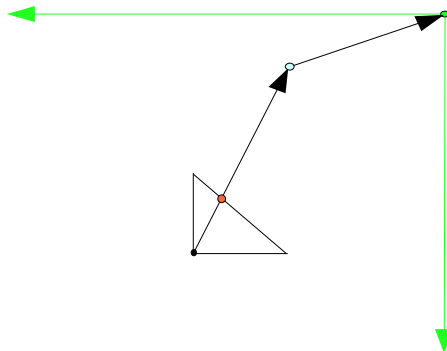




INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa



PRIORIDADES SUAVIZADAS

Francisco José Pais de Almeida de Varennes e Mendonça

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e Computadores

Júri

Presidente: Prof. Carlos Jorge Ferreira Silvestre

Orientador: Prof. Carlos Filipe Gomes Bispo

Vogal: Prof. Susana Isabel Carvalho Relvas

Outubro de 2008

Aos meus Pais que me inspiraram com a sua
vontade de saber mais

Ao minhas Irmãs Joana e Mariana, ao meu
Irmão Vasco, aos meus Primos e à minha "prima"
Maria que me fazem caminhar destemidamente para
o futuro

Agradecimentos

Agradeço a todos os professores que durante o meu percurso académico no Instituto Superior Técnico me ajudaram a aumentar o meu nível de conhecimento e que por isso contribuiriam para a realização desta tese. Agradeço especialmente ao professor Carlos Bispo pela sua orientação na realização desta tese e pelas muitas conversas interessantes sobre diversos assuntos.

Agradeço a minha família e amigos pelo seu constante encorajamento durante a realização desta tese e por todo o apoio e carinho que sempre demonstraram.

Agradeço também a todos os meus colegas, professores amigos e rivais que contribuiriam para a minha formação, tanto académica como pessoal, pelos desafios que me permitiram crescer e alcançar novos horizontes.

Resumo

Em sistemas com limites de capacidade e múltiplos produtos, sujeitos a procura aleatória, é necessário decidir que quantidade de cada produto produzir quando o que seria ideal produzir excede a capacidade. Para tomar essa decisão recorre-se a uma política de produção. Nesta tese é estudada uma nova política de produção chamada "prioridades suavizadas" que se pretende como alternativa mais justa às prioridades estritas. É pensado um modelo geral com capacidade de descrever tanto sistemas série como re-entrantes. São apresentadas as equações de IPA para o modelo com a política de prioridades suavizadas. São feitas comparações teóricas às políticas comuns prioridades estritas e "linear scaling". É pensado software que implemente o modelo, um simulador geral e um otimizador geral permitindo o estudo de qualquer sistema que se inclua no modelo. É usada a Optimização baseada em simulação IPA para comparar praticamente as prioridades suavizadas e estritas concluindo-se que as prioridades suavizadas apresentam melhores resultados para a generalidade dos casos estudados. É estudada a estrutura da função de custo das prioridades suavizadas, com base em gráficos obtidos por simulação, verificando-se que no caso geral a função pode apresentar mínimos locais.

Palavras Chave: *Controlo de Inventário de Revisão Periódica, Múltiplas Máquinas e Produtos, Sistemas Capacitados, Políticas de Base-stock Escalão, Análise de Perturbações Infinitesimais*

Abstract

In multi-product, limited capacity production systems with uncertain demands a decision on the amount to produce for each product is needed when what would ideally be produced exceeds the capacity. One uses a production policy to provide that decision. In this thesis a new production policy called smooth priorities and that is intended as a fairer alternative to strict priorities is studied. A general model for production systems with various products that can describe both series and re-entrant systems is devised. IPA equations for the model with a smooth priorities policy are presented. Theoretical comparisons are made to the common policies of strict priorities and linear scaling. Software that implements the model, a general simulator and a general optimizer are devised allowing the study of any system that fits the model. Optimization based on IPA simulation is used to practically compare smooth and strict priorities with the conclusion that in general smooth priorities present better result for the studied cases. Function plots obtained from simulation are used to study the structure of the cost function of smooth priorities resulting in the knowledge that in general the function can exhibit local minimums.

Keywords: *Periodic Review Inventory Control, Multiple products and machines, Capacitated Systems, Base-stock Echelon Policies, Infinitesimal Perturbation Analysis*

Índice

1. Introdução.....	1
2. Revisão Literária.....	3
2.1 Modelo Não Capacitado.....	3
2.2. Modelos Capacitados.....	6
2.3. Modelos Capacitados com Múltiplos Produtos.....	7
PARTE I: Trabalho Teórico.....	10
3. Bases	11
3.1 Modelo do Sistema.....	11
3.2 Simulação.....	13
3.2.1 Geração de variáveis aleatórias.....	13
3.2.2 Estimação de parâmetros.....	14
3.3 Optimização.....	14
4. Políticas de Produção.....	17
4.1 Políticas de Base-Stock	17
4.2 Prioridades Suavizadas.....	19
4.2.1 Introdução da Política.....	19
4.2.2 Comparação com LSR.....	20
4.2.3 Comparação com Prioridades Estritas.....	21
PARTE II: Trabalho Prático.....	23
5. Software	24
5.1 Modelo do Sistema.....	24
5.2 Simulação.....	24
5.3 Optimização.....	25
6. Comparação com Prioridades Estritas.....	27
6.1. Confirmação de Resultados Prévios.....	27
6.2. Diferentes Custos.....	28
6.3. Sistema Complexo de 3 Máquinas e 2 Produtos Finais.....	29
6.4. Estrutura da Função de Custo.....	30
7. Conclusões e Trabalho Futuro.....	32
8. Bibliografia.....	33
ANEXOS.....	34
A. Diagramas UML.....	35
Diagrama de Classes de FactoryModel.....	35
Diagrama de Classes de Factory::Control	36
Diagrama de Classes de PDFRandomGeneration.....	36
Diagrama de Classes de DESSimulation.....	37
Diagrama de Classes de Optimization.....	38
B. Gráficos da Função de Custo.....	39
B.1. Base-Stocks Iguais aos Óptimos para Prioridade do Produto 1.....	39
B.2. Base-Stocks Iguais aos Óptimos para Prioridade do Produto 2.....	40
B.3. Base-Stocks Iguais aos Óptimos para Prioridades Suavizadas.....	41
B.4. Mesmo Produto de Fase: Delta e Alfa.....	42
B.5. Mesma Máquina: Delta e Alfa de produtos de fase diferentes.....	43
B.6. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0$).....	44
B.7. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0.2$).....	45
B.8. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0.4$).....	46

B.9. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0.6$).....	47
B.10. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0.8$).....	48
B.11. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 1$).....	49
B.12. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 1$) ZOOM 1.....	50
B.13. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 1$) ZOOM 2.....	51
B.14. Mesmo Produto Final: Alfas em várias máquinas ($\text{Alfa}(2,3) = 0$).....	52
B.15. Mesmo Produto Final: Alfas em várias máquinas ($\text{Alfa}(2,3) = 0.2$).....	53
B.16. Mesmo Produto Final: Alfas em várias máquinas ($\text{Alfa}(2,3) = 0.4$).....	54
B.17. Mesmo Produto Final: Alfas em várias máquinas ($\text{Alfa}(2,3) = 0.6$).....	55
B.18. Mesmo Produto Final: Alfas em várias máquinas ($\text{Alfa}(2,3) = 0.8$).....	56
B.19. Mesmo Produto Final: Alfas em várias máquinas ($\text{Alfa}(2,3) = 1$).....	57
C. Resultados de Optimização com Vários Custos.....	58

Lista de Abreviaturas

i.e. - id est (isto é)

IPA - Infinitesimal Perturbation Analysis

maq. - máquina

LSR - Linear Scaling Rule

PE - Prioridades Estritas

prod. - produto

PS - Prioridades Suavizadas

UML - Unified Modeling Language

XML - eXtensible Markup Language

1. Introdução

No contexto de sistemas de produção com limites de capacidade para múltiplos produtos, sujeitos a procura aleatória a política de produção óptima tem desafiado formulações em forma fechada e mostrado-se excessivamente complexa para se calcular numericamente. Por essa razão, é frequente recorrer-se a políticas sub-óptimas definidas por um conjunto pequeno de parâmetros que permite a optimização baseada em simulação.

De entre as políticas sub-óptimas normalmente usadas, a decisão com base numa lista de prioridades é das mais comuns. No entanto, o mecanismo de prioridade estrita dificulta a satisfação da procura para os produtos menos prioritários. Em determinados casos essa dificuldade pode não ser satisfatória. Para além disto, a simples determinação de qual a lista de prioridades adequada é um problema combinatório de difícil resolução. Nesse sentido, existe espaço para se explorarem mecanismos de prioridade que permitam um comportamento mais equitativo para todos os produtos, sem com isso prejudicar excessivamente aqueles que são mais prioritários. Por trás destas ideias está o conceito de prioridades suavizadas, onde todos os produtos têm sempre acesso aos recursos produtivos através de alguma ponderação que reflecte a sua prioridade relativa. A formulação do problema da determinação destas ponderações permite transformar o problema combinatório de definição de prioridades num problema de programação não linear para variáveis contínuas, o que significa uma enorme vantagem do ponto de vista da complexidade computacional necessária para essa determinação (i.e. uma redução significativa).

Os objectivos desta tese são aprofundar o estudo sobre a política de prioridades suavizadas e criar software genérico para Análise de Perturbações Infinitesimais (ver [1]) realizada sobre simulação, contendo algoritmos de optimização e permitindo o estudo de vários sistemas produtivos.

Esta tese encontra-se dividida diversos capítulos e o trabalho realizado é apresentado numa primeira parte teórica e numa segunda parte prática.

No Capítulo 2 é feita a revisão literária relevante ao tema desta tese, sendo composta por três secções. A primeira secção apresenta o modelo de Clark-Scarf (ver [2]), considerado o modelo base dos sistemas de inventário/produção e alguns resultados sobre um sistema de duas máquinas (resumido de [3]). A segunda secção faz a revisão dos resultados mais relevantes para modelos de inventário/produção em que as etapas podem apresentar limites de capacidade ([4],[5],[6],[7] e [8]). A terceira secção apresenta resultados para modelos de inventário/produção em que cada máquina pode processar mais do que um produto e que têm limites de capacidade ([9],[10] e [11]).

Na primeira parte é apresentado o trabalho teórico desenvolvido, sendo composta pelo terceiro e quarto capítulos. O Capítulo 3 apresenta o modelo usado tal como a base teórica (de forma resumida) e as principais decisões necessárias à implementação da parte experimental. O capítulo 4 apresenta a aplicação de algumas políticas de base-stock simples ao modelo desta tese, introduz as prioridades suavizadas comparando-a às anteriores.

Na segunda parte é apresentado o trabalho prático desenvolvido, sendo composta pelo quinto e sexto capítulos. O Capítulo 5 descreve de maneira sucinta o estrutura e funcionalidade do software desenvolvido e utilizado nesta tese. O Capítulo 6 compara as prioridades suavizadas com as prioridades estritas. Para tal, retoma um caso de aplicação apresentado em [10] confirmando os resultados prévios, tenta verificar a relevância das prioridades suavizadas introduzindo dois novos casos de aplicação para o efeito e tenta também tirar conclusões sobre a estrutura da função de custo.

A tese termina com as conclusões e trabalho futuro no Capítulo 7, bibliografia no Capítulo 8 e os anexos A, B e C com os diagramas UML do software desenvolvido, os gráficos de amostra da função de custo e uma tabela de resultados.

Considera-se que as principais contribuições desta tese são, em primeiro lugar, o modelo de

sistema e o software gerais e flexíveis, em segundo lugar, apresentar as prioridades suavizadas como uma alternativa às estritas mas também como metodologia que permite determinar a ordem de prioridade e, em terceiro lugar, investigar a estrutura da função de custo para esta nova política.

2. Revisão Literária

Este capítulo encontra-se dividido em três secções e apresenta uma revisão da literatura de sistemas de inventário/produção.

2.1 Modelo Não Capacitado

Nesta secção apresenta o modelo básico dos sistemas de inventário/produção e alguns resultados sobre um sistema de duas máquinas. Os primeiros resultados de política óptima sobre este modelo foram apresentados em [2] e o modelo é normalmente chamado de modelo de Clark-Scarf, cujas características são enumeradas de seguida (o mesmo estilo será usado para as descrições de modelos apresentados subsequentemente). O conteúdo desta secção consiste num resumo parcial do capítulo 2 de [3].

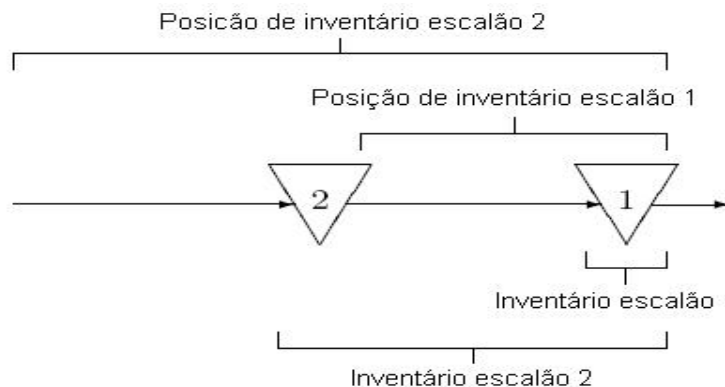


Figura 2.1.1 - Representação dos pontos de stock de duas etapas e ilustração do conceito de inventário escalão e posição de inventário escalão

Modelo:

- Tempo passa em períodos de tempo iguais;
- Etapas numeradas de 1 a N;
- Etapa N encomenda a um fornecedor externo;
- Matéria Prima sempre disponível (i.e. não existe atraso na encomenda ao fornecedor);
- Etapa k encomenda à etapa k+1;
- Procura externa ocorre na etapa 1;
- Cada etapa:
 - representa nó de transporte ou produção;
 - os produtos são guardados em pontos de stock no final da etapa;
 - pode encomendar no início de cada período de tempo;
 - não pode encomendar mais do que o stock existente na etapa fornecedora;
 - os produtos encomendados chegam ao final da etapa que encomendou ao fim de um número constante de períodos ;
- A procura é um processo estocástico estacionário;
- Os custos são a soma de custos de holding (stock existente) e de backlog (procura não satisfeita) lineares;
- A sequência em cada período é:
 - 1 – cada etapa encomenda;
 - 2 – encomendas chegam;
 - 3 – procura ocorre ;
 - 4 – custos do período são calculados;

- Horizonte Temporal infinito;
- Inventário e encomendas são consideradas valores reais.

Sistema de duas máquinas/etapas (caso particular que ilustra o caso geral)

Notação:

t - Tempo, $t \in \mathbb{N}_0$

l_n - Tempos de atraso, $l_n \in \mathbb{N}$ (para etapa $n = 1$ ou 2)

D_{t_1, t_2} - Procura acumulada entre t_1 e t_2

x_n - Inventário escalão (escalão da etapa $n = 1, 2$)

c_n - Função de custo (convexa e é subóptimo deixar backlog crescer até infinito)

Π - conjunto de todas as políticas de encomenda

$G(\pi)$ - média do custo médio da política $\pi \in \Pi$

De modo a que os resultados que se seguem sejam extensíveis por indução considera-se a estrutura de custo como soma de função convexas (não restringindo a custos de holding e backlog lineares), i.e. $\sum c_n(x_n)$.

Achar política óptima (média do custo)

Objectivo: achar $\arg \min_{\pi} G(\pi)$

Passos a dar:

- 1 – definição de ciclos e custos de ciclo;
- 2 – solução dum problema de ciclo único relaxado;
- 3 – solução do problema de horizonte infinito.

Passo 1: Definição de ciclos e custo de ciclo

Notação:

$IL_{t,n}$ - nível de inventário escalão (início período) ($t = n^\circ$ de período, $n = n^\circ$ etapa)

$IP_{t,n}$ - posição de inventário escalão (início período) ($t = n^\circ$ de período, $n = n^\circ$ etapa)

$C_{t,n}$ - custo devido ao inventário escalão (fim período) ($t = n^\circ$ de período, $n = n^\circ$ etapa)

Os ciclos são definidos por decisões sobre $IP_{t_0,2}$ ($=z_2$) e sobre $IP_{t_0+l_2,1}$ ($=z_1$). É importante reforçar a distinção entre períodos e ciclos. O ciclo é definido como um conjunto de decisões que se influenciam directamente em períodos de tempo diferentes, ou seja, um ciclo é definido pela decisão na máquina 2 em t_0 e a decisão na máquina 1 em t_0+l_2 (quando o resultado da decisão em dois chega a 1).

Custo esperado da decisão em t_0 na etapa 2 em t_0+l_2 :

$$E[C_{t_0+l_2,2} | IP_{t_0,2} = z_2] = E[c_2(z_2 - D_{t_0, t_0+l_2})]$$

Custo esperado da decisão em t_0+l_2 na etapa 1 em $t_0+l_2+l_1$:

$$E[C_{t_0+l_2+l_1,1} | IP_{t_0+l_2,1} = z_1] = E[c_1(z_1 - D_{t_0+l_2, t_0+l_2+l_1})]$$

$$z_2 \geq IL_{t_0,2}, \quad z_1 \geq IL_{t_0+l_2,1}, \quad z_1 \leq z_2 - D_{t_0, t_0+l_2-1} = IL_{t_0+l_2,2}$$

Custo total do ciclo que começa em t_0 :

$$C_{t_0} := C_{t_0+l_2,2} + C_{t_0+l_2+l_1,1}$$

De seguida demonstra-se que o valor da média do custo é igual à média dos custos de ciclo que garante que a política óptima encontrada seja óptima pelo critério de custo médio:

$$\begin{aligned}
G(\pi) &= \lim_{T \rightarrow \infty} \frac{1}{T} E \left[\sum_{t=0}^{T-1} (C_{t,2} + C_{t,1}) \right] \\
&= \lim_{T \rightarrow \infty} \frac{1}{T} E \left[\sum_{t=0}^{T-1} C_t + \sum_{t=0}^{l_2-1} C_{t,2} + \sum_{t=0}^{l_2+l_1-1} C_{t,1} - \sum_{t=T}^{T+l_2-1} C_{t,2} - \sum_{t=T}^{T+l_2+l_1-1} C_{t,1} \right] \\
&= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E[C_t] = E[C_t]
\end{aligned}$$

Passo 2: Solução dum problema de ciclo único relaxado

Objectivo: minimizar $E[C_{t_0}]$ através das decisões na etapa 1 e 2 (z_1 e z_2)

Ignorando as restrições que limitam inferiormente z_2 e z_1 ($z_2 \geq IL_{t_0,2}$, $z_1 \geq IL_{t_0+l_2}$) temos o problema relaxado:

$$(RP(t_0)): \text{Min } E[C_{t_0}] = E[C_{t_0+l_2,2}] + E[C_{t_0+l_2+l_1,1}]$$

s.a

$$E[C_{t_0+l_2,2}] = E[c_2(z_2 - D_{t_0, t_0+l_2})]$$

$$E[C_{t_0+l_2+l_1,1}] = E[c_1(z_1 - D_{t_0+l_2, t_0+l_2+l_1})]$$

$$z_1 \leq IL_{t_0+l_2,2}$$

$$IL_{t_0+l_2,2} = z_2 - D_{t_0, t_0+l_2-1}$$

(RP(t_0)) é um problema estocástico de duas etapas.

Para o resolver, primeiro determinamos a decisão óptima para z_1 definindo para isso a função de custo $G_1(y_1)$ em que y_1 é a possível decisão para a qual se quer achar o óptimo:

$$G_1(y_1) := E[c_1(y_1 - D_{t_0+l_2, t_0+l_2+l_1})], \quad y_1 \in \mathbb{R}$$

Pode-se constatar que:

- (i) $G_1(y_1)$ é convexa em y_1 ;
- (ii) escolhendo $S_1 \in \mathbb{R} \cup \{\infty\}$ tal que:

$$S_1 := \underset{y_1}{\text{arg min}} G_1(y_1)$$

então é óptimo escolher:

$$z_1 = S_1, \quad S_1 \leq IL_{t_0+l_2,2}$$

$$z_1 = IL_{t_0+l_2,2}, \quad c.c.$$

i.e., aplicar política de base-stock S_1 a z_1 .

Uma política de base-stock tenta portanto atingir um nível de stock determinado, neste caso S_1 , e se tal não for possível aproxima-se o mais possível dele.

Depois definimos $G_2(y_1, y_2)$ quando se aplica uma política de base-stock y_2 a z_2 e base-stock y_1 a z_1 , $y_1, y_2 \in \mathbb{R}$

$$z_1 = \min\{IL_{t_0+l_2,2}, y_1\} = \min\{z_2 - D_{t_0, t_0+l_2-1}, y_1\}$$

$$z_2 = y_2, \text{ matéria prima sempre disponível}$$

$$G_2(y_1, y_2) := E[c_2(z_2 - D_{t_0, t_0+l_2}) + c_1(\min\{z_2 - D_{t_0, t_0+l_2-1}, y_1\} - D_{t_0+l_2, t_0+l_2+l_1})]$$

Pode-se constatar que:

- (i) $G_2(S_1, y_2)$ é convexa em $y_2 \in \mathbb{R}$
- (ii) escolhendo $S_2 \in \mathbb{R} \cup \{\infty\}$ tal que:

$$S_2 := \underset{y_2}{\text{arg min}} G_2(S_1, y_2),$$

então para o problema (RP(t_0)) é óptimo escolher $z_2 = S_2$

Destes dois resultados tem-se que é óptimo, para o problema (RP(t_0)), aplicar uma política de base-stock com níveis S_1 e S_2 e o custo óptimo é $G_2(S_1, S_2)$. Como se ignoraram as restrições que limitam inferiormente, $G_2(S_1, S_2)$ é um limite inferior para o custo óptimo do problema original.

Passo 3: solução do problema de horizonte infinito

As funções $G_1(y_1)$ e $G_2(y_1, y_2)$ têm interpretações alternativas. $G_1(y_1)$ representa o custo médio dum política de base-stock y_1 a um sistema de uma etapa, ao qual chamamos subsistema 1. $G_2(y_1, y_2)$ representa o custo médio de uma política de base-stock (y_1, y_2) para o sistema de duas etapas completo.

Supondo que uma política de base-stock (S_1, S_2) é usada no problema original então o custo médio obtido é $G_2(S_1, S_2)$ que é igual ao limite inferior. Isto implica que é óptimo para o problema original. É de notar que a política é de base-stock para inventário escalão.

