxima do projeto será de 32,3 unidades de tempo e é esperado que a duração fique entre 23,9 e 27,8 unidades de tempo.

5 Conclusão

A utilização da metodologia proposta neste trabalho, baseada na análise da superfície de resposta *fuzzy*, mostrou-se valiosa na resolução de projetos cujos tempos de atividades não possuem valores históricos, inclusive com custo computacional menor do que a simulação de Monte Carlo. Tendo o devido cuidado na análise da superfície de resposta *fuzzy*, essa metodologia é recomendada para a solução deste tipo de problema. As principais vantagens dessa metodologia são:

- facilidade para descrever a duração das atividades através da linguagem verbal;
- menor esforço computacional em comparação com a simulação de Monte Carlo;
- pode ser adaptado para simulação para otimização.

Talvez o principal benefício deste tipo de abordagem seja a aderência com a realidade. A duração das atividades é em geral facilmente descrita através da linguagem verbal. Além disso, a modelagem da duração das atividades através de números *fuzzy* compensa a ausência de dados históricos inerente a projetos inéditos.

Para comprovar a vantagem, dentro deste contexto de imprecisão, da simulação *fuzzy* em relação à simulação de Monte Carlo, no que diz respeito à velocidade de processamento, devem-se compreender os fundamentos e implicações desse tipo de simulação. A utilização de números *fuzzy* para descrever a duração das atividades do projeto permite a soma direta da data de término da atividade com a duração da atividade subsequente. Dessa forma, elimina-se a necessidade de amostragens e a realização de diversas replicações. Com essas simplificações a simulação necessita de apenas um ciclo completo para gerar o(s) resultado(s) de um determinado cenário.

A análise das respostas obtidas pela combinação de 2 (dois) fatores (recursos X e Y) pode ser facilmente realizada graficamente. A resposta que minimiza a quantidade de recursos pode ser obtida através da análise da topologia da superfície de resposta como foi feito no capítulo destinado à análise dos resultados.

Entretanto, é claro que, em casos reais muitos outros recursos poderiam ser considerados na análise, fato que impossibilitaria uma análise visual. No entanto, a adaptação de técnicas da superfície de resposta à superfície de resposta *fuzzy* possibilitariam a construção de algoritmos de simulação para otimização *fuzzy*, que, em tese, seriam capazes de retornar o valor já otimizado. Outro fator que conta a favor da simulação *fuzzy* para aplicação em simulação para otimização é a rapidez de geração de resposta de cada cenário. Essa característica possibilita inclusive aplicação de outros algoritmos além dos que utilizam às técnicas de superfície de resposta mencionadas no capítulo destinado a revisão bibliográfica.

6 Sugestões para Trabalhos Futuros

Dada a potencialidade da aplicação da simulação *fuzzy* em algoritmos de simulação para otimização e o tempo curto para desenvolver todo esse tópico em uma única dissertação, recomenda-se a condução de pesquisas que contribuam para o desenvolvimento desse tema. Os trabalhos futuros podem desenvolver, por exemplo, os seguintes tópicos:

- criação metamodelos a partir de números *fuzzy*;
- otimização de superfícies de respostas *fuzzy*;
- integração entre algoritmos de simulação e otimização *fuzzy*;
- outras aplicações dos números *fuzzy* em gerenciamento de projetos; e
- implementação desses modelos computacionais dentro do paradigma da orientação a objeto; e
- desenvolvimento de um software que contenha um ambiente de simulação genérico (simulador) parametrizável que seja capaz de suportar a implementação rápida e fácil de qualquer projeto, sem a necessidade de conhecimentos aprofundados de programação em C++ ou outra linguagem qualquer.

É fundamental para que os benefícios do desenvolvimento desse tema sejam sentidos pela sociedade, o desenvolvimento e a integração desses algoritmos de simulação para otimização a softwares comerciais de gerenciamento de projetos. Uma vez que esta união for concretizada, a aplicação desses algoritmos estará à disposição do público do gerenciamento de projetos.

Referências Bibliográficas

ALKHAMIS, T. M., AHMED, M. A.. Simulation-based Optimization using simulated annealing with rank and selection. *Computers & Operations Research*, v. 29, p. 387-402, 2002.

ANDRADÓTTIR, S.. A global search method for discrete stochastic optimization. *SIAM Journal on Optimization*, v. 6, p. 513-530. 1996.

ANDRADÓTTIR, S.. Accelerating the Convergence of Random Search Methods for Discrete Stochastic Optimization. *ACM Transactions on Modeling and Computer Simulation*, v. 9 (4), p. 349-380. 2002.