Para provas das afirmações nos passos 2 e 3 ver [3]

Notas finais:

- Esta prova pode ser facilmente estendida para mais de duas etapas por indução.
- Clark e Scarf provaram em [2] optimalidade das políticas de base-stock e que os níveis de base-stock são obtidos por minimização de funções de custo convexas uni-dimensionais.
- O facto de os níveis de base-stock poderem ser obtidos por minimização sequencial de funções uni-dimensionais é conhecido por resultado de decomposição (*decomposition result*)
- Em [2] a prova equivalente à aqui apresentada é feita com base em funções de custo induzidas. Exemplo para o subsistema 1: $\tilde{G}_1(y) = G_1(\min\{y, S_1\}) - G_1(S_1)$ representa o custo de o stock em 2 ser insuficiente para aumentar $IP_{t,1}$ até ao nível óptimo S_1 .

2.2. Modelos Capacitados

Nesta secção faz-se a revisão dos resultados mais relevantes para modelos de inventário/produção em que as etapas podem apresentar limites de capacidade. Começa por se apresentar os resultados de optimalidade com apenas uma máquina sobre diversos critérios de optimalidade. De seguida, apresentam-se resultados de optimalidade para o caso de duas máquinas em série. Para finalizar apresentam-se resultados para várias máquinas em série que assumem uma política de base-stock. Nos modelos aqui apresentados evidenciam-se os aspectos que diferem do modelo de Clack-Scarf, sendo similares as características omissas.

Em [4] com o seguinte modelo:

- uma máquina;
- capacidade de produção finita;
- procura discreta.

Notação:

C - capacidade da máquina/etapa

D - distribuição da procura

É provada a optimalidade de política de base-stock para o critério de custo médio para o horizonte infinito na seguinte condição $C > E[D]$.

Em [5] com o seguinte modelo:

- uma máquina;
- capacidade de produção finita;
- procura contínua.

Notação:

C - capacidade da máquina/etapa

D - distribuição da procura

É provada optimalidade de política de base-stock para o critério de custo descontado para o horizonte infinito na seguinte condição $C > E[D]$.

Em [6] com o seguinte modelo:

- duas máquinas em série;

- capacidade de produção finita (máquina que recebe a procura tem capacidade menor).

É provada a optimalidade de política de base-stock modificada (com limite do inventário local) para os critérios de custo descontado e custo médio para o horizonte infinito. A prova assenta no resultado de separabilidade (para duas máquinas) da função de custo numa banda que é provada ser onde uma política ótima opera.

Em [7] e [8] explora-se uma aproximação mais prática e assume-se uma política de base-stock (provada ótima para casos mais simples) ao seguinte modelo:

- N máquinas em série;

- capacidade finita em todas as máquinas.

Notação:

K_i - capacidade da máquina i

D - distribuição da procura

[7] prova estabilidade da política de base-stock quando $\min\{K_i\} < E[D]$ e [8] desenvolve estimadores de derivadas e mostra que convergem para o valor correcto. Com estes resultados é possível então saber se uma política de base-stock será estável e utilizar optimização por gradiente para achar os níveis de base-stock óptimos.

2.3. Modelos Capacitados com Múltiplos Produtos

Nesta secção apresentam-se alguns resultados para modelos de inventário/produção em que cada máquina pode processar mais do que um produto e que têm limites de capacidade. Começa-se por apresentar os resultados da tese de doutoramento [9] que inspirou a criação das prioridades suaves. De seguida, apresenta-se o trabalho final de curso [10] que apresenta a regra sobre o nome de Linear Scaling Rule em duas fases e os primeiros resultados práticos que provam poder obter custos médios mais baixos do que as prioridades estritas. Para finalizar apresenta-se os resultados de um "working paper" [11] que se debruça sobre o tema da optimalidade numa máquina com múltiplos produtos.

Em [9] com o seguinte modelo:

- M máquinas em série;

- capacidade finita em todas as máquinas;

- P produtos que percorrem as M máquinas K vezes (K níveis).

Notação:

C_i - capacidade da máquina i

d_n^p - procura do produto p no período de tempo n

D - distribuição conjunta das procuras dos produtos ($D_n = [d_n^1 \dots d_n^P]$)

I^{mkp} - inventário do nível k na máquina m do produto p

h^{mkp} - custo de holding associado a I^{mkp}

b^p - custo de backlog associado ao produto p

São analisadas várias regras de produção (usadas quando existe capacidade partilhada):

- Linear Scaling Rule (LSR);
- Prioridades estritas (PE);
- Equalized Shortfall Rule (ESR);

e políticas de alocação de capacidade:

- a capacidade é dividida por cada nível de cada produto (sem partilha);
- a capacidade é dividida por cada nível (partilha parcial);
- a capacidade não é dividida (partilha total).

É provada a aplicabilidade de IPA (infinitesimal perturbation analysis) às várias regras/políticas. É provada a estabilidade para $E[||D||] < \min\{C_i\}$ para as várias políticas e é deduzida a seguinte condição (necessária) de optimalidade $Pr(d_n^p \leq I^{11 p}) = \frac{b^p}{(b^p + h^{11 p})}$

São tiradas várias conclusões práticas sobre regras de produção e políticas de alocação sendo as mais relevantes para esta tese:

- o modo de partilha total permite melhores resultados que a partilha parcial (embora LSR tenha melhores resultados em partilha parcial);
- para casos em que não há uma razão clara para a prioridade de qualquer produto ESR tem melhores resultados;
- para os casos em que há uma clara razão para a prioridade as Prioridades estritas têm melhores resultados;
- no caso de cargas (sobre a capacidade) diferentes para os vários produtos e níveis as Prioridades estritas e a ESR revelam grandes variações no gradiente estimado em períodos de grande procura e a LSR não.

Em [10] com o seguinte modelo:

- M máquinas em série;
- capacidade finita em todas as máquinas;
- P produtos percorrem as M máquinas.

São apresentadas três novas regras de produção baseadas na LSR:

- LSR em duas fases (Prioridades Suavizadas);
- LSR com limite de produção local;
- LSR com limite de inventário local;

e as derivadas necessárias à aplicação de IPA ao modelo

São apresentados resultados práticos da aplicação da LSR em duas fases com uma só máquina e dois produtos em que se verifica ter melhores resultados que a regra de prioridades estritas

São apresentados resultados da variação de vários parâmetros do modelo no custo comparando a LSR em duas fases e as prioridades estritas no caso de três máquinas e dois produtos em que se verifica obter sempre melhores resultados com a LSR em duas fases.

Em [11] com o seguinte modelo:

- uma máquina;
- capacidade limitada de produção;
- N produtos.

São estudadas as prioridades estritas e WBR (Weighted Balancing Rule). A WBR aloca capacidade para o produto com maior "shortfall" (pesada pelo inverso da média da procura) até à "shortfall" coincidir com o segundo maior e assim iterativamente até acabar a capacidade.

Diz provar que existe uma política estacionária de base-stock óptima para o caso de custo

médio e custo descontado de horizonte infinito.

Diz provar que a distribuição dos "shortfalls" finais é independente dos níveis de base-stock e converge quando $t \rightarrow \infty$.

É apresentada a fórmula para o cálculo dos níveis de base-stock ótimos a partir da convolução da distribuição marginal das "shortfalls" de cada produto e a sua distribuição de procura

$$e^{-\frac{b^n}{b^n + h^n}}.$$

Diz provar optimalidade da WBR para o caso de $h^n = h$, $b^n = b$, (D^1, \dots, D^N) simétrica e inventários iniciais iguais aos níveis de base-stock.

Diz provar optimalidade assintótica de WBR para níveis de serviço altos, i.e., $h^n = h$, (D^1, \dots, D^N) simétrica e (b^1, \dots, b^N) vão para infinito por multiplicação por um escalar.

PARTE I: Trabalho Teórico

3. Bases

Este capítulo encontra-se dividido em três secções e apresenta o modelo usado tal como a base teórica (de forma resumida) e as principais decisões necessárias à implementação da parte experimental. Na primeira secção apresenta-se o modelo do sistema utilizado para a realização desta tese, a notação utilizada, as equações relevantes e as derivadas das equações necessárias à implementação da IPA. Na segunda secção apresenta-se o algoritmo de simulação, técnicas para geração de variáveis aleatórias e pormenores sobre a estimação de parâmetros. Na terceira secção apresentam-se vários métodos de optimização e o algoritmo escolhido.

3.1 Modelo do Sistema

O modelo do sistema foi pensado de forma a englobar tanto o modelo série e como o reentrante, apresentados em [10] e [9] respectivamente, permitindo qualquer fluxo pelas diversas máquinas de linhas de produção que não divirjam nem converjam (i.e. não é necessário mais do que um produto para criar o seguinte nem um produto serve para criar mais de um produto).

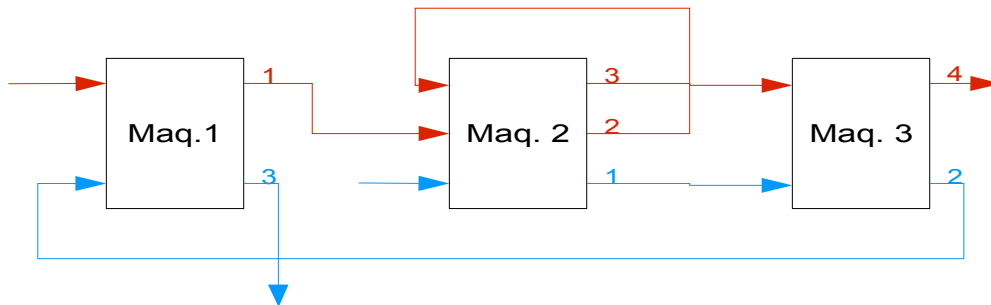


Figura 3.1.1 - Representação de um sistema com 3 máquinas e 2 produtos que exemplifica a flexibilidade do modelo

Modelo:

- \mathcal{M} máquinas;
- capacidade finita;
- \mathcal{P} produtos finais;
- cada produto final p tem \mathcal{F}_p produtos de fase;
- cada produto de fase é produzido numa das máquinas (sem qualquer restrição);
- matéria prima com entrega perfeita (tudo o que é necessário está logo disponível);
- procura contínua sobre o último produto de fase de cada produto final após produção;
- cada produto de fase pode consumir quantidades de capacidade diferentes na produção;
- custos lineares de holding e backlog (logo após procura);
- quantidades de produtos de valores contínuos.

Notação:

- \mathcal{M} - número de máquinas;
- \mathcal{P} - número de produtos finais;
- \mathcal{F}_p - número de produtos de fase do produto final p ($p=1, \dots, \mathcal{P}$);
- M_m - conjunto dos produtos de fase que são feitas na máquina m ;
- K_m - capacidade da máquina m ;
- $I_t^{p,f}$ - inventário do produtos de fase f do produto final p no período de tempo t ;
- $E_t^{p,f}$ - inventário escalão do produtos de fase f do produto final p no período de tempo t ;
- $P_t^{p,f}$ - produção do produtos de fase f do produto final p no período de tempo t ;

- d_t^p - procura do produto p no fim do período de tempo t ;
- $h^{p,f}$ - custo de holding para o produto de fase f do produto final p ;
- b^p - custo de backlog para o produto final p ;
- $\tau^{p,f}$ - carga (i.e unidades de capacidade por unidades de produto) do produto de fase f do produto final p ;

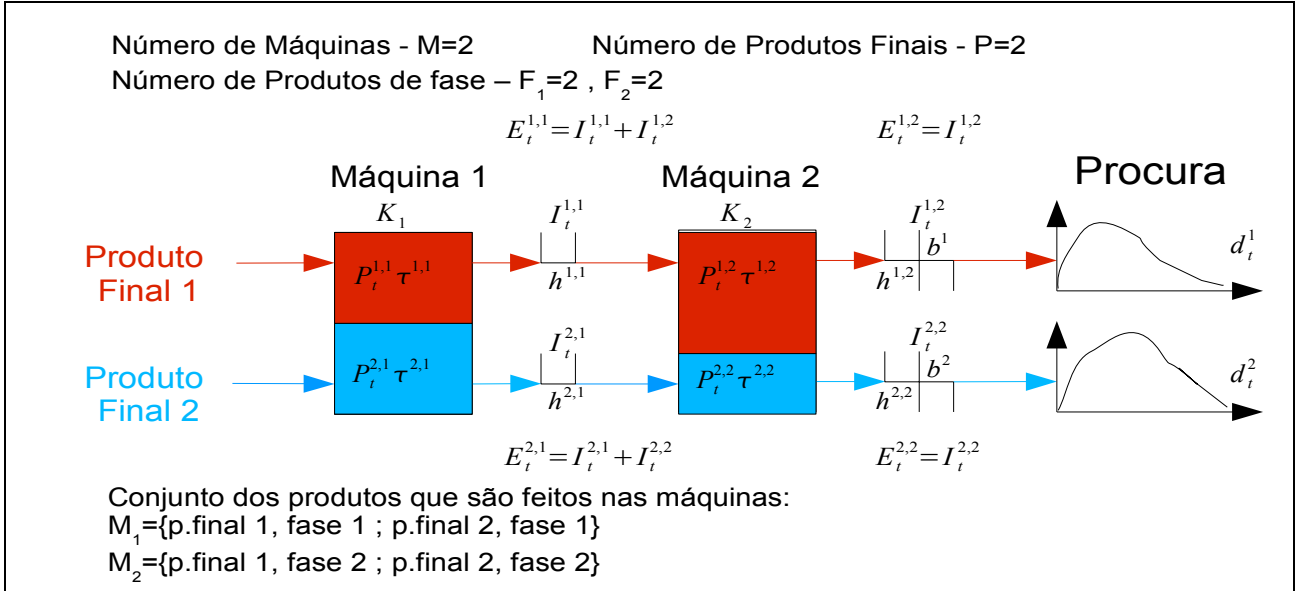


Figura 3.1.2 - Exemplo ilustrativo da notação

As seguintes equações mostram como o sistema evolui de período para período:

$$I_{t+1}^{p,f} = I_t^{p,f} + P_t^{p,f} - P_t^{p,f+1} \quad , \quad P_t^{p,\mathcal{F}_p+1} \equiv d_t^p$$

$$E_t^{p,f} = \sum_{i=f}^{\mathcal{F}_p} (I_t^{p,i})$$

$$E_{t+1}^{p,f} = E_t^{p,f} + P_t^{p,f} - d_t^p$$

As seguintes equações permitem calcular os custos operacionais em cada período:

$$C_t = \sum_{p=1}^{\mathcal{P}} (C_t^p)$$

$$C_t^p = -\min(I_t^{p,\mathcal{F}_p}, 0) \times b^p + \max(I_t^{p,\mathcal{F}_p}, 0) \times h^{p,\mathcal{F}_p} + \sum_{f=1}^{\mathcal{F}_p-1} (I_t^{p,f} \times h^{p,f}) =$$

$$= -\min(I_t^{p,\mathcal{F}_p}, 0) \times (b^p + h^{p,\mathcal{F}_p}) + \sum_{f=1}^{\mathcal{F}_p} (I_t^{p,f} \times h^{p,f})$$

Qualquer política de produção possível cumpre para cada máquina: $K_m \geq \sum_{(i,j) \in M_m} (\tau^{i,j} P^{i,j})$

Derivadas do modelo necessárias à IPA:

$$\frac{\delta I_{t+1}^{p,f}}{\delta x} = \frac{\delta I_t^{p,f}}{\delta x} + \frac{\delta P_t^{p,f}}{\delta x} - \frac{\delta P_t^{p,f+1}}{\delta x} \quad , \quad \frac{\delta P_t^{p,\mathcal{F}_p+1}}{\delta x} \equiv \frac{\delta d_t^p}{\delta x} = 0$$

$$\frac{\delta C_t}{\delta x} = \sum_{p=1}^{\mathcal{P}} \frac{\delta C_t^p}{\delta x}$$

$$\frac{\delta C_t^p}{\delta x} = -\{I_t^{p,\mathcal{F}_p} < 0\} \times \frac{\delta I_t^{p,\mathcal{F}_p}}{\delta x} b^p + \sum_{f=1}^{\mathcal{F}_p} \{I_t^{p,f} > 0\} \times \frac{\delta I_t^{p,f}}{\delta x} h^{p,f}$$

Como C_t^p não é diferenciável em $I_t^{p,\tilde{p}}=0$ tem que se ter $I_t^{p,\tilde{p}}\neq 0$ com probabilidade 1 para que seja válida. Como se assume que a procura é contínua tal é sempre verdade se a distribuição não tiver um ponto de massa num valor igual ao base-stock.

3.2 Simulação

Optou-se por um simulador bastante geral que apenas assume que a criação de eventos é independente do estado do sistema. De seguida apresenta-se a descrição do algoritmo para o simulador.

Notação:

\mathbf{x} - estado;

\mathbf{e} - evento;

$\mathbf{x}^+ = f(\mathbf{x}, \mathbf{e})$ - modelo que descreve a mudança de estado quando ocorre um evento;

\mathbf{t} - o tempo (contínuo) do sistema;

$(\mathbf{e}, \mathbf{t}^+) = g()$ - gerador de eventos que emite o próximo evento \mathbf{e} e tempo em que ocorre \mathbf{t}^+ ;

$\mathbf{a}(\mathbf{t}, \mathbf{x}, \mathbf{x}^+, \mathbf{e})$ - avaliador permite ir calculando estimativas de quantidades de interesse sem ter de guardar todos os dados da simulação para análise posterior;

\mathbf{A} - conjunto de avaliadores;

\mathbf{x}_0 - estado inicial da simulação;

\mathbf{t}_0 - tempo inicial da simulação;

$\mathbf{t}_{\text{terminal}}$ - tempo até ao qual se quer simular;

Descrição do Algoritmo de Simulação:

1) $\mathbf{x} = \mathbf{x}_0$, $\mathbf{t} = \mathbf{t}_0$;

2) $(\mathbf{e}, \mathbf{t}^+) = g()$;

3) se $\mathbf{t}^+ > \mathbf{t}_{\text{terminal}}$ termina simulação;

4) $\mathbf{t} = \mathbf{t}^+$;

5) $\mathbf{x}^+ = f(\mathbf{x}, \mathbf{e})$;

6) para todos \mathbf{a} em \mathbf{A} : $\mathbf{a}(\mathbf{t}, \mathbf{x}, \mathbf{x}^+, \mathbf{e})$;

7) $\mathbf{x} = \mathbf{x}^+$;

8) volta a 2;

3.2.1 Geração de variáveis aleatórias

A geração de eventos requer a geração de números aleatórios provenientes de diversas distribuições de probabilidade. Para tal são necessárias várias técnicas para as gerar. Em [12] são apresentadas várias técnicas que se explicam de seguida.

1. Transformada Inversa

notação:

X - variável aleatória;

$F(x) = P(X \leq x)$ - função distribuição acumulada de X ;

$U[0,1]$ - distribuição de probabilidade uniforme entre 0 e 1;

Como $F(x)$ é crescente e o contradomínio pertence a $[0,1]$ sendo U um número aleatório tirado de $U[0,1]$ existe uma equivalência de um para um entre U no contradomínio de $F(x)$ e um x no domínio, logo $U = F(X)$ e $F^{-1}(U)$ existe sempre e podemos usar $X = F^{-1}(U)$

2. Convolução

Tendo em conta o facto de que uma variável aleatória $X = \sum_{i=1}^N Y_i$ com $Y_i \sim G_i(x)$ independentes tem distribuição $F(x) = G_1(x) * \dots * G_N(x)$, um número x proveniente de um distribuição $G_1(x) * \dots * G_N(x)$ pode ser gerado somando os números y_i gerados dos vários $G_i(x)$

3. Composição

Um número x proveniente de $F(x) = \sum_{i=1}^N p_i \times F_i(x)$, $\sum_{i=1}^N p_i = 1$ e $p_i \geq 0$ pode ser pensado como vindo de $F_i(x)$ com probabilidade p_i logo pode ser gerado em dois passos:

- 1) gerar M com $P(M=i) = p_i$;
- 2) gerar x de $F_M(x)$;

3.2.2 Estimação de parâmetros

Em [12] é feita também uma explicação sobre a estimação de parâmetros em simulação e resume-se de seguida o que mais relevante para esta tese.

Numa simulação a estimação de parâmetros em geral é feita a partir de dados X_1, \dots, X_N que não são de variáveis aleatórias independentes. Neste caso, se quisermos estimar a média da distribuição de X através da variável aleatória que é a média amostral podemos pois esta continua a ser um estimador não enviesado.

Se quisermos, no entanto, ter também uma estimativa dum qualquer intervalo de confiança para o estimador média amostral não podemos contar com a variância amostral pois esta será enviesada (visto não considerar a correlação entre os dados). Para contornar este problema podem-se correr várias simulações independentes (diferentes números aleatórios) com as mesmas condições iniciais. Iremos então obter várias estimativas para a média da distribuição X que serão independentes e das quais se poderá usar a média e variância amostral para se obter um intervalo de confiança aproximado pelo resultado do teorema do limite central.

Visto que o teorema do limite central só é uma boa aproximação quando o número de amostras é suficientemente grande, em vez da distribuição normal que seria o limite no infinito para a variável $Z_n = \frac{\theta_n - E[\theta_n]}{\sqrt{Var[\theta_n]}}$ (θ_n estimador média amostral com n amostras) usa-se a distribuição de t-Student que seria a distribuição exacta se X fosse uma distribuição normal.

3.3 Optimização

Esta secção começa por fazer um breve resumo dos métodos de optimização considerados que, em geral, se podem consultar em [13] e depois descreve o algoritmo escolhido.

Muitos dos métodos de optimização multi-dimensionais requerem minimizações em linha segundo uma certa direcção de procura. Para esse efeito podem usar-se diferentes métodos de minimização em linha:

1. Procura uniforme

amostra a função em N pontos do segmento de recta considerado e retorna o ponto com o menor valor

2. Procura dicotômica

para o segmento de recta em consideração amostra dois pontos no meio do segmento separados por um valor pequeno. Dependendo de qual dos dois é maior conclui-se em que metade está o mínimo (do lado do menor) e reduz-se o segmento a essa metade repetindo-se até se atingir o intervalo de incerteza que se queira atingir.

3. Golden Section

tal como na procura dicotômica usam-se dois pontos mas com o intuito de reduzir o número de pontos amostrados faz-se coincidir esses pontos com um dos pontos da iteração seguinte do algoritmo, dando uma proporção $\sim 0,618$ entre o segmento inicial e o da seguinte iteração que é a proporção do número de ouro donde surge o nome do algoritmo.

4. Fibonacci

com o mesmo objectivo do golden section de reduzir o número de pontos da função amostrados mas garantindo uma convergência mais rápida (logo menos iterações) esses dois pontos são calculados com base nas proporções entre números de Fibonacci $\frac{F_n}{F_{n+1}}$ e $\frac{F_{n-1}}{F_{n+1}}$ (exige no entanto o cálculo dos números de Fibonacci que para N grandes tornam-se grandes de mais para a representação dum computador.

Existem vários métodos de optimização com diversas variações. Tenta-se de seguida e de maneira concisa dar uma visão geral dos métodos que usam a informação do gradiente (que estará disponível graças à IPA) tomados em consideração no processo de decisão.

1. Métodos de gradiente

usam a recursão:

$$x_{k+1} = x_k - \alpha_k D_k \nabla f(x_k)$$

em que $\alpha_k = \underset{\alpha}{\operatorname{arg\,min}} f(x_k - \alpha D_k \nabla f(x_k))$

e $D_k = D_k^T > 0$

Algoritmos:

com $D_k = I$ é "Steepest Descent"

com $D_k = (\nabla^2 f(x_k))^{-1}$ é Newton

com $D_k =$ aproximação de $(\nabla^2 f(x_k))^{-1}$ são Quase-Newton

2. Métodos de direcções conjugadas

começa com $d_0 = -\nabla f(x_0)$

usam a recursão:

$$x_{k+1} = x_k + \alpha_k d_k$$

em que $\alpha_k = \frac{-\nabla f(x_k)^T d_k}{d_k^T Q d_k}$

e $Q > 0$ e $d_i^T Q d_j = 0, \forall i \neq j$

retornando a d_0 quando $k =$ numero de dimensões de x

Algoritmos:

com $Q = (\nabla^2 f(x_k))^{-1}$ é gradiente conjugado e $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$ com

$$\beta_k = \frac{\nabla f(x_k)^T Q d_k}{d_k^T Q d_k}$$

Quando não se tem $(\nabla^2 f(x_k))^{-1}$ $\alpha_k = \arg \min_{\alpha} f(x_k - \alpha d_k)$ e usam-se aproximações

para β_k

Algoritmos:

com $\beta_k = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)}$ é Fletcher-Reeves

Com Restrições

1. Funções Barreira

Para que os algoritmos não saiam das restrições do problema usam-se funções que vão para infinito quando se aproxima do contorno de restrição, chamadas funções de barreira.

Em vez de $f(x)$, otimiza-se:

$$f(x) + \epsilon_k B(x)$$

com $0 < \epsilon_{k+1} < \epsilon_k$ e $\epsilon_k \rightarrow 0$ com $k \rightarrow \infty$.

Em que $B(x)$ é a função barreira construída a partir das restrições do problema.

2. Gradiente projectado

Em vez de se tomar a direcção indicada toma-se a direcção da projecção do ponto $x^k + d^k$ projectado para dentro das restrições.

Escolha do Algoritmo

Como a função a otimizar não é convexa, não se tem sempre $(\nabla^2 f(x_k))^{-1} > 0$ logo não se podem usar algoritmos que necessitem que tal se verifique. Escolheu-se portanto o algoritmo de Fletcher-Reeves que tem normalmente convergência mais rápida que o "Steepest Descent".

Visto as restrições (com os níveis de base-stock representados nas variáveis alternativas) serem do tipo $x_i > a$ ou $x_i < a$ com a constante e x_i a i -ésima coordenada do vector x , a projecção é muito simples de fazer e foi por isso escolhido o gradiente projectado para impôr as restrições.

A convergência para um mínimo local está provada para o "Steepest Descent" e como o algoritmo tem um passo de gradiente cada n iterações e a minimização em linha nos restantes passos garante que os passos nunca aumentam a função de custo fica também garantida a convergência para o algoritmo de Fletcher-Reeves com direcção projectada.

Para a minimização em linha necessária ao algoritmo escolheu-se o método golden section visto este reduzir o número de pontos amostrados (que no caso desta tese são simulações pesadas) e não requer o cálculo dos números de Fibonacci que se pode tornar incomportável para intervalos de incerteza muito pequenos (i.e. número de iterações elevado).

4. Políticas de Produção

Este capítulo encontra-se dividido em duas secções, apresenta a aplicação de algumas políticas de base-stock simples ao modelo desta tese e introduz a política em estudo nesta tese, comparando-a às anteriores. Na primeira secção começa-se por apresentar a notação e equações gerais às políticas de base-stock, seguindo-se a descrição das prioridades estritas e da LSR e finaliza-se com as derivadas necessárias à IPA. Na segunda secção começa-se por introduzir a política de prioridades suavizadas com a sua notação, as suas equações e as derivadas necessárias à IPA, continuando-se com a comparação da mesma à LSR, no caso geral, e às prioridades estritas, no caso de dois produtos.

4.1 Políticas de Base-Stock

Uma política de base-stock tenta levar o stock de cada produto para um determinado valor, o seu nível de base-stock

Notação:

- $z^{p,f}$ - nível escalão de base-stock do produto de fase f do produto final ($z^{p,f} \geq z^{p,f+1}$);
- $\Delta^{p,f}$ - conjunto de variáveis alternativas (a $z^{p,f}$) que relacionam o inventário entre fases ($\Delta^{p,f} > 0$);
- $y_t^{p,f}$ - shortfall do produto de fase f do produto final p ;
- $f_t^{p,f}$ - necessidade de produção (o que produziríamos se não houvesse restrição de capacidade) do produto de fase f do produto final p .

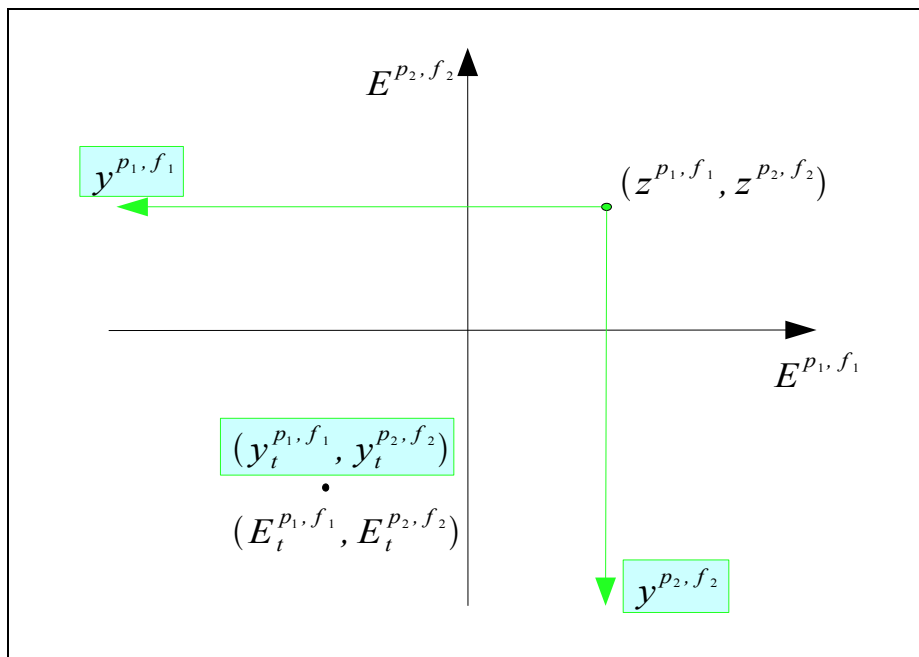


Figura 4.1.1 - Gráfico relacionando o sistemas de coordenadas dos inventários escalão com o sistema de coordenadas dos shortfalls

Equações:

$$\Delta^{p,f} = z^{p,f} - z^{p,f+1}, \quad z^{p,F_p+1} = 0$$

$$y_t^{p,f} = \max(z^{p,f} - E_t^{p,f}, 0)$$

$$f_t^{p,f} = \min(y_t^{p,f}, I_t^{p,f-1})$$

As políticas de base-stock decidem produção dentro das seguintes restrições:

$$0 \leq P_t^{p,f} \leq f_t^{p,f}$$

$$\sum_{(i,j) \in M} P_t^{i,j} \tau^{i,j} = K_m \text{ se } \sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j} > K_m$$

e $P_t^{p,f} = f_t^{p,f}$, $p, f \in M_m$ quando $\sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j} \leq K_m$.

Descrição de algumas políticas comuns simples (o que acontece se $\sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j} > K_m$):

- prioridade estrita:

cada p, f tem uma prioridade e atribui $P_t^{p,f} = f_t^{p,f}$, $(p, f) \in M_m$ por ordem de prioridade até esgotar a capacidade da máquina m .

- LSR:

$P_t^{p,f}$ é linearmente proporcional a $f_t^{p,f}$ com factor de proporcionalidade

$$\frac{K_m}{\sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j}}, \text{ ou seja, } P_t^{p,f} = f_t^{p,f} \frac{K_m}{\sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j}} \text{ se } \sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j} > K_m$$

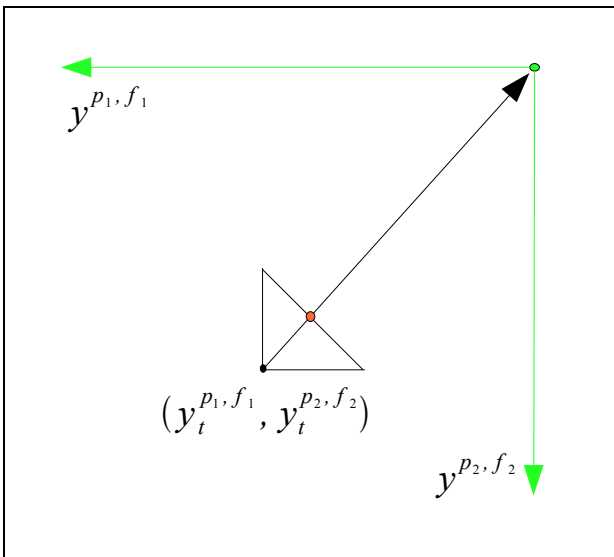


Figura 4.1.2 - Ilustração do funcionamento da LSR

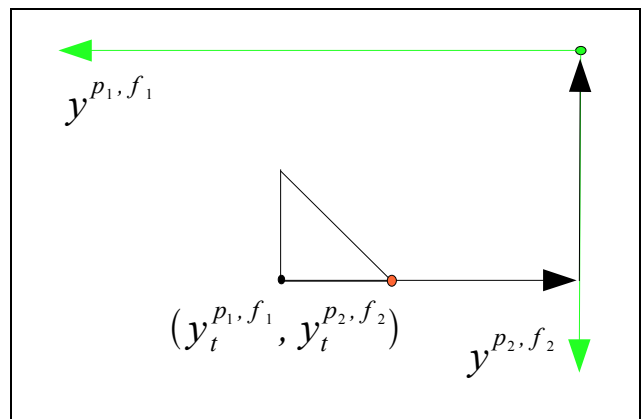


Figura 4.1.3 - Ilustração do funcionamento das prioridades estritas

Derivadas da necessidade de produção (necessárias à IPA):

$$\frac{\delta f_t^{p,f}}{\delta x} = \begin{cases} 0, & f_t^{p,f} = 0 \\ \sum_{i=f}^{F_p} \frac{\delta \Delta^{p,i}}{\delta x} - \sum_{i=f}^{F_p} \frac{\delta I_t^{p,i}}{\delta x}, & y_t^{p,f} < I_t^{p,f-1} \wedge f_t^{p,f} > 0 \\ \frac{\delta I_t^{p,f-1}}{\delta x}, & y_t^{p,f} > I_t^{p,f-1} \wedge f_t^{p,f} > 0 \end{cases},$$

com $I_t^{p,0} = \infty$ e $\frac{\delta I_t^{p,0}}{\delta x} = 0$

Derivadas da produção para prioridade estrita (necessárias à IPA):

$$\frac{\delta P_t^{p,f}}{\delta x} = \begin{cases} \frac{\delta f_t^{p,f}}{\delta x}, & \sum_{(i,j)=\text{mais prioritário} \in M_m}^{p,f} f_t^{i,j} \tau^{i,j} < K_m \\ \sum_{(i,j)=\text{mais prioritário} \in M_m}^{(p,f)-1} \frac{\delta P_t^{i,j}}{\delta x}, & \sum_{(i,j)=\text{mais prioritário} \in M_m}^{p,f} f_t^{i,j} \tau^{i,j} > K_m \end{cases}$$

Notas Finais:

- A validação da IPA para estes casos é apresentada em [9].
- Os casos da igualdade que não estão definidos nas equações de IPA têm probabilidade de ocorrência 0 visto que se assume que a procura é contínua e a distribuição da procura não tem pontos de massa

4.2 Prioridades Suavizadas

4.2.1 Introdução da Política

As Prioridades Suavizadas são uma política de base-stock que tem duas fases de produção em que se aloca a capacidade existente segundo a LSR. A cada produto de fase atribui-se um alfa ($0 \leq \alpha \leq 1$) que indica a quantidade relativa do que seria necessário produzir (se não existisse limite de capacidade) que entra na primeira fase ($1 - \alpha$ entrarão na segunda).

Notação:

$\alpha^{p,f}$ - parâmetro α do produto de fase f do produto final p .

Equações:

$$P_{t,I}^{p,f} = \alpha^{p,f} f_t^{p,f} \min\left(\frac{K_m}{\sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j}}, 1\right)$$

$$P_{t,II}^{p,f} = (1 - \alpha^{p,f}) f_t^{p,f} \min\left(\frac{K_m - \sum_{(i,j) \in M_m} P_{t,I}^{i,j} \tau^{i,j}}{\sum_{(i,j) \in M_m} (1 - \alpha^{i,j}) f_t^{i,j} \tau^{i,j}}, 1\right)$$

$$P_t^{p,f} = P_{t,I}^{p,f} + P_{t,II}^{p,f}$$

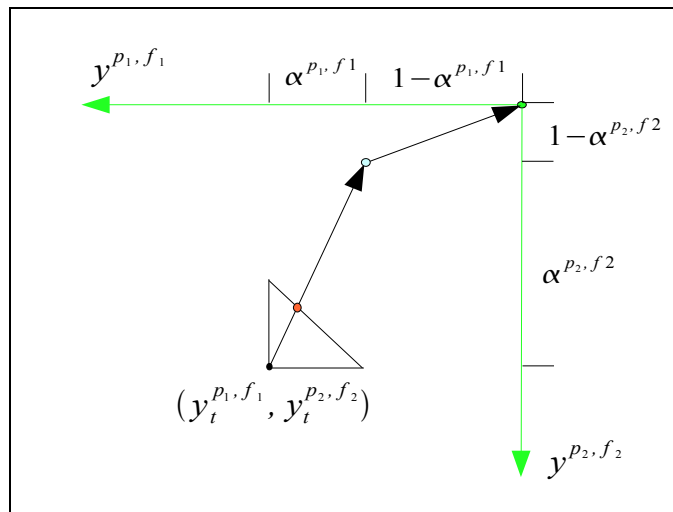


Figura 4.2.1.1 - Ilustração do funcionamento das prioridades suavizadas

IPA derivadas da produção para as prioridades suavizadas:

$$\frac{\delta P_t^{p,f}}{\delta x} = \frac{\delta P_{t,I}^{p,f}}{\delta x} + \frac{\delta P_{t,II}^{p,f}}{\delta x}$$

$$\frac{\delta P_{t,I}^{p,f}}{\delta x} = \begin{cases} \frac{\delta \alpha^{p,f}}{\delta x} f_t^{p,f} + \alpha^{p,f} \frac{\delta f_t^{p,f}}{\delta x} = A_{t,I}^{p,f}(x), & \sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j} < K_m \\ K_m \frac{A_{t,I}^{p,f}(x) \sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j} - \alpha^{p,f} f_t^{p,f} \sum_{(i,j) \in M_m} A_{t,I}^{i,j}(x) \tau^{i,j}}{(\sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j})^2}, & \sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j} > K_m \end{cases}$$

$$\frac{\delta P_{t,II}^{p,f}}{\delta x} = \begin{cases} -\frac{\delta \alpha^{p,f}}{\delta x} f_t^{p,f} + (1-\alpha^{p,f}) \frac{\delta f_t^{p,f}}{\delta x} = A_{t,II}^{p,f}(x), & \sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j} < K_m \wedge \sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j} < K_m \\ A_{t,II}^{p,f}(x) \frac{K_m - \sum_{(i,j) \in M_m} P_{t,I}^{i,j} \tau^{i,j}}{\sum_{(i,j) \in M_m} (1-\alpha^{i,j}) f_t^{i,j} \tau^{i,j}} + (1-\alpha^{p,f}) f_t^{p,f} \frac{B_{t,I}^{p,f}(x) - B_{t,II}^{p,f}(x)}{(\sum_{(i,j) \in M_m} (1-\alpha^{i,j}) f_t^{i,j} \tau^{i,j})^2}, & \sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j} < K_m \wedge \sum_{(i,j) \in M_m} f_t^{i,j} \tau^{i,j} > K_m \end{cases}$$

$$B_{t,I}^{p,f}(x) = \frac{\delta(K_m - \sum_{(i,j) \in M_m} P_{t,I}^{i,j} \tau^{i,j})}{\delta x} \times \sum_{(i,j) \in M_m} (1-\alpha^{i,j}) f_t^{i,j} \tau^{i,j} = -(\sum_{(i,j) \in M_m} A_{t,I}^{i,j}(x) \tau^{i,j}) (\sum_{(i,j) \in M_m} (1-\alpha^{i,j}) f_t^{i,j} \tau^{i,j})$$

$$B_{t,II}^{p,f}(x) = (K_m - \sum_{(i,j) \in M_m} P_{t,I}^{i,j} \tau^{i,j}) \times \frac{\delta(\sum_{(i,j) \in M_m} (1-\alpha^{i,j}) f_t^{i,j} \tau^{i,j})}{\delta x} = (K_m - \sum_{(i,j) \in M_m} \alpha^{i,j} f_t^{i,j} \tau^{i,j}) (\sum_{(i,j) \in M_m} A_{t,II}^{i,j}(x) \tau^{i,j})$$

4.2.2 Comparação com LSR

De seguida prova-se que a LSR está contida na política de prioridades suavizadas (PS). LSR e prioridades suavizadas são ambas políticas de base-stock logo são iguais quando $\sum_{i \in M_m} f_t^i \tau^i \leq K_m$ e fazem $P_t^{p,f} = f_t^{p,f}$, $p, f \in M_m$. Quando $\sum_{i \in M_m} f_t^i \tau^i > K_m$:

LSR:

$$P_t^{p,f} = f_t^{p,f} \frac{K_m}{\sum_{i \in M_m} f_t^i \tau^i}$$

PS (incluindo $\sum_{i \in M_m} f_t^i \tau^i \leq K_m$):

$$P_{t,I}^{p,f} = \alpha^{p,f} f_t^{p,f} \min\left(\frac{K_m}{\sum_{i \in M_m} \alpha^i * f_t^i \tau^i}, 1\right)$$

$$P_{t,II}^{p,f} = (1-\alpha^{p,f}) f_t^{p,f} \min\left(\frac{K_m - \sum_{i \in M_m} P_{t,I}^i \tau^i}{\sum_{i \in M_m} (1-\alpha^i) f_t^i \tau^i}, 1\right)$$

$$P_t^{p,f} = P_{t,I}^{p,f} + P_{t,II}^{p,f}$$

Desfazendo-nos dos mínimos fazendo a disjunção e retirando os termos comuns a todas as políticas de base-stock as PS :

$$P_t^{PS, p, f} = \alpha^{p, f} f_t^{p, f} + (1 - \alpha^{p, f}) f_t^{p, f} \frac{K_m - \sum_{i \in M_m} \alpha^i f_t^i \tau^i}{\sum_{i \in M_m} (1 - \alpha^i) f_t^i \tau^i} \text{ se } \sum_{i \in M_m} \alpha^i f_t^i \tau^i \leq K_m$$

$$P_t^{PS, p, f} = \alpha^{p, f} f_t^{p, f} \frac{K_m}{\sum_{i \in M_m} \alpha^i f_t^i \tau^i} \text{ se } \sum_{i \in M_m} \alpha^i f_t^i \tau^i > K_m$$

fazendo $\alpha^i = \alpha_m$, $\forall i \in M_m$ temos:

$$P_t^{PS, p, f} = \alpha_m f_t^{p, f} + (1 - \alpha_m) f_t^{p, f} \frac{K_m - \sum_{i \in M_m} \alpha_m f_t^i \tau^i}{\sum_{i \in M_m} (1 - \alpha_m) f_t^i \tau^i} = \alpha_m f_t^{p, f} + (1 - \alpha_m) f_t^{p, f} \frac{K_m - \alpha_m \sum_{i \in M_m} f_t^i \tau^i}{(1 - \alpha_m) \sum_{i \in M_m} f_t^i \tau^i} =$$

$$= \alpha_m f_t^{p, f} + f_t^{p, f} \frac{K_m - \alpha_m \sum_{i \in M_m} f_t^i \tau^i}{\sum_{i \in M_m} f_t^i \tau^i} = \alpha_m f_t^{p, f} + f_t^{p, f} \frac{K_m}{\sum_{i \in M_m} f_t^i \tau^i} - f_t^{p, f} \frac{\alpha_m \sum_{i \in M_m} f_t^i \tau^i}{\sum_{i \in M_m} f_t^i \tau^i} =$$

$$= \alpha_m f_t^{p, f} + f_t^{p, f} \frac{K_m}{\sum_{i \in M_m} f_t^i \tau^i} - f_t^{p, f} \alpha_m = f_t^{p, f} \frac{K_m}{\sum_{i \in M_m} f_t^i \tau^i} \text{ se } \sum_{i \in M_m} \alpha_m f_t^i \tau^i \leq K_m$$

$$P_t^{PS, p, f} = \alpha_m f_t^{p, f} \frac{K_m}{\sum_{i \in M_m} \alpha_m f_t^i \tau^i} = \alpha_m f_t^{p, f} \frac{K_m}{\alpha_m \sum_{i \in M_m} f_t^i \tau^i} = f_t^{p, f} \frac{K_m}{\sum_{i \in M_m} f_t^i \tau^i} \text{ se } \sum_{i \in M_m} \alpha_m f_t^i \tau^i > K_m$$

$$P_t^{PS, p, f} = f_t^{p, f} \frac{K_m}{\sum_{i \in M_m} f_t^i \tau^i} = P_t^{LSR, p, f}$$

Fica portanto provado que LSR \subset PS visto que LSR = PS (com alfas todos iguais)

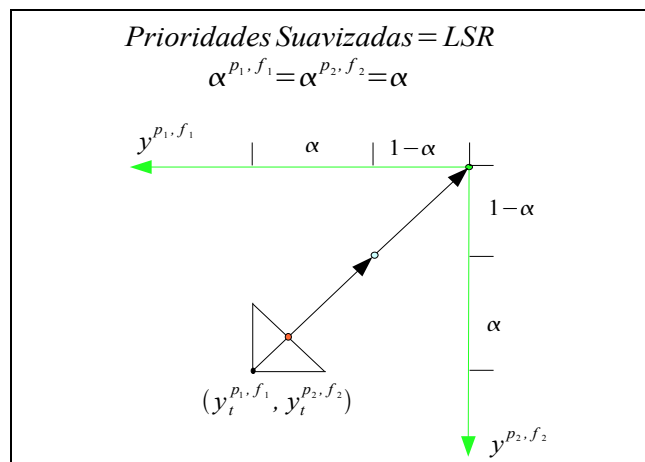


Figura 4.2.2.1 - Ilustração gráfica da prova (LSR)

4.2.3 Comparação com Prioridades Estritas

De seguida prova-se para o caso de dois produtos que prioridades estritas (PE) está contida nas prioridades suavizadas (PS):

$$P_t^{PE,j} = \min\left(f_t^j, \frac{K_m - \sum_{i=0}^{j-1} P_t^{PE,i} \tau^i}{\tau^j}\right) \text{ com } j \text{ ordenado por prioridades (1 mais prioritário) e } P_t^{PE,0} = 0$$

$$P_{t,I}^{PS,p,f} = \alpha^{p,f} f_t^{p,f} \min\left(\frac{K_m}{\sum_{i \in M_m} \alpha^i f_t^i \tau^i}, 1\right)$$

$$P_{t,II}^{PS,p,f} = (1 - \alpha^{p,f}) f_t^{p,f} \min\left(\frac{K_m - \sum_{i \in M_m} P_{t,I}^{PS,i} \tau^i}{\sum_{i \in M_m} (1 - \alpha^i) f_t^i \tau^i}, 1\right)$$

$$P_t^{PS,p,f} = P_{t,I}^{PS,p,f} + P_{t,II}^{PS,p,f}$$

PE com apenas dois produtos ficam:

$$P_t^{PE,1} = \min\left(f_t^1, \frac{K_m}{\tau^1}\right)$$

$$P_t^{PE,2} = \min\left(f_t^2, \frac{K_m - P_t^{PE,1} \tau^1}{\tau^2}\right)$$

PS com apenas dois produtos e fazendo $\alpha^1 = 1 \wedge \alpha^2 = 0$ ficam:

$$P_{t,I}^{PS,1} = f_t^1 \min\left(\frac{K_m}{f_t^1 \tau^1}, 1\right) \quad P_{t,II}^{PS,1} = 0 \quad P_t^{PS,1} = \min\left(\frac{K_m}{\tau^1}, f_t^1\right) = P_t^{PE,1}$$

$$P_{t,I}^{PS,2} = 0 \quad P_{t,II}^{PS,2} = f_t^2 \min\left(\frac{K_m - P_t^{PS,1} \tau^1}{f_t^2 \tau^2}, 1\right) \quad P_t^{PS,2} = \min\left(\frac{K_m - P_t^{PS,1} \tau^1}{\tau^2}, f_t^2\right) = P_t^{PE,2}$$

Fica portanto provado que para dois produtos $PE \subset PS$ visto que $PE = PS(\text{com } \alpha^1 = 1 \wedge \alpha^2 = 0)$. É de notar também que caso se estendam as prioridades suavizadas para terem tantas fases quanto os produtos que existem na máquina ter-se-ia que $PE \subset PS$ com qualquer número de produtos.