BANKS, J., J.S Carson, B.L. Nelson, D.M.Nicol. (2000). Discrete Event Systems Simulation. 3ª Edição ed. Prentice Hall, Englewoods Cliffs.

BARTON, R.. Simulation metamodels. *Proceedings of the Winter Simulation Conference*, p. 167-174. 1998.

CHEN, H., CHEN, C. YUCESAN, E.. Computing Efforts Allocation for Ordinal Optimization and Discrete Event Simulation. *IEEE Transactions on Automatic Control*, v. 45 (5), p. 960-965. 2000

CONSIGLIO, A., ZENIOS, S. A.. Designing Portfolios of Financial Products via Integrated Simulation and Optimization Models. *Operations Research*, v. 47 (2), p. 195-209. 1999.

EI SAKKOUT, H., WALLACE, M.. Probe backtrack search for minimal perturbation in dynamic scheduling. *Constraints*, v. 5 (4), p. 359–388. 2000.

ELMAGHRABY, S. E., TANER, M. R., FATHI, Y.. On the sensitivity of project variability to activity mean duration. *Int. J. Production Economics*, v. 62, p. 219-232. 1999.

FU, M. C.. Optimization for Simulation: Theory vs. Practice. *INFORMS Journal on Computing*, v. 14 (3), p. 192-215. 2002.

FU, M. C., S. D. HILL.. Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions* , v. 29, p. 233-243. 1997.

- GASS, S. I., HARRIS C.M. (2000). Encyclopedia of Operations Research and Management Science, 2ª edição, Editora Kluwer Academic, Boston.
- GLOVER, F.. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, v. 8, p. 156-166. 1977.
- GLYNN, P. W.. Additional Perspectives on Simulation for Optimization. *INFORMS Journal on Computing*, v. 14 (3), p. 220-222. 2002.
- HO, Y.C., C. G CASSANDRAS, C. H. CHEN, L. Y. DAI.. Ordinal optimization and simulation. *Journal of Operations Research Society*, v. 51, p. 490-500. 2000.
- HUMPHREY, D. G., WILSON, J. R.. A Revised Simplex Search Procedure for Stochastic Simulation Response Surface Optimization. *INFORMS Journal on Computing*, v. 12 (4), p. 272-284. 2000.
- KELLY, J. P.. Simulation Optimization is Evolving. *INFORMS Journal on Computing*, v. 14 (3), p. 223-225. 2002.
- KLEIJNEN, J. P. C., SANCHEZ, S. M., LUCAS T. W., CIOPPA T. M.. A User's Guide to the Brave New World of Designing Simulation Experiments. *INFORMS Journal on Computing*, v. 17 (3), p. 263-289. 2005.
- LAI, S. M.. Surrogate search: a simulation optimization methodology for large scale systems. 2006. School of Engineering, University of Pittsburgh, Pittsburgh, 2006.
- LAW, A.M., KELTON, W.D. (2000). Simulation Modeling and Analysis, 3ª edição. Editora McGraw-Hill, New York.
- LEUS, R.. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, v. 165, p. 289-306. 2005.
- PINTO, Luiz Ricardo. Abordagens usadas em simulação. Belo Horizonte: Escola de Engenharia da UFMG, 2003. 6 f. Notas de aula.
- PRADO, D. S. (2004). Planejamento e Controle de Projeto. Editora INDG Tecs, Belo Horizonte.
- PROJECT MANAGEMENT INSTITUTE. A Guide to the Project Management Body of Knowledge (PMBok). Maryland: Project Management Institute Inc., 2004.
- ROBINSON, S. M.. Analysis of sample-path optimization. *Mathematics of Operations Research*, v. 21, p. 513-528. 1996.

SALIBY, E.. Descriptive Sampling: An Improvement Over Latin Hypercube Sampling. In: *Proceedings of the Winter Simulation Conference* [edited by S. Andradóttir, K.J. Healy, D.H. Withers and B.L. Nelson]. 1997.

TAVARES, L. V.. A review of the contribution of Operational Research to Project Management. *European Journal of Operational Research*, v. 136, p. 1-18. 2002.

TOCHER, K. D. (1963). The art of simulation. English Universities Press, Londres.

TRAN, L., DUZKSTEIN, L.. Comparison of *fuzzy* numbers using a *fuzzy* distance measure. *Fuzzy Sets and Systems*, v. 130, p. 331-341. 2002.

ZHANG, H., TAM, C. M., LI, H.. Modeling uncertain activity duration by *fuzzy* number and discrete-event simulation. *European Journal of Operational Research*, v. 164, p. 715-729. 2005.