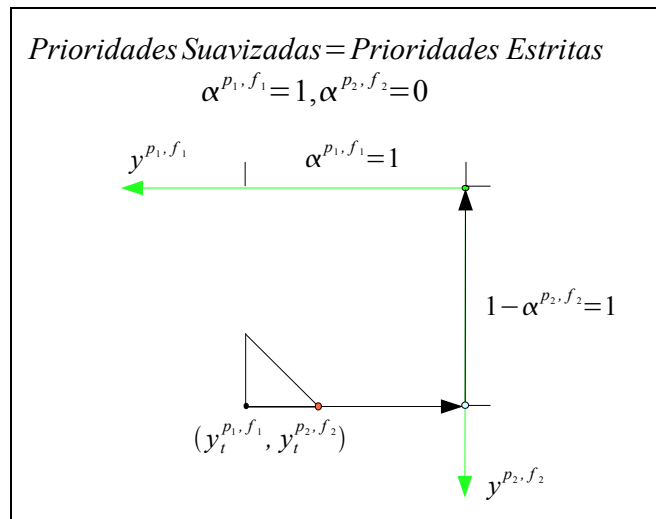


Figura 4.2.3.1 - Ilustração gráfica da prova (PE)

PARTE II: Trabalho Prático

5. Software

Este capítulo encontra-se dividido em três secções e apresenta de forma resumida a estrutura e a função do código desenvolvido. A estrutura encontra-se, no entanto, expressa de forma mais completa nos diagramas UML iniciais do projecto no anexo A desta tese.

5.1 Modelo do Sistema

namespace FactoryModel

Conjunto das classes necessárias à implementação do modelo do sistema de produção adoptado nesta tese

Factory contém a representação do modelo

Tem um vector de Machine que representa as máquinas do modelo.

Tem um vector de FinalProduct que representa as linhas de produção existentes na fábrica com um vector de Products que representa um produto de fase.

Tem um Control que representa uma política de controlo genérica que se está a aplicar na fábrica.

As políticas implementadas são de base-stock derivando da classe EchelonBaseStockControl derivada de Control. As prioridades estritas estão implementadas na classe StrictPriorities e as prioridades suaves estão implementadas na classe SoftPriorities.

MakeFromXML: funções que permitem criar os vários componentes do modelo a partir de ficheiros xml usando as funções "open source" tinyXML para manipulação dos ficheiros. A estrutura dos ficheiros xml tem de ser a indicada nos seguintes ficheiros dtd:

controlvariables.dtd - parâmetros necessários às políticas de prioridade estrita e prioridade suavizada

factory.dtd - máquinas existentes na fábrica;

products.dtd - linhas de série dos produtos que se querem produzir na fábrica;

costparameters.dtd - custos de holding e backlog associados aos produtos;

demand.dtd - distribuições da procura dos vários produtos.

5.2 Simulação

namespace DESSimulation

Conjunto das classes necessárias à implementação do simulador descrito na parte teórica desta tese.

DESSimulator implementa o ciclo principal do simulador:

Tem um double a representar o tempo actual;

Tem um Model que é uma interface para qualquer modelo que se deseje simular;

Tem um EventGenerator que é uma interface para qualquer gerador de eventos, do tipo genérico Event, que se deseje para o simulador;

Tem um State que é uma interface para qualquer representação do estado que se queira usar na simulação;

Tem um vector de Evaluator que é uma interface para qualquer tipo de avaliação que se queira ir fazendo ao longo da simulação.

namespace PDFRandomGeneration

Conjunto das classes que implementam os geradores de números aleatórios com probabilidades correspondentes a diferentes funções densidade de probabilidade.

PDFRandomGenerator interface que representa qualquer gerador de números aleatórios a partir de funções densidade de probabilidade:

Tem um double que representa a média e outro correspondente ao último número aleatório gerado.

FixedValueGenerator implementa um PDFRandomGenerator que gera sempre um número igual à média.

ExponentialPDFGenerator implementa um PDFRandomGenerator que gera números com base na distribuição exponencial com parâmetro igual à média.

PDFConvolution implementa um PDFRandomGenerator que gera números com base nos valores gerados por vários PDFRandomGenerator correspondendo a uma correlação.

namespace FactoryDESInterface

Conjunto das classes que implementam as interfaces necessárias para usar o modelo da fábrica com o simulador.

FactoryS implementa um State e contém um vector de vectores em que a posição [p][f] corresponde ao stock do produto de fase $f-1$ do produto final $p-1$.

FactoryE implementa um Event e contém um vector com a procura dos produtos.

FactoryM implementa um Model e usa uma Factory com o respectivo controlo para determinar qual será o estado seguinte se se começar com o estado inicial e ocorrer o evento dado.

FactoryEG implementa um EventGenerator e usa um vector de PDFRandomGenerator para criar um evento do tipo FactoryE.

FactoryAveragegCostEvaluator implementa um Evaluator que vai calculando o custo médio da simulação a partir dos parâmetros de custo dos produtos.

FactorySoftPAverageCostGradientEvaluator implementa um Evaluator que vai calculando o gradiente do custo médio a partir das formulas da IPA para as prioridades suaves, dos parâmetros de custo dos produtos e da Factory.

5.3 Optimização

auxiliarVectorAlgebra: contém funções que implementam as operações de álgebra vectoriais necessárias ao optimizador.

namespace Optimization

Conjunto das classes necessárias à implementação do algoritmo de optimização escolhido na parte teórica desta tese.

- `Optimizer` implementa o ciclo principal do otimizador:
 - Tem um `LineOptimizer` que é uma interface para qualquer algoritmo de optimização em linha que se queira usar na optimização;
 - Tem um `DirectionChooser` que é uma interface para qualquer algoritmo que escolha a próxima direcção de procura;
 - Tem um `MinimalStepDeterminator` que é uma interface para qualquer algoritmo que determine o valor mínimo tamanho do intervalo de procura em linha (sem projecção);
 - Tem um `StepDeterminator` que é uma interface para qualquer algoritmo que determine o tamanho do intervalo de procura em linha;
 - Tem um `Bounds` que é uma interface para qualquer objecto que permita impôr limites ao domínio da procura e projectar qualquer ponto fora dele para o seu limite;
 - Tem um `Terminator` que é uma interface para qualquer algoritmo que decida quando a optimização deve terminar;
 - Tem um `FunctionToOptimize` que é uma interface para qualquer função que se queira otimizar.

- `AdaptingSteps` implementa um `MinimalStepDeterminator` e o algoritmo tenta fazer com que o intervalo mínimo escolhido seja o mais próximo possível do que se irá escolher depois multiplicando ou dividindo o valor anterior por um factor de adaptação;

- `BinaryEscalation` implementa um `StepDeterminator` verifica se o passo mínimo corresponde a um aumento da função de custo se tal não acontecer dobra o passo até que tal aconteça ou se o passo sair dos limites do domínio de optimização fazendo depois uma procura binária pelo ponto limite interior;

- `BoxBounds` implementa um `Bounds` que impõe limites sobre o domínio de procura do tipo $x_i > k$ ou $x_i < k$ em que a projecção é simplesmente $x_i = k + \delta$ ou $x_i = k - \delta$;

- `FletcherReeves` implementa um `DirectionChooser` que usa a fórmula de Fletcher-Reeves para escolher a direcção seguinte com um passo de gradiente cada N passos com N igual a dimensão do espaço de procura;

- `GoldenSection` implementa um `LineOptimizer` correspondente ao algoritmo de procura em linha golden section;

- `nDiffTolerance` implementa um `Terminator` que termina a simulação quando a diferença entre o valor no ponto actual e o valor há N passos atrás é menor que um determinado valor.

`SimulationOptimizationInterface`: contém classes que permitem otimizar os parâmetros das políticas de controlo da fábrica simulada;

- `SoftSimulationFunction` implementa um `FunctionToOptimize` usando o simulador e os avaliadores do custo e gradiente médios para calcular o valor médio do custo e o gradiente (ou só o custo) para os parâmetros da política de prioridades suaves de entrada.

6. Comparação com Prioridades Estritas

Este capítulo encontra-se dividido em quatro secções e estuda a relevância das prioridades suavizadas confrontando-as com a regra simples mais utilizada, i.e., as prioridades estritas. Na primeira secção é utilizado o novo modelo e software para confirmar alguns resultados anteriores apresentados em [10]. Na segunda secção usam-se vários custos num caso simples e retiram-se várias conclusões práticas. Na terceira secção explora-se um caso com a estrutura apresentada na Figura 3.1.1 fazendo o paralelo dos resultados da optimização com os apresentados em [14]. Na quarta secção é explorada a estrutura da função de custo e é tentada a extrapolação para o caso geral.

6.1. Confirmação de Resultados Prévios

No capítulo 5.1 de [10] é apresentado um caso de aplicação em que, segundo as conclusões de [9], as prioridades estritas teriam melhores resultados do que a ESR e a LSR, visto que um dos produtos apresenta a mais baixa procura média, os maiores custos de backlog e menor variância. De modo a confirmar os resultados prévios e confirmar o bom funcionamento do optimizador tentou-se reproduzir o melhor possível o caso e utilizar o optimizador para obter os parâmetros das prioridades suavizadas óptimos e os níveis de base-stock óptimos para ambos os casos de prioridades estritas.

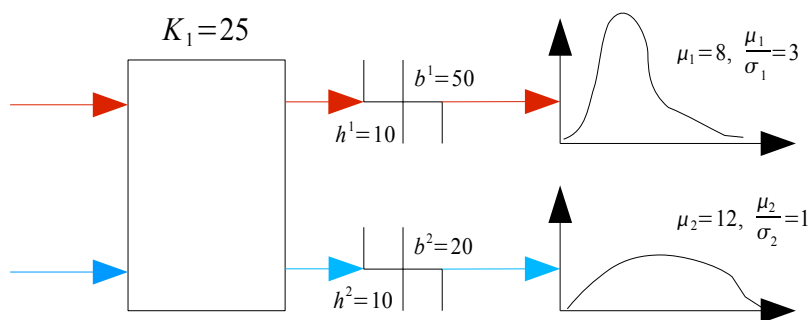


Figura 6.1.1 - Figura representativa do caso (resultados prévios)

O caso é o seguinte:

- 1 Máquina
- 2 Produtos Finais com 1 fase
- Capacidade da máquina = 25
- 1º produto final:
 - Média da procura = 8
 - Inverso do coeficiente de variância = 3
 - Custo de backlog = 50
 - Custo de holding = 10
 - Carga = 1
- 2º produto final:
 - Média da procura = 12
 - Inverso do coeficiente de variância = 1
 - Custo de backlog = 20
 - Custo de holding = 10
 - Carga = 1

Para obter as funções de distribuição de procura com as características descritas usaram-se distribuições exponenciais que têm inverso do coeficiente de variância igual a 1 e convoluíram-se n^2 distribuições idênticas com média $\frac{1}{n^2}$ vezes a média pretendida para obter distribuições com inversos do coeficiente de variância iguais a n . Os resultados obtidos são mostrados na tabela 6.1.1 juntamente com os resultados obtidos em [10].

Resultados da Optimização					Resultados Anteriores		
Δ^1	Δ^2	α^1	α^2	Custo	α^1	α^2	Custo
10.513	25.6131	1	0	299.484	1	0	340.624020
11.9863	23.764	0.54975	0.01064	292.810	0.413715	0.000060	330.286164
26.2372	16.1068	0	1	444.731	0	1	462.954474

Tabela 6.1.1 - Comparação com resultados anteriores

Embora os resultados sejam diferentes em termos numéricos as conclusões são similares, i.e., as prioridades suavizadas conseguem obter melhores resultados do que as prioridades estritas. As diferenças podem dever-se a diferenças nas distribuições de procura utilizadas ou no número de períodos de tempo da simulação sendo que nesta tese foram usados como número de períodos os que garantem um intervalo de confiança a 95% com uma variação do custo de 0,1%

6.2. Diferentes Custos

Pretende-se verificar se a desigualdade entre o óptimo das prioridades estritas e suavizadas (i.e. superioridade do óptimo das prioridades suavizadas) acontece na prática de forma relevante.

Para tal escolheu-se o caso simples de dois produtos (com a mesma distribuição de procura) numa máquina variando apenas os custos de holding e backlog e otimizando para as prioridades suavizadas e as duas possibilidades das prioridades estritas.

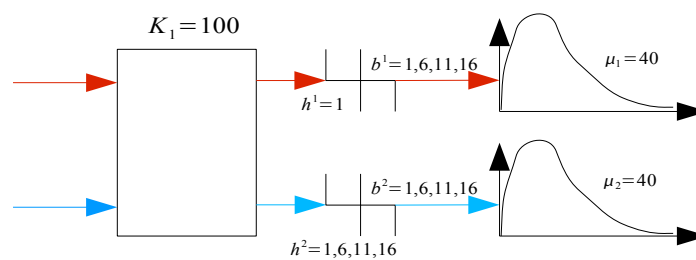


Figura 6.2.1 - Figura representativa do caso (diferentes custos)

Os dados específicos deste caso são:

- Capacidade da Máquina = 100
- 1º produto final:
 - Média da procura = 40
 - Distribuição de Procura Exponencial
 - Carga = 1
- 2º produto final:
 - Média da procura = 40
 - Distribuição de Procura Exponencial
 - Carga = 1

Manteve-se o custo de holding do primeiro produto fixo com valor 1 como referência. O custo de backlog do produto 1 e os custos de holding e backlog do produto 2 foram variados entre 1 e 16 (com intervalos de 5) dando assim os resultados apresentados na tabela do anexo C.

Para os casos em que os custos de holding e backlog são iguais as prioridades suavizadas apresentam sempre melhores resultados do que as prioridades estritas e são equivalentes ao uso da LSR. Tal era de esperar visto não existir qualquer factor diferenciador entre os dois produtos, sabemos também pelo apresentado em [11] que a WBR seria óptima (e visto as médias serem iguais também a ESR seria óptima).

Para os casos em que os custos de holding são iguais mas os de backlog são diferentes verifica-se também que as prioridades suavizadas apresentam sempre melhores resultados do que as prioridades estritas.

De um modo geral nota-se que a ordem das prioridades estritas com os melhores resultados são sempre as correspondentes a ordem dos alfas das prioridades suavizadas (i.e. se $\alpha_1 > \alpha_2$ então as prioridades estritas têm melhores resultados quando a prioridade do produto 1 é maior do que a do produto 2). Tal propriedade a verificar-se de um modo geral permitirá que se possa transformar o problema combinatório da determinação da lista de prioridades para a prioridade estrita no problema não linear de determinar os alfas nas prioridades suavizadas, tornando as prioridades suavizadas úteis, mesmo quando se deseja manter a extrema simplicidade das prioridades estritas na regra de produção.

Verifica-se também que a prioridade suavizada dos produtos se aproxima mais da prioridade estrita de um determinado produto sobre o outro quando os níveis de holding ou de backlog sobem para esse produto e para a prioridades estrita inversa quando estes descem.

Fica então demonstrado praticamente que, além das garantias teóricas de as prioridades suavizadas no máximo igualarem as prioridades estritas quando optimizadas, as prioridades suavizadas têm em variados casos melhores resultados que as prioridades estritas.

6.3. Sistema Complexo de 3 Máquinas e 2 Produtos Finais

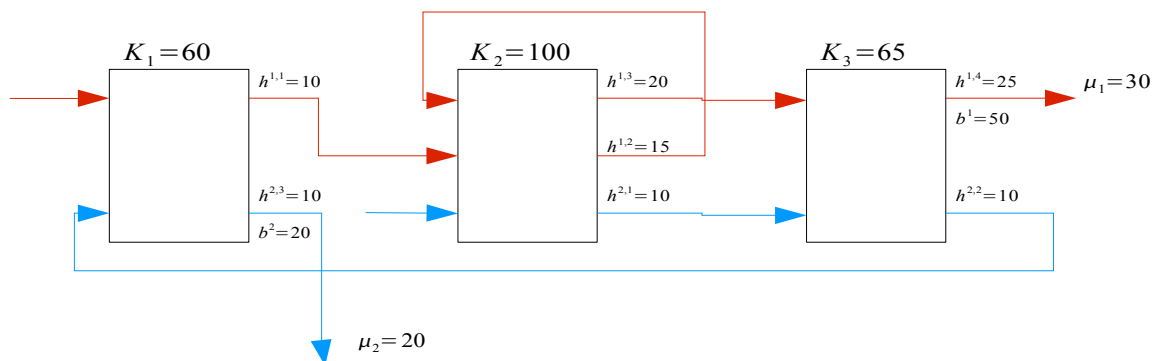


Figura 6.3.1 - Figura representativa do caso (sistema complexo)

De forma a explorar o uso das prioridades suavizadas num caso mais complexo tomou-se a estrutura para o sistema apresentada na Figura 6.3.1 (três máquinas e dois produtos finais) e com os seguintes dados:

- 3 Máquinas;
- Capacidades das Máquinas: $K_1 = 60$, $K_2 = 100$, $K_3 = 65$;
- 1º Produto Final:
 - Média de procura = 30;
 - Distribuição de Procura Exponencial;
 - 4 produtos de fase com holding costs: $h^{1,1} = 10$, $h^{1,2} = 15$, $h^{1,3} = 20$, $h^{1,4} = 25$;
 - custo de backlog = 50;

- 2º Produto Final:
 - Média de procura = 20;
 - Distribuição de Procura Exponencial;
 - 3 produtos de fase com holding costs: $h^{2,1}=10$, $h^{2,2}=10$, $h^{2,3}=10$;
 - custo de backlog = 20;
- Na primeira máquina são produzidos:
 - Produto de fase 1 do produto final 1;
 - Produto de fase 3 do produto final 2;
- Na segunda máquina são produzidos:
 - Produto de fase 2 do produto final 1;
 - Produto de fase 3 do produto final 1;
 - Produto de fase 1 do produto final 2;
- Na terceira máquina são produzidos:
 - Produto de fase 4 do produto final 1;
 - Produto de fase 2 do produto final 2.

Optimizou-se o sistema com todas as máquinas com prioridades suavizadas e, seguidamente, com a primeira máquina com prioridade estrita para o produto dois e as restantes máquinas com prioridades suavizadas. Obtiveram-se os resultados apresentados (arredondados às centésimas) na tabela 6.3.1 .

Resultados da Optimização														Custo
$\Delta^{1,1}$	$\Delta^{1,2}$	$\Delta^{1,3}$	$\Delta^{1,4}$	$\Delta^{2,1}$	$\Delta^{2,2}$	$\Delta^{2,3}$	$\alpha^{1,1}$	$\alpha^{1,2}$	$\alpha^{1,3}$	$\alpha^{1,4}$	$\alpha^{2,1}$	$\alpha^{2,2}$	$\alpha^{2,3}$	
40.78	39.13	36.74	89.17	4.03	43.71	52.70	0.45	0.59	1.00	0.99	0.20	0.00	0.99	4855.37
57.37	16.37	47.06	82.40	7.38	37.10	48.71	0.00	0.68	1.00	0.92	0.21	0.30	1.00	4927.03

Tabela 6.3.1 - Prioridades suavizadas vs prioridade estrita do produto 2 na máquina 1

Destes resultados pode-se ver que os parâmetros alfa dão prioridade ao produtos de fase mais próximos da procura o que faz lembrar a política de prioridades estritas "Last Buffer First Served" que segundo os resultados de [14] é a que obtém melhores resultados, dentro das políticas aí propostas, em termos de tempo de ciclo médio num sistema de estrutura re-entrante.

É de notar que a aplicação da política de prioridades estritas na máquina 1, quando as prioridades suavizadas apresentam parâmetros alfa com uma clara ordem de prioridade mas não de prioridade estrita, leva a um aumento do custo.

6.4. Estrutura da Função de Custo

De modo a poder especular sobre a estrutura da função de custo foram feitas várias simulações. Utilizou-se o caso de aplicação apresentado na secção 6.1 e simulou-se para base-stock constantes variando os alfa de 0 a 1 (em intervalos de 0.05) e traçaram-se os gráficos que se apresentam no anexo B. Utilizou-se também o caso mais complexo apresentado na secção 6.3 variando dois ou três parâmetros com os restantes iguais aos óptimos para as prioridades suavizadas, os gráficos apresentam-se também no anexo B.

Os gráficos B.1., B.2. e B.3. são resultado das simulações com os base-stock constantes correspondentes os das optimizações com prioridade do produto 1, prioridade do produto 2 e prioridades suavizadas da secção 6.1 desta tese. Os gráficos B.4 a B.19 são resultado das simulações sobre o caso da secção 6.3 com alguns parâmetros a variar e os restantes igual aos óptimos para as prioridades suavizadas.

Todos os gráficos são suaves, não apresentando quaisquer zonas de variação brusca, o que faz supor que se possa passar o mesmo no caso geral.

O gráfico que apresenta a curva mais complexa no caso de uma máquina só é B.2. mas uma inspeção aos valores (não apresentados nesta tese pelo elevado número) permite-nos concluir que tal como nos outros não existe nenhum mínimo local. Tal facto permite especular que a estrutura da função seja quasi-convexa pelo menos para o caso apresentado.

Para o caso de várias máquinas, embora a maior parte dos gráficos apresentem apenas um mínimo, nos gráficos da variação dos alfas numa só máquina verifica-se que a partir do valor para $\alpha^{1,3}=0.6$ até $\alpha^{1,3}=1$ parece existir, de forma cada vez mais acentuada, a possibilidade de mínimos locais.

De modo a confirmar que tal acontece geraram-se os gráficos B.12 e B.13, que são gráficos mais detalhados da zona problemática, que permitem afirmar que existem mínimos locais neste caso e portanto a função não pode ser quasi-convexa. Como de todos os casos este é o único de uma máquina com mais de dois produtos de fase mas também o único com re-entrância levanta a questão de se tal se deve à re-entrância ou se o mesmo acontece no caso de vários produtos sem re-entrância.

Os resultados práticos são portanto coerentes com as expectativas teóricas (não apresentadas visto estarem baseadas em especulação e saltos intuitivos e não em prova matemática) de que a função é contínua mas não com as expectativas de ser quasi-convexa. Tal quer dizer que podem existir mínimos locais e portanto a optimização baseada em gradiente pode não obter os parâmetros óptimos.

7. Conclusões e Trabalho Futuro

Esta tese apresentou um modelo de sistema que permite descrever linhas de produção com múltiplos produtos de uma forma geral que permite abranger os sistemas re-entrantes e série.

Foi desenvolvido software que implementa o funcionamento desses sistemas conjuntamente com um simulador e um otimizador de índole geral que foram integrados de forma a permitir a otimização baseada em simulação desses mesmos sistemas produtivos.

Foram apresentadas as equações que expressam o funcionamento do modelo e foi feita a adaptação de várias políticas de produção ao modelo.

A política das prioridades suavizadas, em estudo nesta tese, foi mostrada, de forma teórica, abranger a LSR, para qualquer número de produtos, e as prioridades estritas, para dois produtos, como casos particulares.

Fazendo uso do software desenvolvido foram confirmados resultados práticos anteriores, foi feita uma experiência com variação dos custo e outra com um sistema complexo e verificando-se que as prioridades suavizadas obtêm melhores resultados do que as prioridades estritas de maneira consistente. Estas experiências também permitiram reforçar a hipótese de a ordem dos alfas nas prioridades suavizadas se traduzir na ordenação óptima para as prioridades dos produtos nas prioridades estritas.

A estrutura da função de custo foi examinada revelando ser coerente com a hipótese de ser contínua mas, pelo menos para alguns casos, não com a hipótese de ser quasi-convexa.

Como trabalho futuro de interesse destaca-se:

- Verificar se a estrutura da função de custo se mantém quasi-convexa, como aparenta ser para dois produtos sem re-entrância, para mais produtos e máquinas sem re-entrância.
- Explorar praticamente os resultados das prioridades suavizadas quando comparada com as prioridades estritas para variadas situações de mais de dois produtos por máquina e mais do que uma máquina;
- Estudar a relevância da inclusão de um limite de inventário local nas várias políticas;
- A criação de uma interface gráfica para a visualização e criação das fábricas juntamente com a criação de um programa com interface gráfica que permitisse simular e otimizar as fábricas para variadas regras.

8. Bibliografia

- [1]: Paul Glasserman, *Gradient Estimation Via Perturbation Analysis*, Kluwer Academic Publishers ,1991
- [2]: A.J Clark and H. Scarf. *Optimal policies for a multi-echelon inventory problem*. Management Science 6:475-490, 1960
- [3]: Geert-Jan van Houtum. *Multiechelon Production/Inventory Systems: Optimal Policies, Heuristics, and Algorithms*. INFORMS 2006
- [4]: A. Federgruen and P.H. Zipkin. *An inventory model with limited production capacity and uncertain demands I: The average cost criterion*. Mathematics of Operations Research 11:193-207, 1986
- [5]: A. Federgruen and P.H. Zipkin. *An inventory model with limited production capacity and uncertain demands II: The discounted cost criterion*. Mathematics of Operations Research 11:208-216, 1986
- [6]: Rodney P. Parker and Roman Kapuscinski. *Optimal Policies for a Capacitated Two-Echelon Inventory System*. Operatons Research 52:739-755, 2004
- [7]: Paul Glasserman and Sridhar Tayur. *The stability of a capacitated, multi-echelon production-inventory system under a base-stock policy*. Operations Research 42:913-925, 1994
- [8]: Paul Glasserman and Sridhar Tayur. *Sensitivity Analysis for Base-stock Levels in Multiechelon Production-inventory Systems*. Management Science 41:263-281, 1995
- [9]: Carlos Filipe Gomes Bispo. *Re-Entrant Flow Lines*. Dissertation. Carnegie Mellon University,1997
- [10]: Célia Nunes, Isabel Sousa e Sofia Sousa. *Controlo de Inventário para Múltiplos Produtos*. Thesis. Instituto Superior Técnico-Universidade Técnica de Lisboa,1999
- [11]: Ganesh Janakiraman, Mahesh Nagarajan and Senthil Veeraraghavan. *Inventory Policies for Capacitated Systems with Multiple Products*. Working paper, March 9, 2007
- [12]: Christos G. Cassandras and Stéphane Lafortune, *Introduction to Discrete Event-Systems*, Springer Science+Business Media, Inc. ,1999
- [13]: David G. Luenberger, *Linear and Nonlinear Programming, Second Edition* , Springer ,2003
- [14]: Lu, S.H. and Kumar, P.R.. *Distributed Scheduling Based on Due-Dates and Buffer Priorities*. IEEE Trans. on Automatic Control 36(12):1406-1416, 1991

ANEXOS

A. Diagramas UML

Diagrama de Classes de FactoryModel

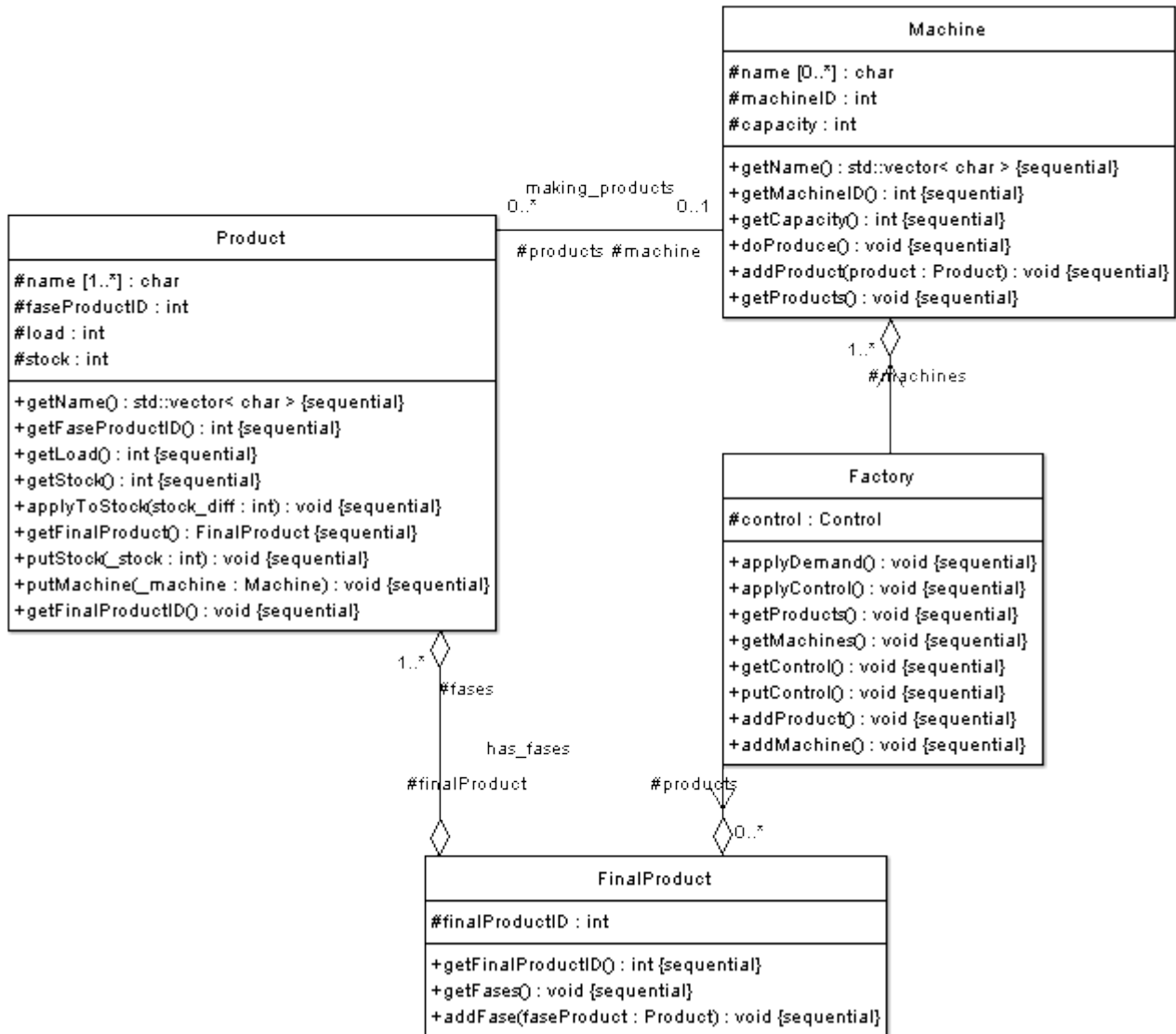


Diagrama de Classes de Factory::Control

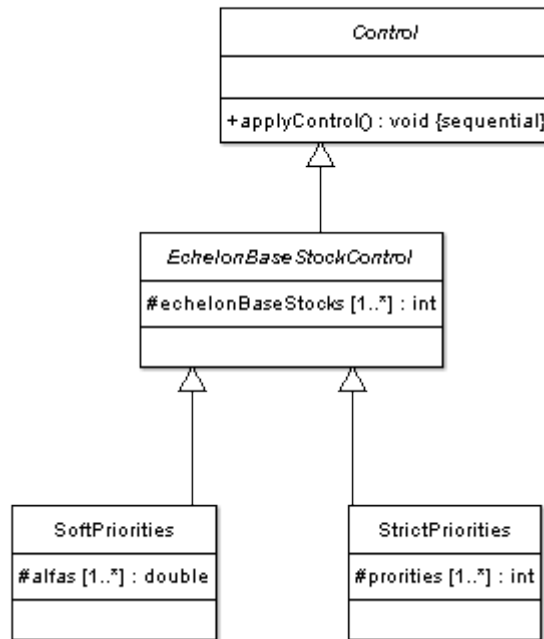


Diagrama de Classes de PDFRandomGeneration

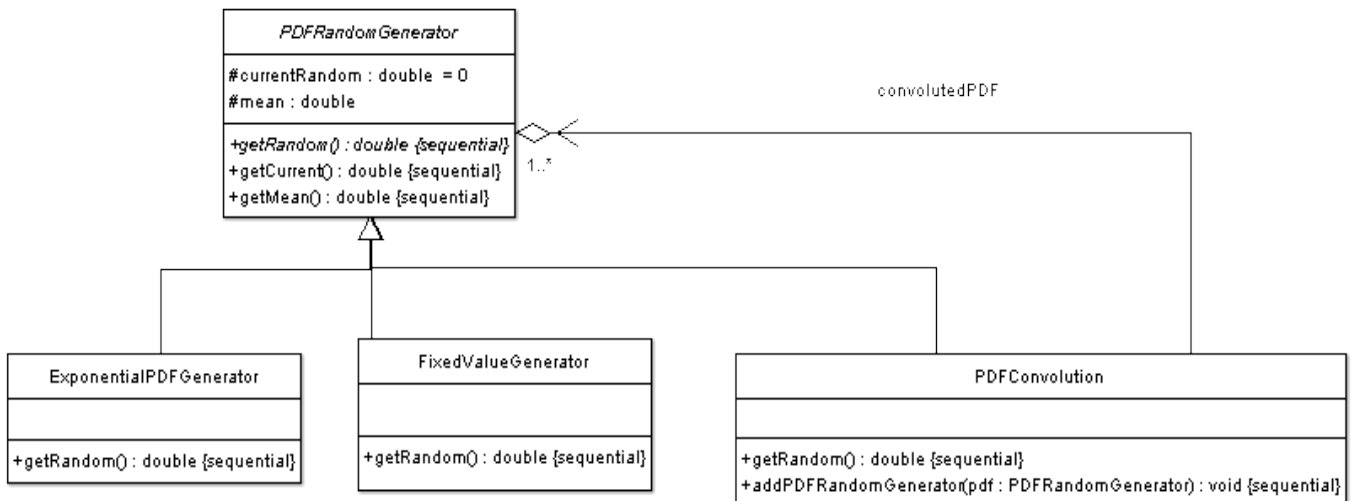


Diagrama de Classes de DESSimulation

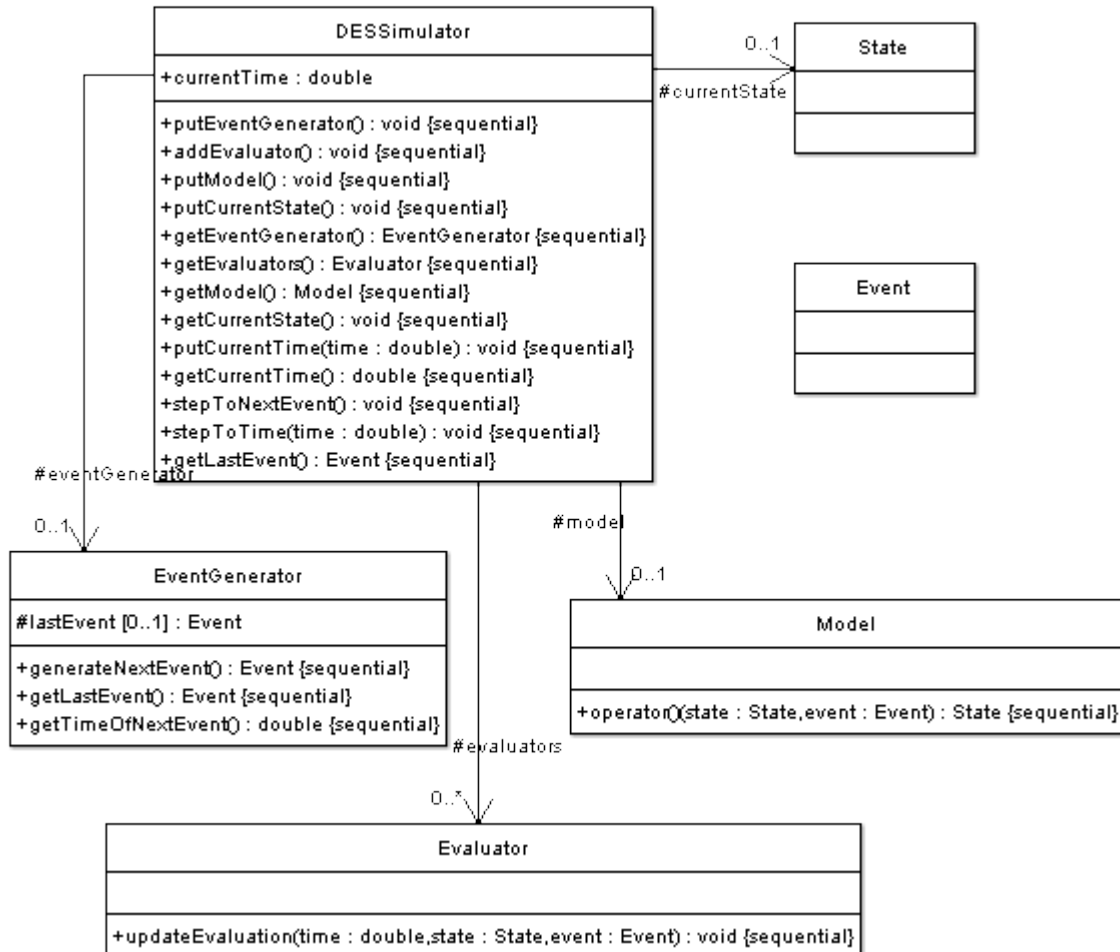
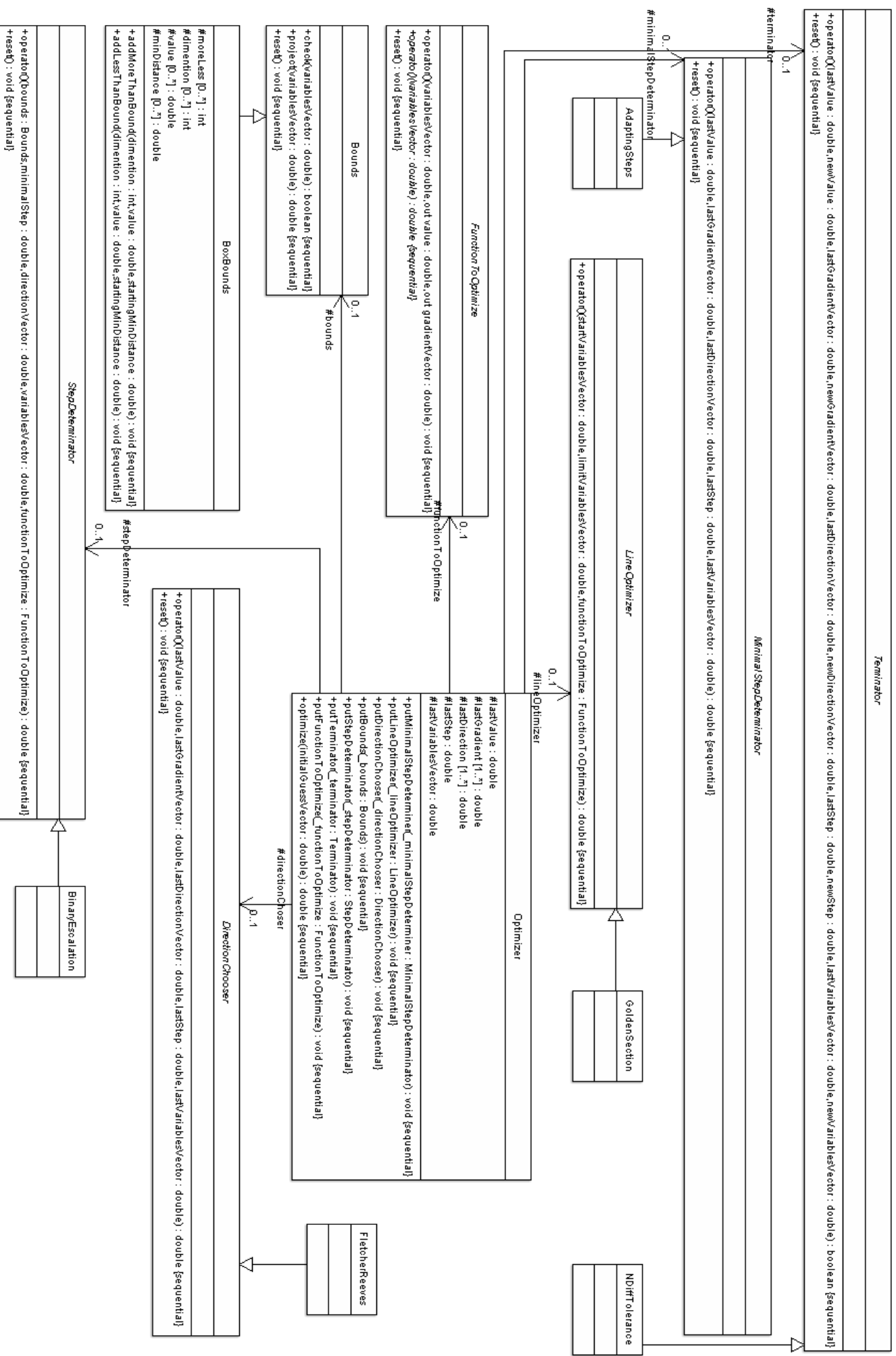
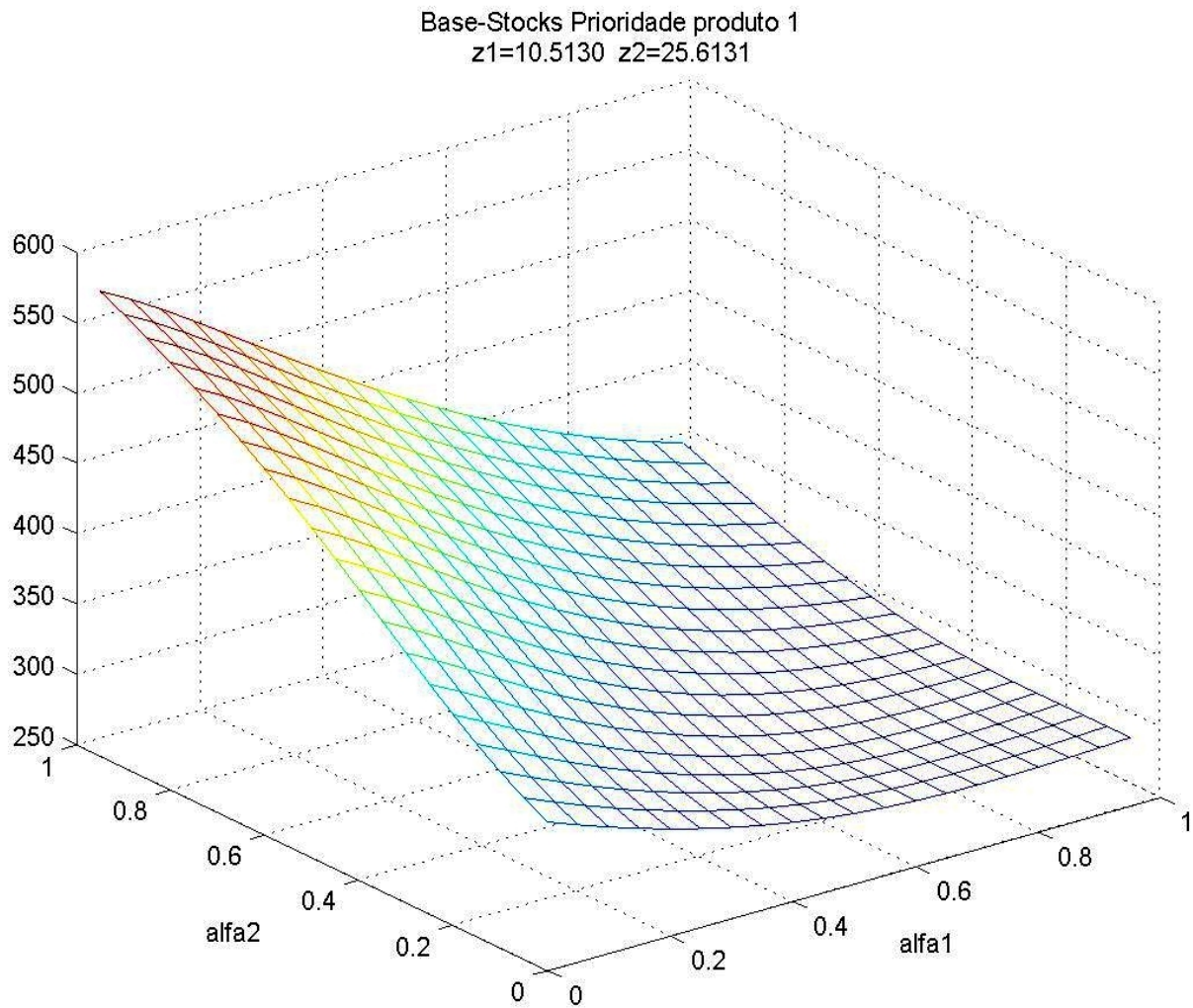


Diagrama de Classes de Optimization

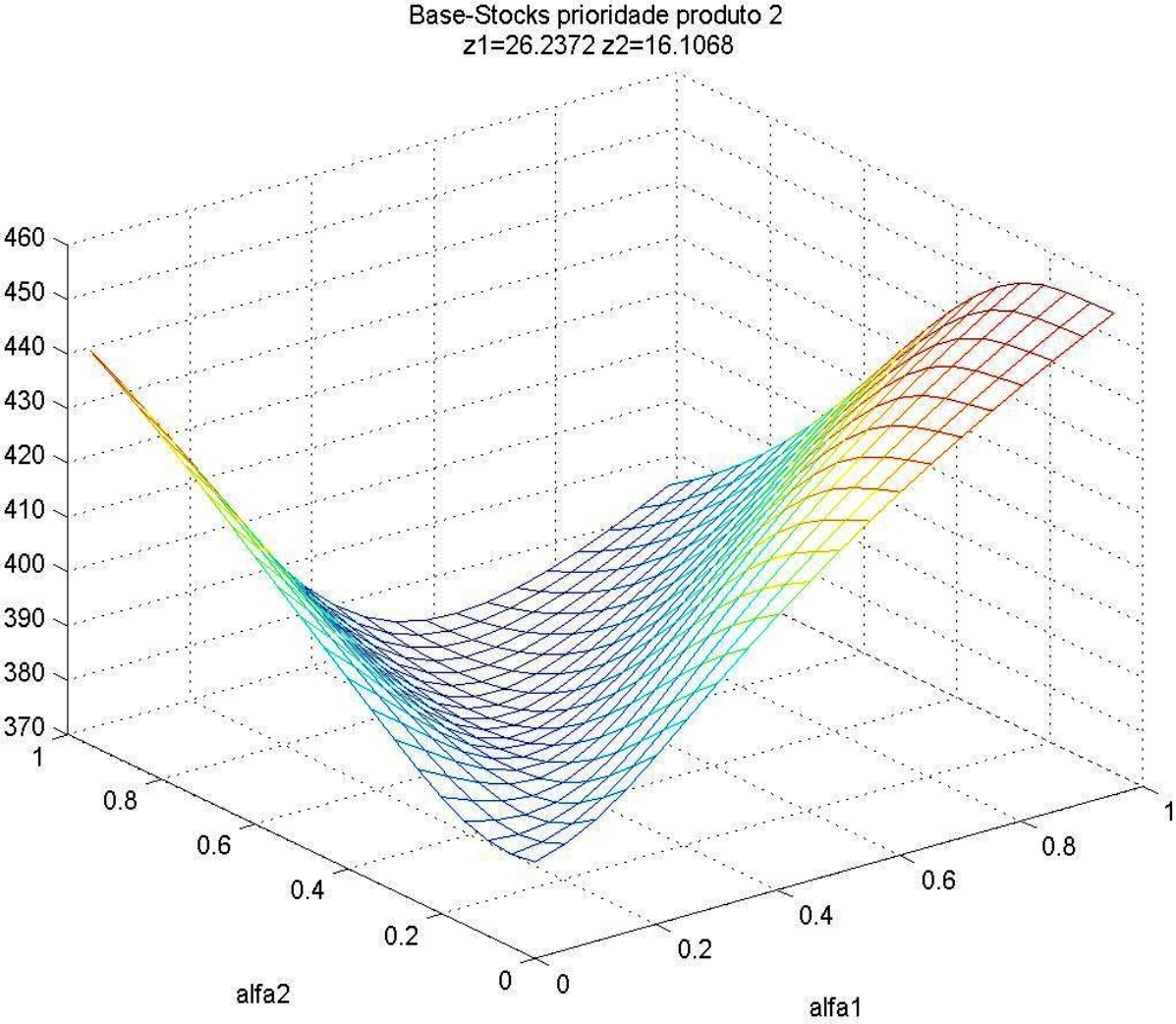


B. Gráficos da Função de Custo

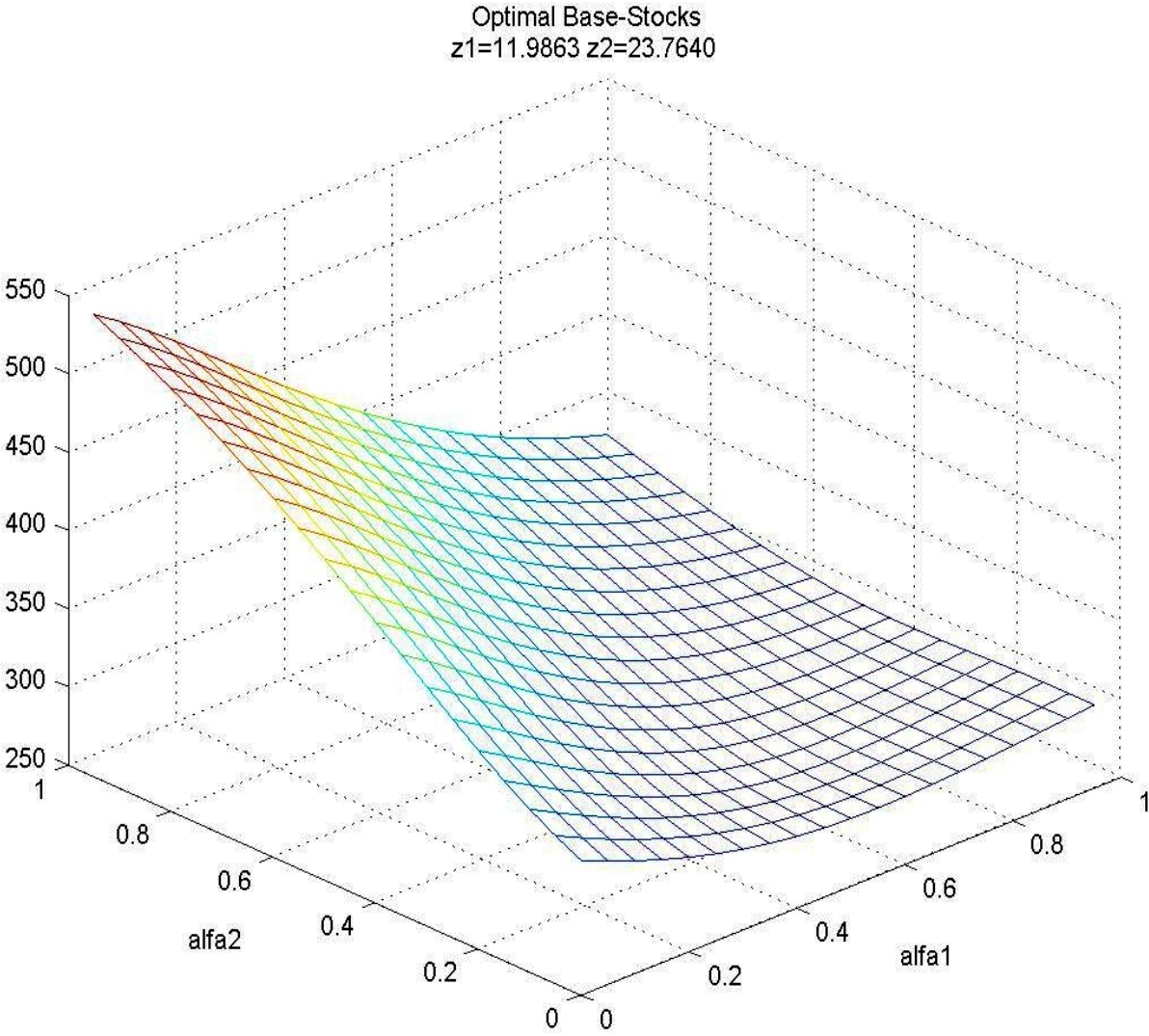
B.1. Base-Stocks Iguais aos Óptimos para Prioridade do Produto 1



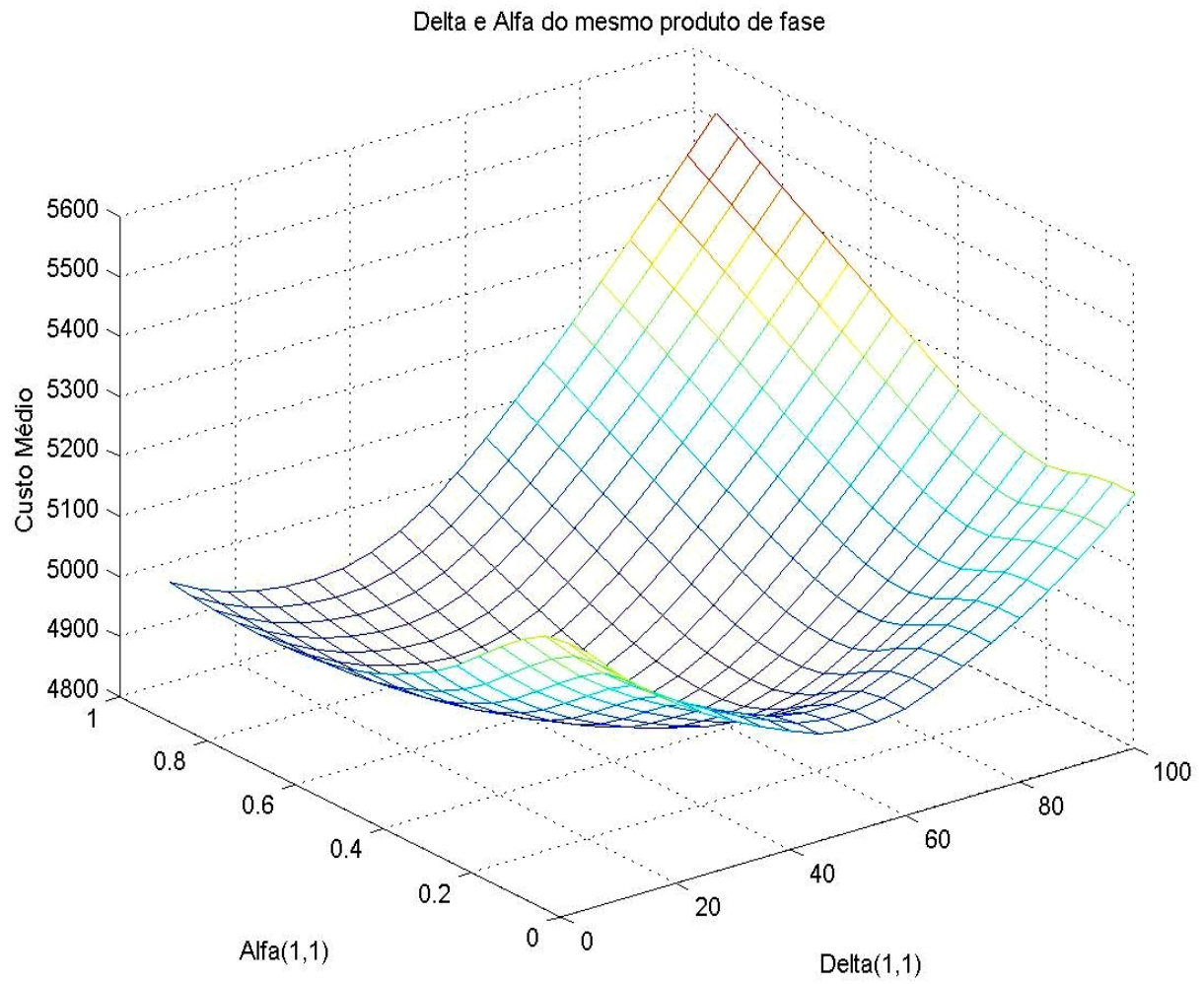
B.2. Base-Stocks Iguais aos Óptimos para Prioridade do Produto 2



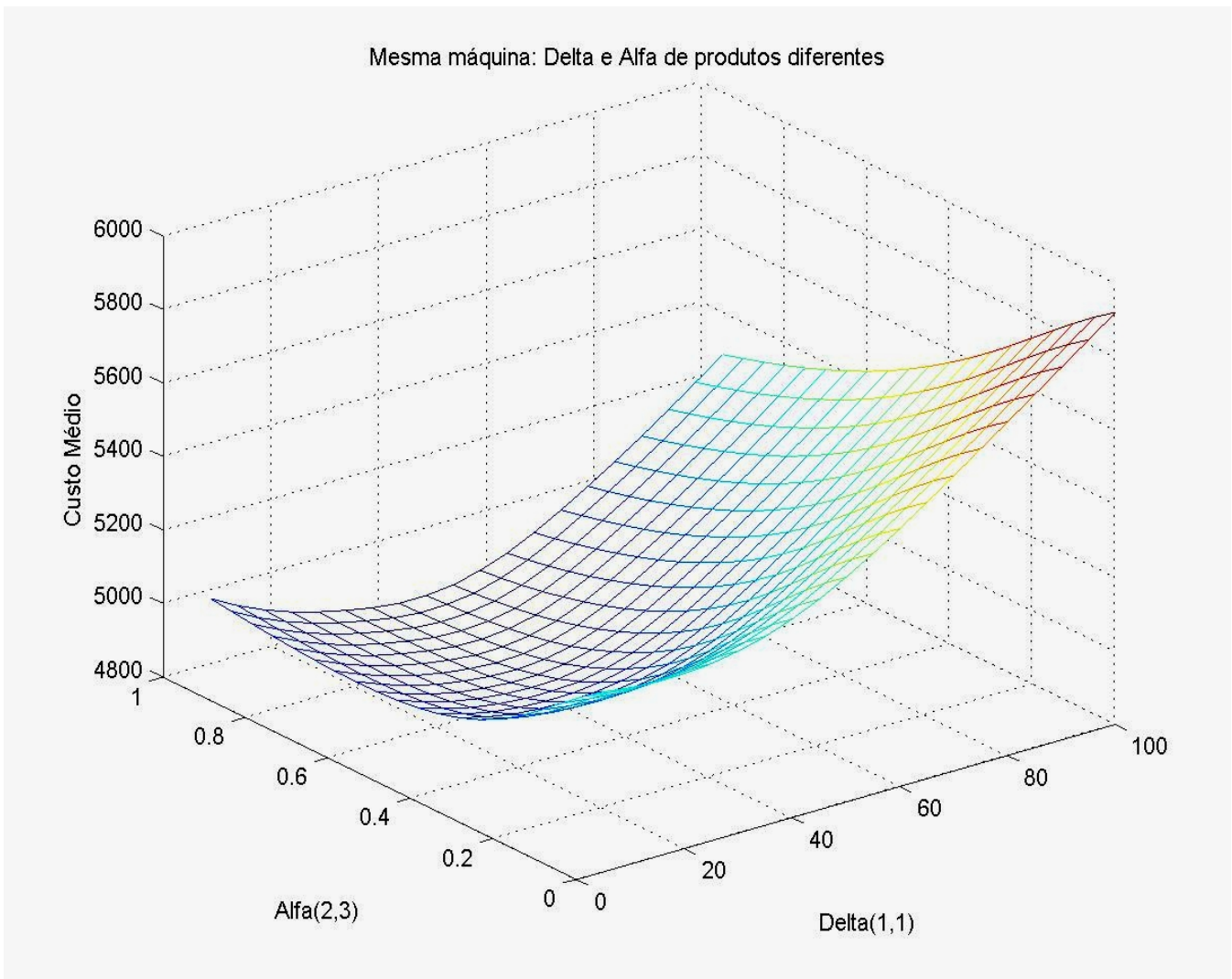
B.3. Base-Stocks Iguais aos Óptimos para Prioridades Suavizadas



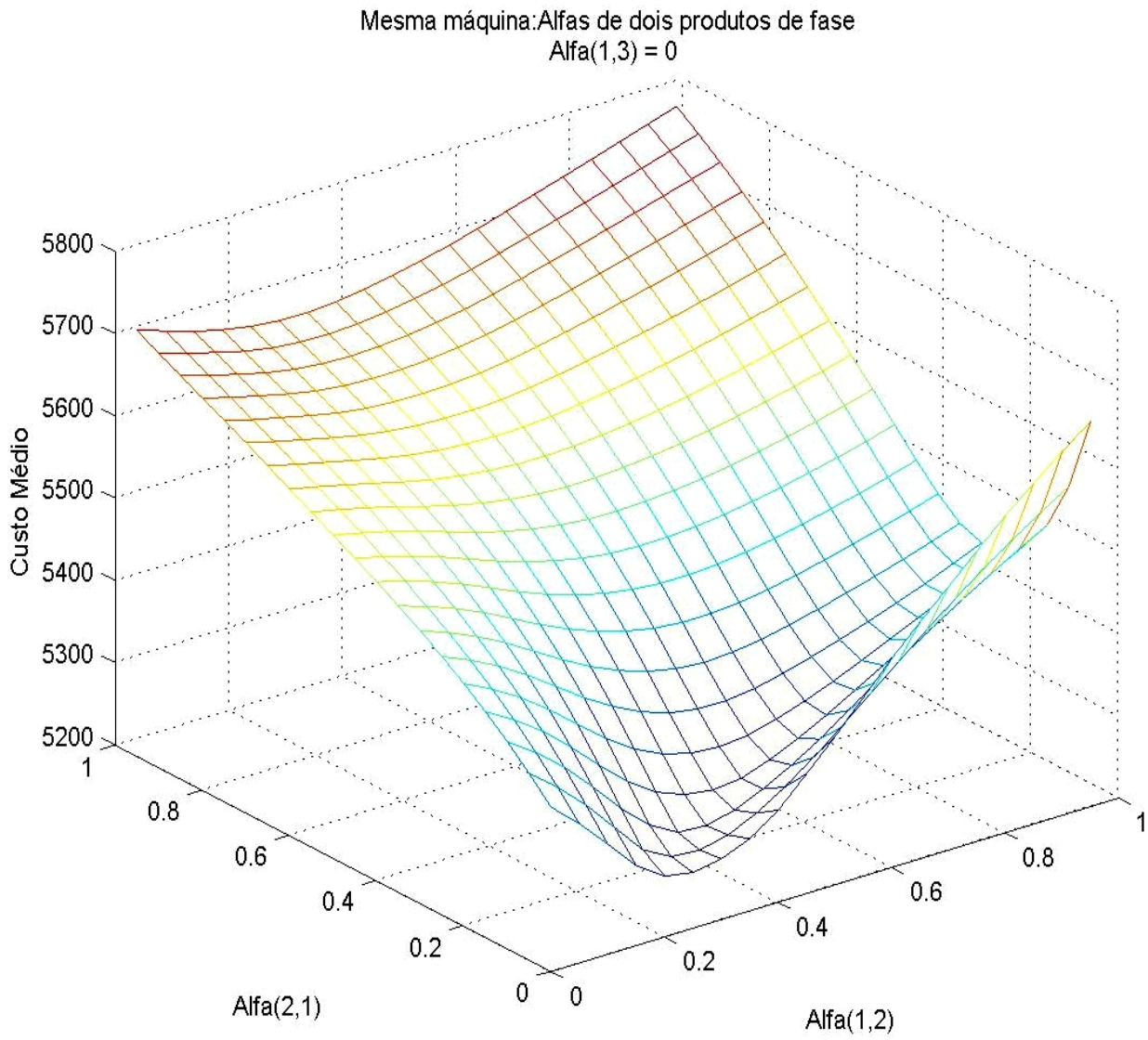
B.4. Mesmo Produto de Fase: Delta e Alfa



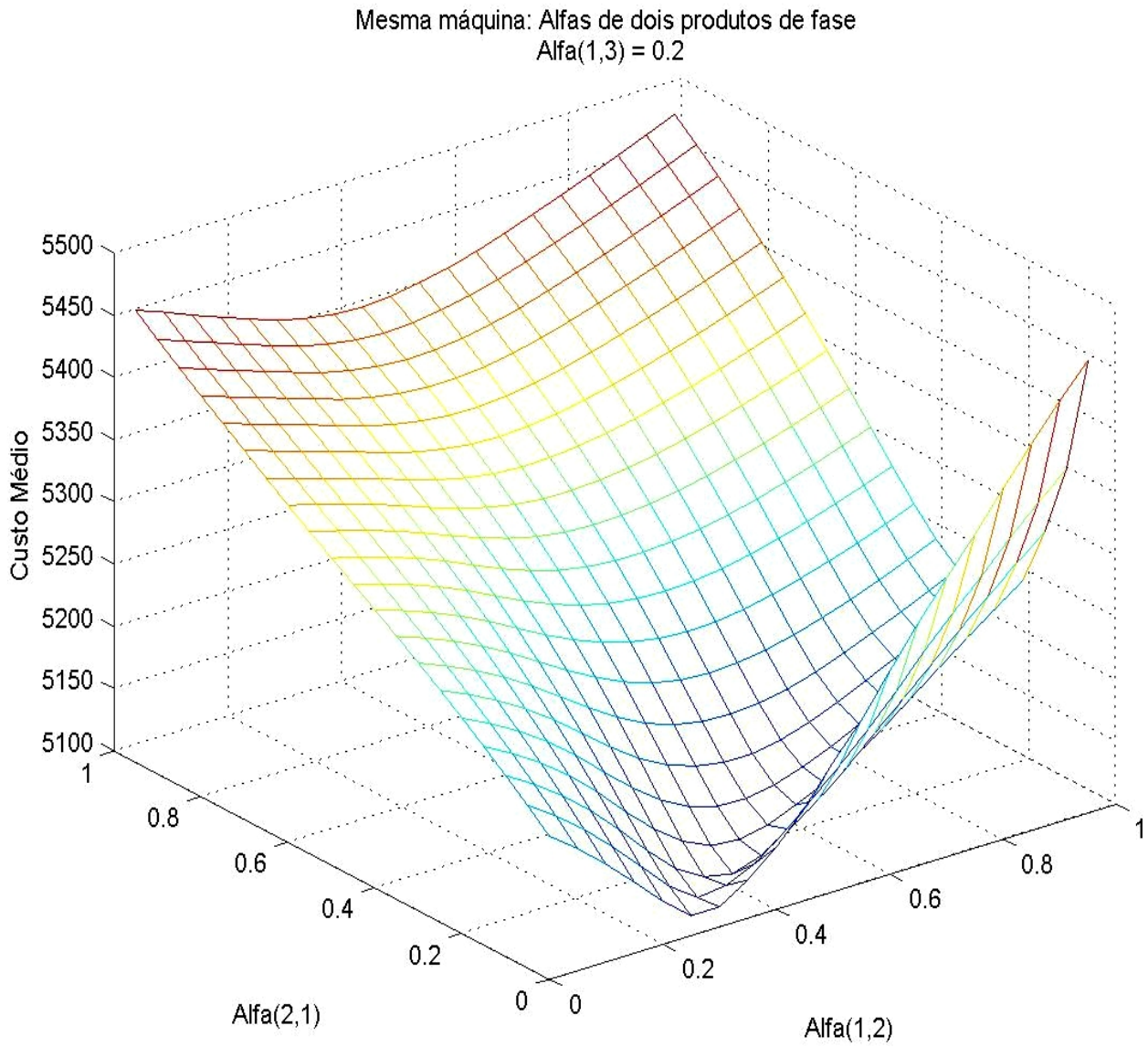
B.5. Mesma Máquina: Delta e Alfa de produtos de fase diferentes



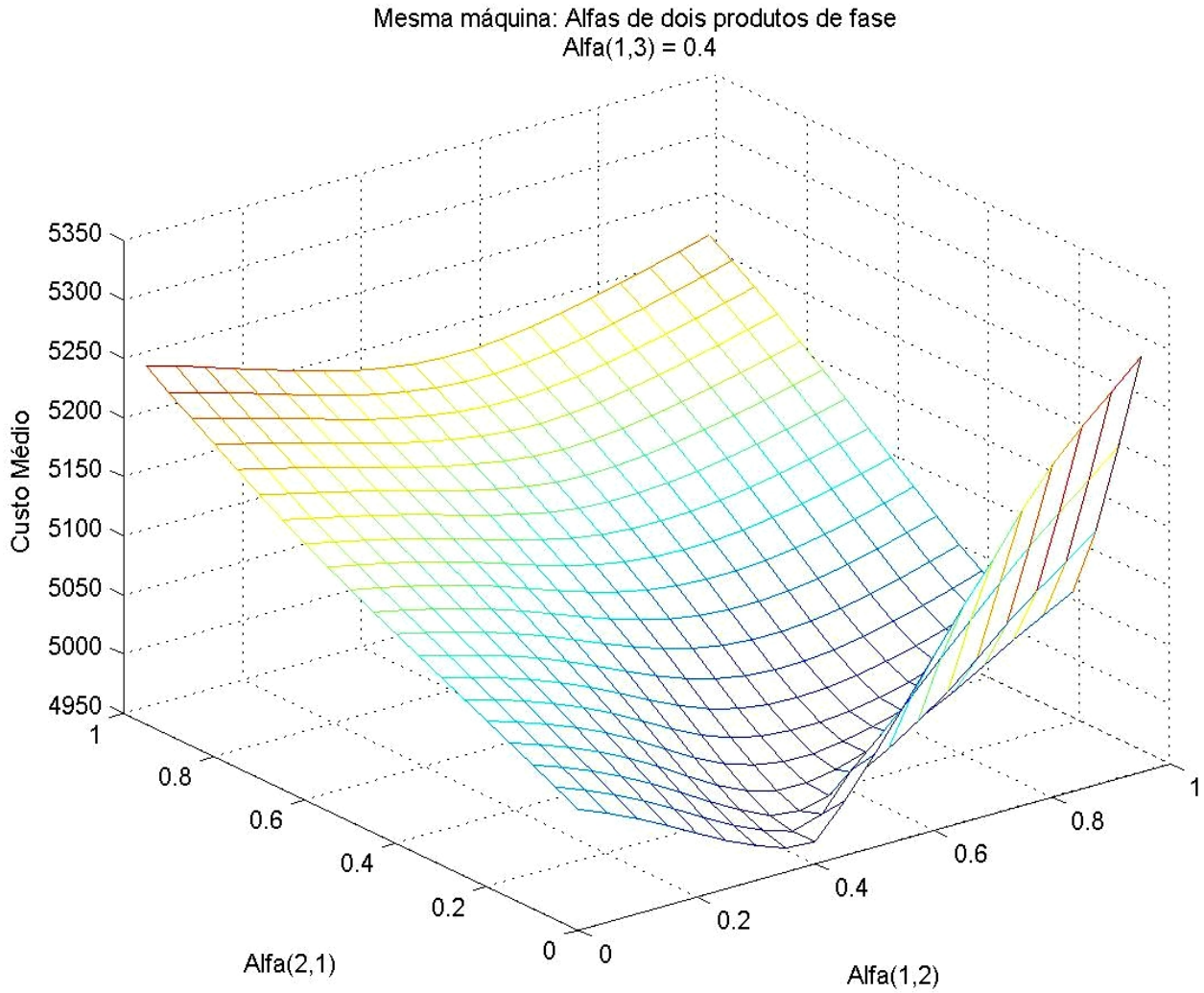
B.6. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0$)



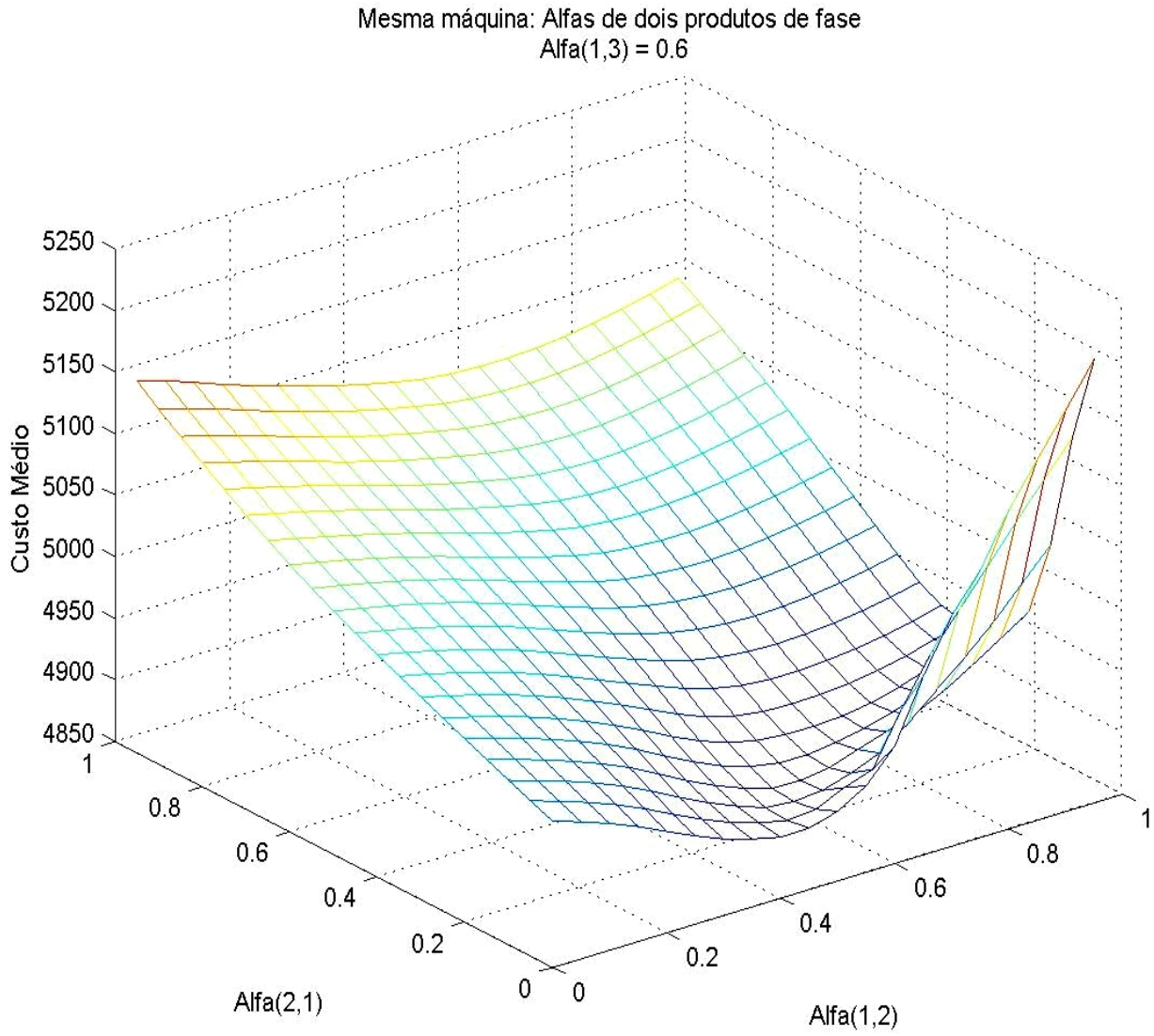
B.7. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0.2$)



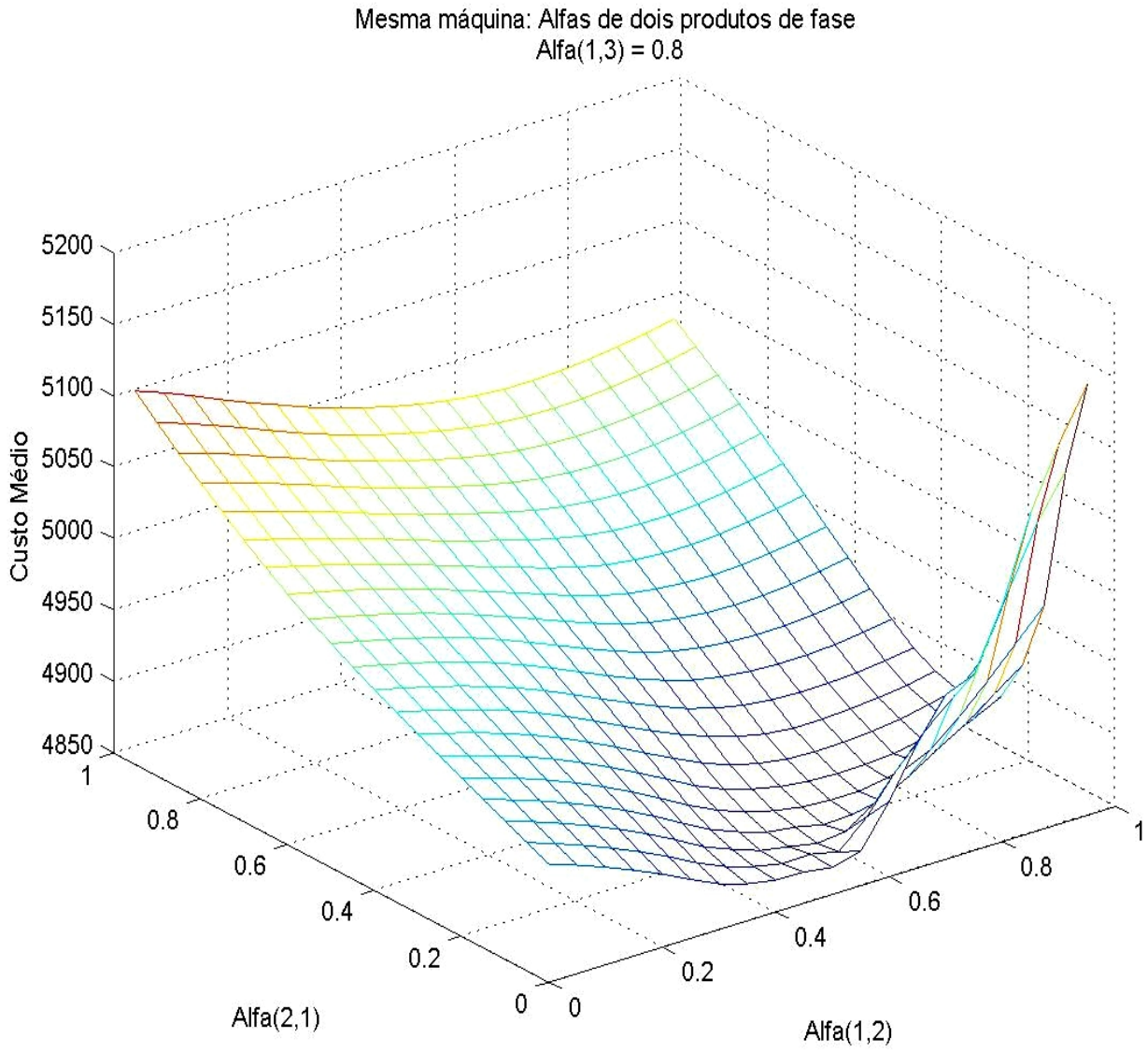
B.8. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0.4$)



B.9. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 0.6$)

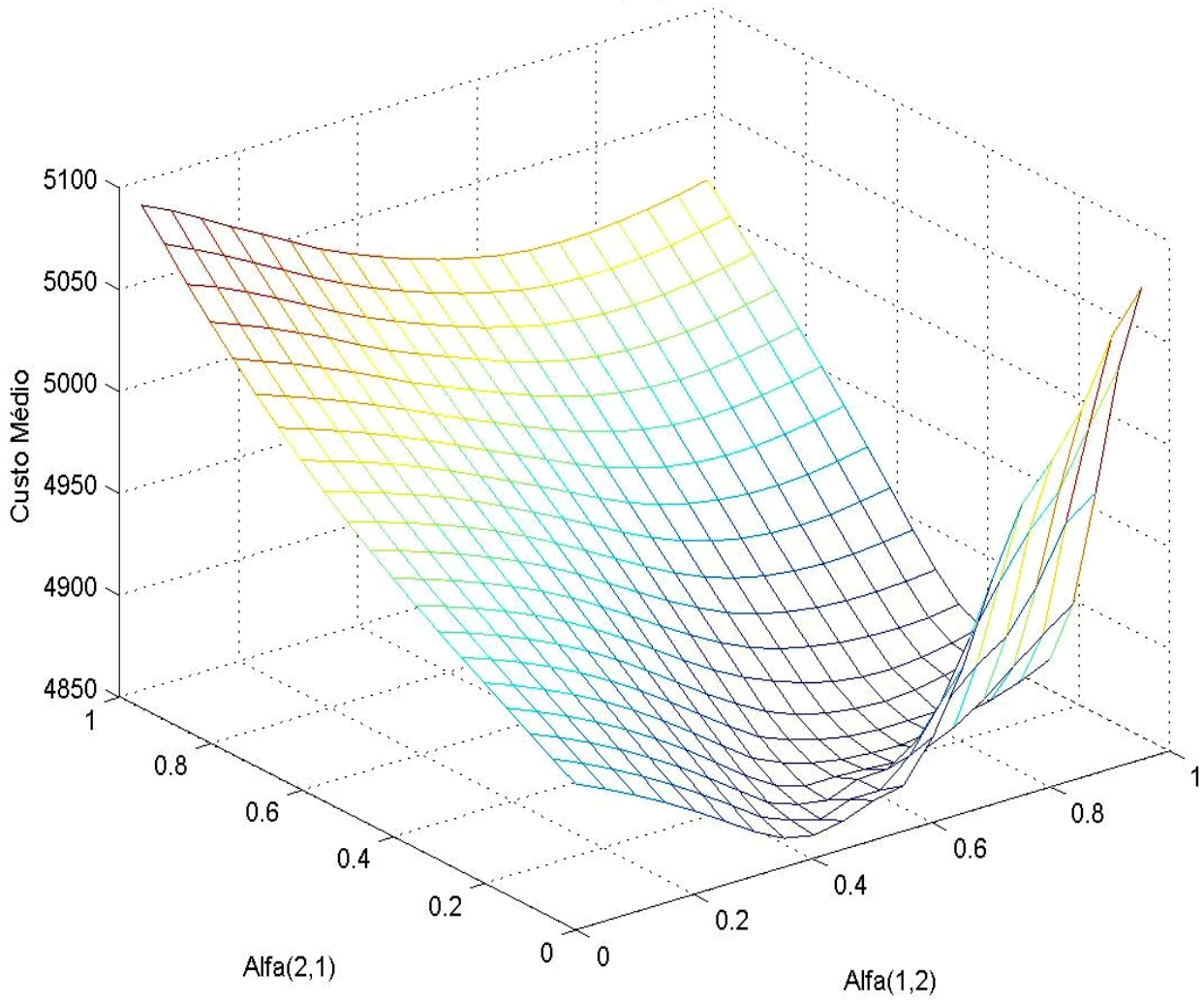


B.10. Mesma Máquina: vários Alfas (Alfa(1,3) = 0.8)

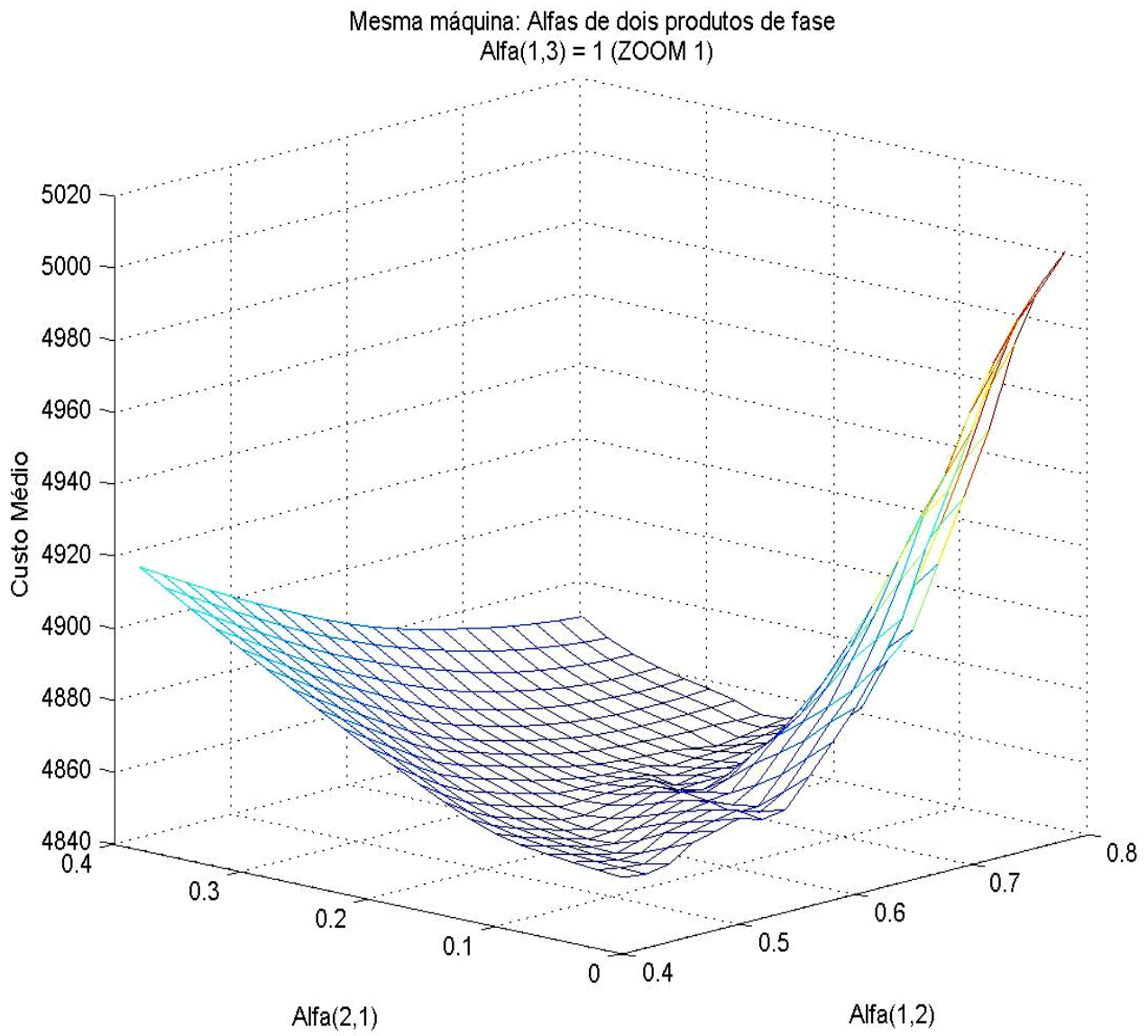


B.11. Mesma Máquina: vários Alfas (Alfa(1,3) = 1)

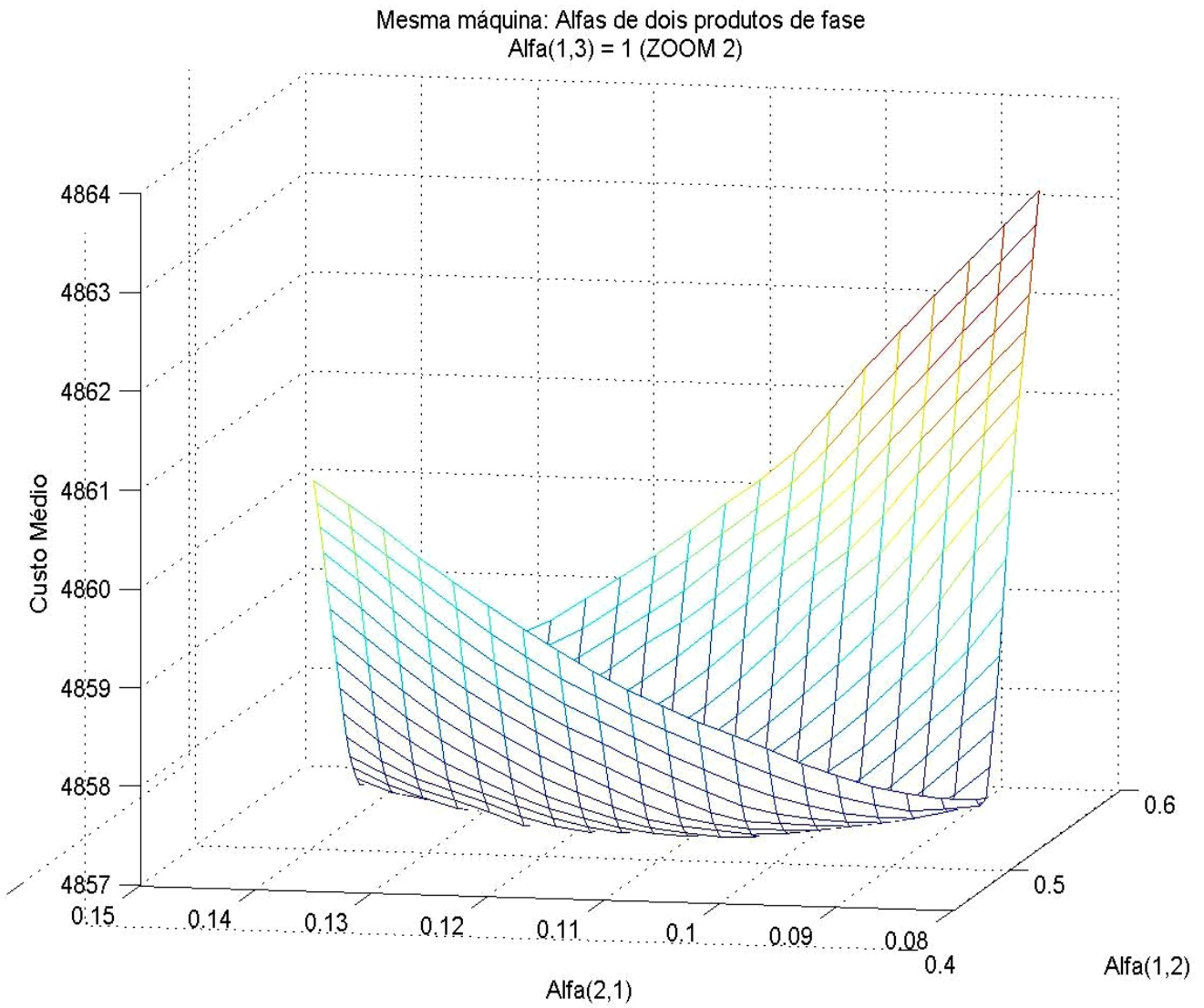
Mesma máquina: Alfas de dois produtos de fase
Alfa(1,3) = 1



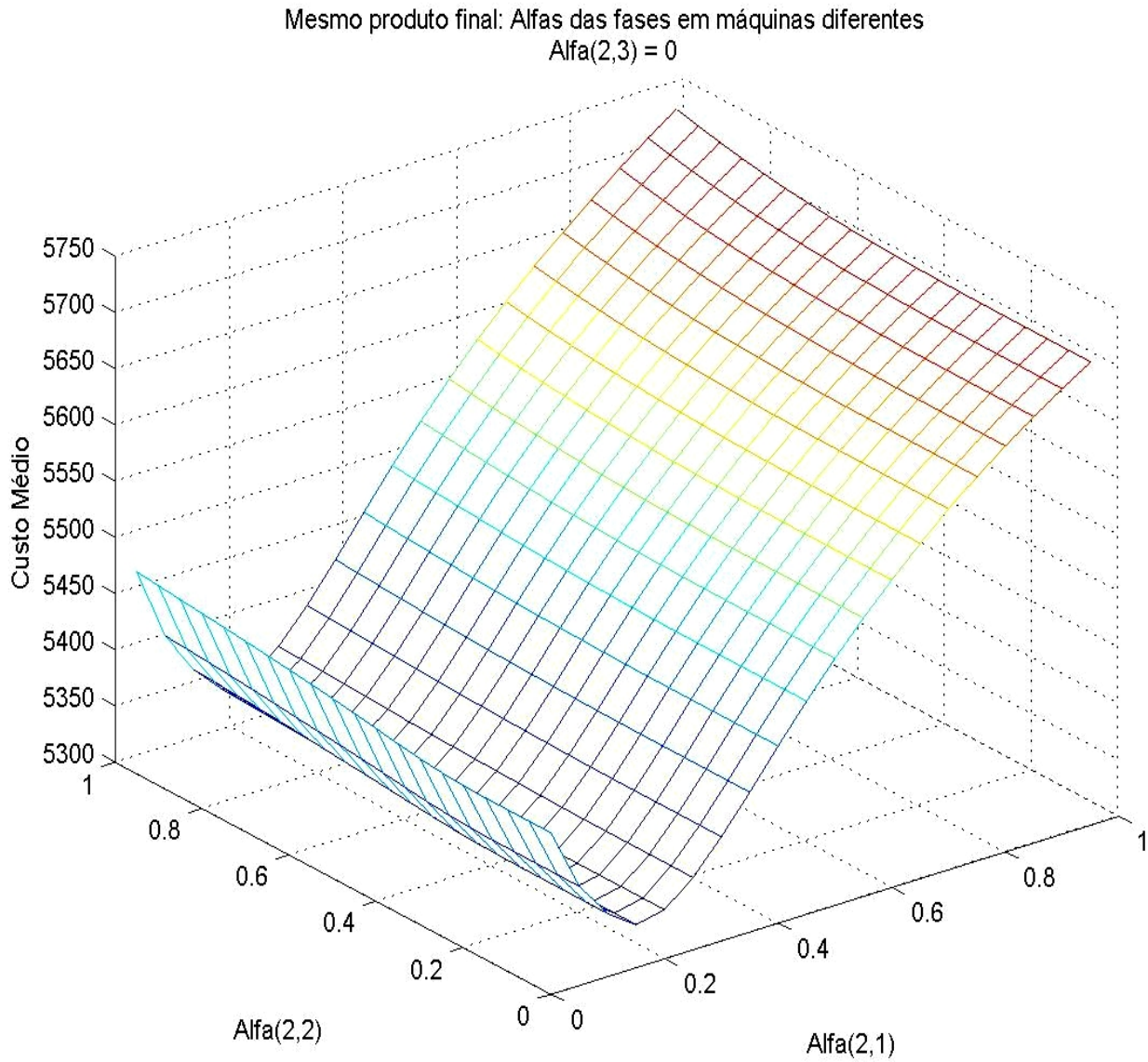
B.12. Mesma Máquina: vários Alfas (Alfa(1,3) = 1) ZOOM 1



B.13. Mesma Máquina: vários Alfas ($\text{Alfa}(1,3) = 1$) ZOOM 2

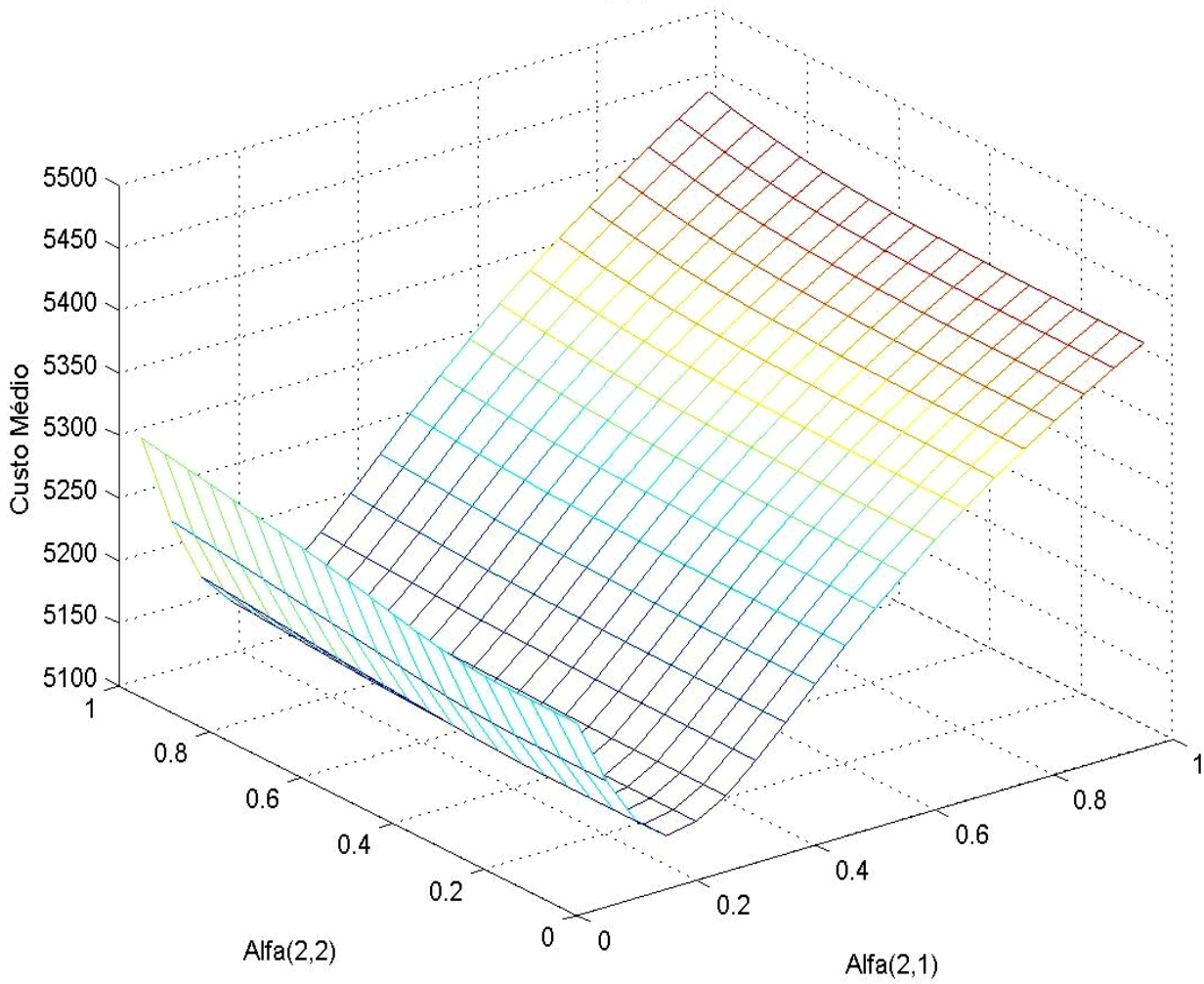


B.14. Mesmo Produto Final: Alfas em várias máquinas (Alfa(2,3) = 0)

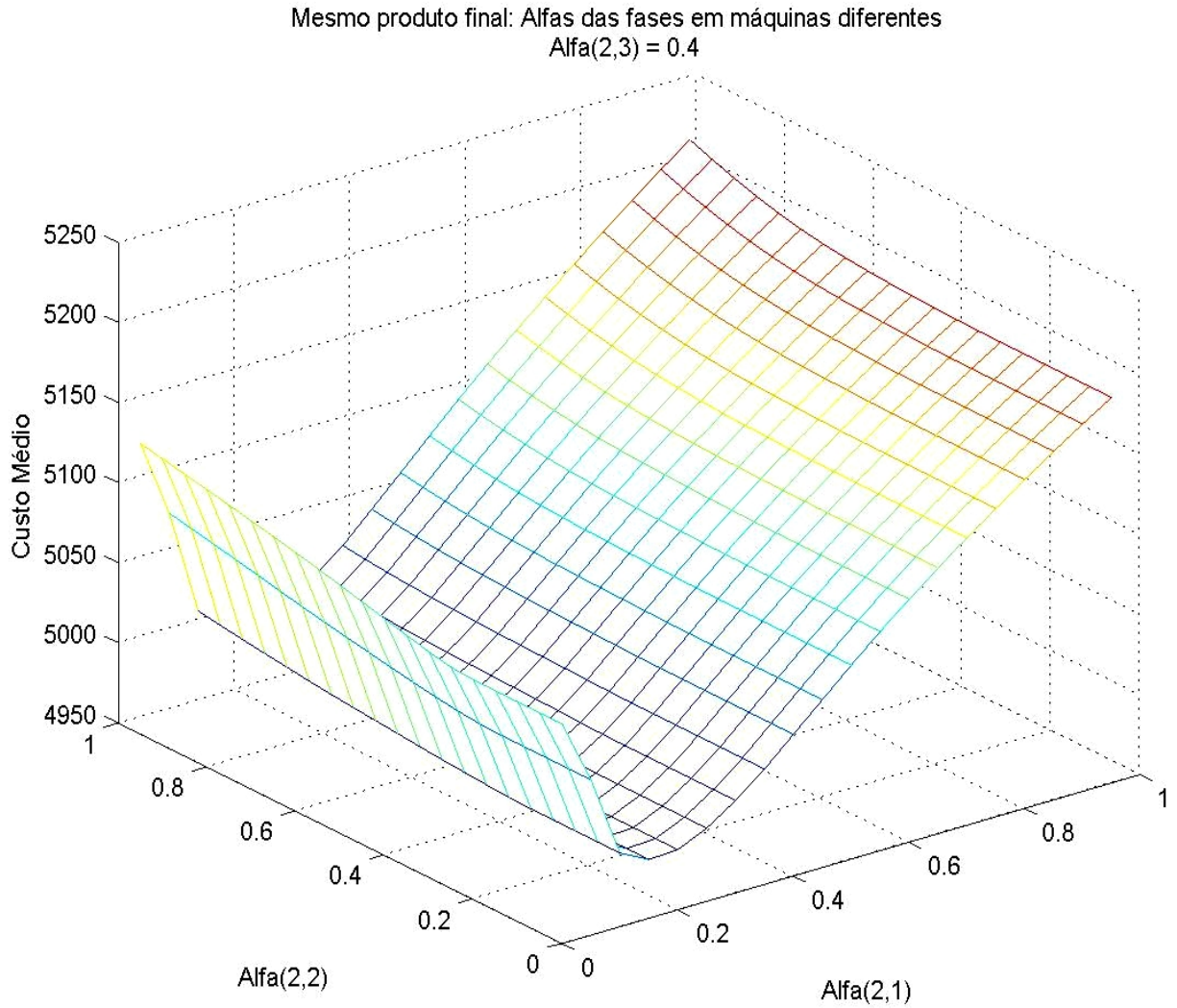


B.15. Mesmo Produto Final: Alfas em várias máquinas (Alfa(2,3) = 0.2)

Mesmo produto final: Alfas das fases em máquinas diferentes
Alfa(2,3) = 0.2

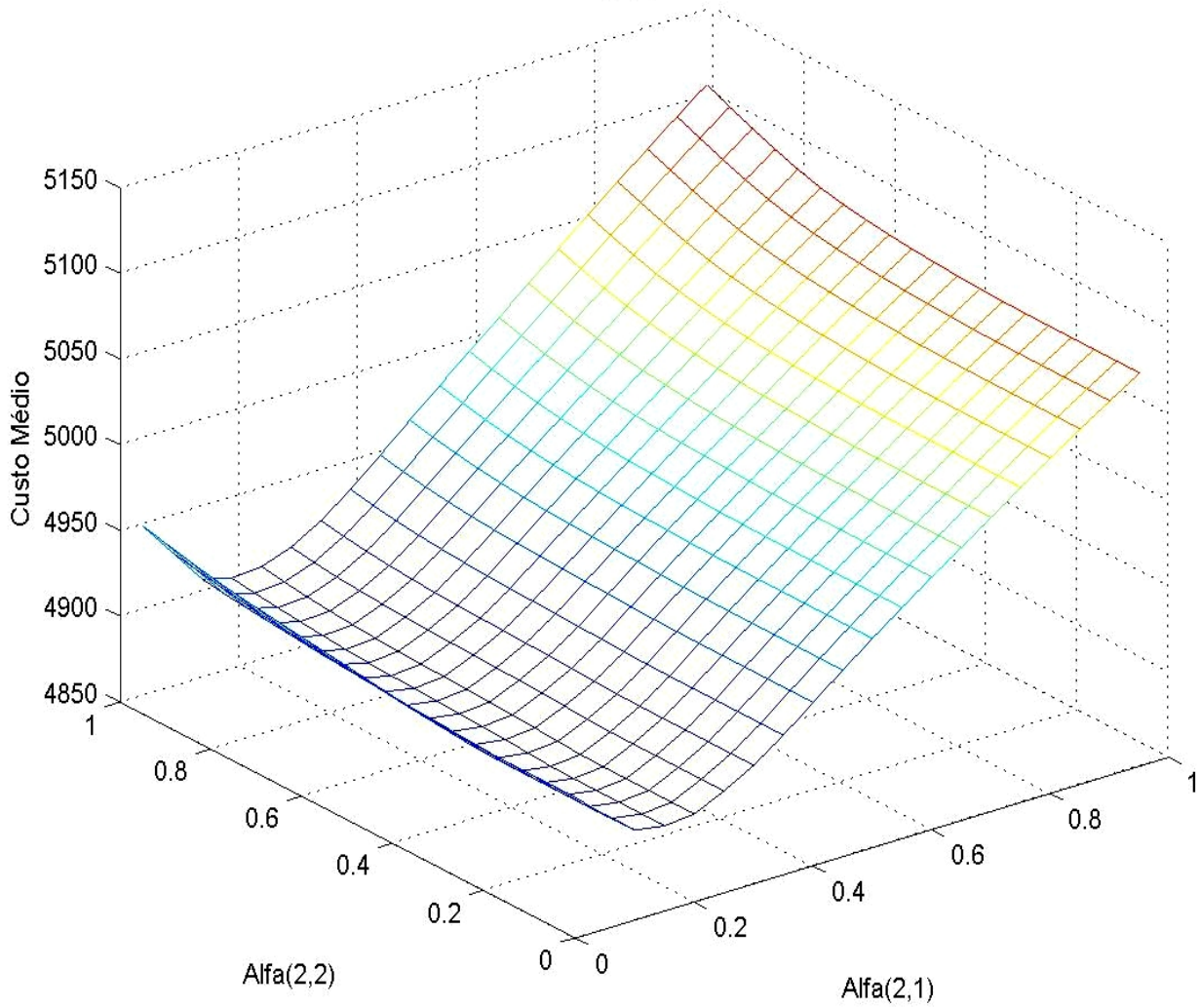


B.16. Mesmo Produto Final: Alfas em várias máquinas (Alfa(2,3) = 0.4)



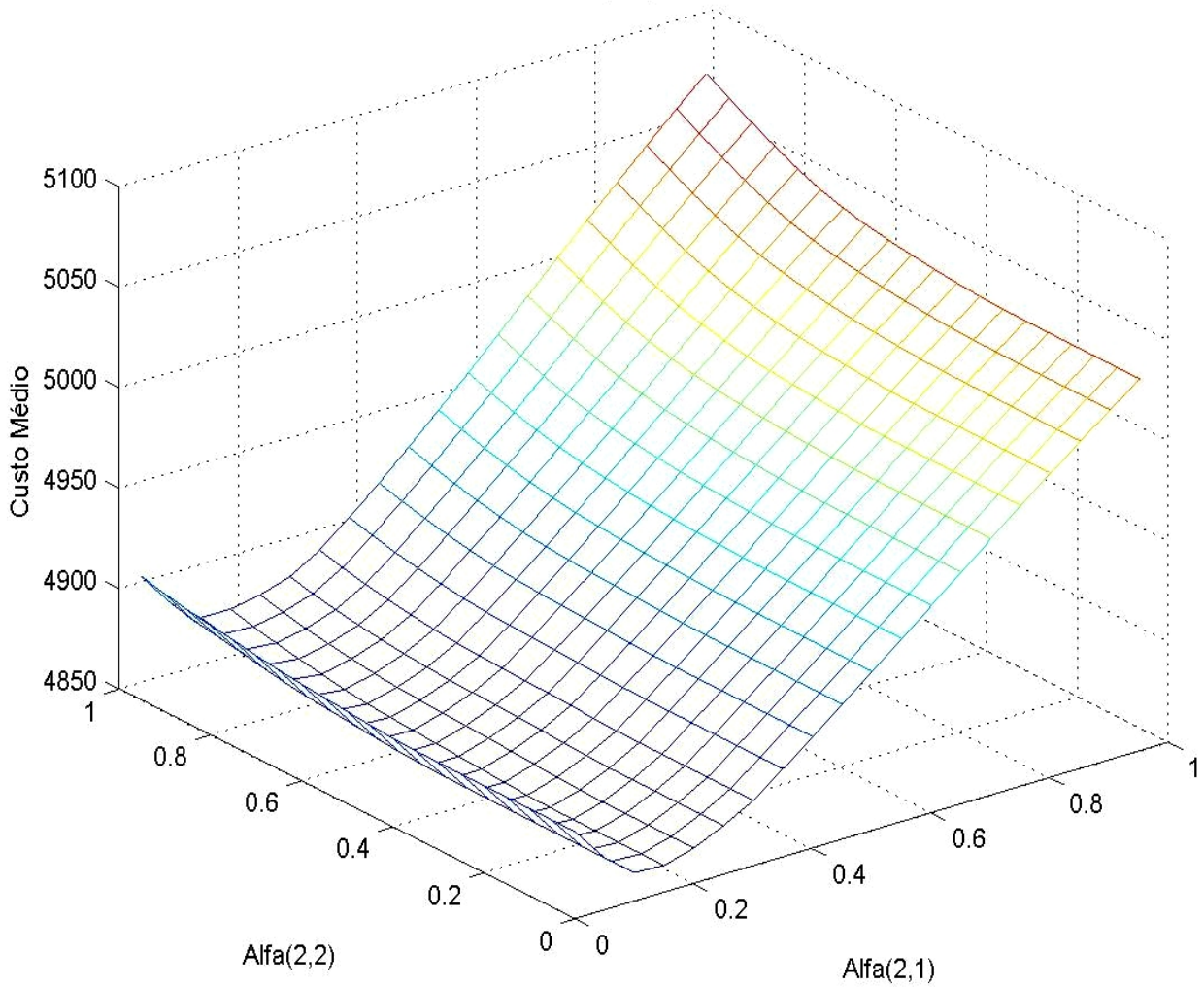
B.17. Mesmo Produto Final: Alfas em várias máquinas (Alfa(2,3) = 0.6)

Mesmo produto final: Alfas das fases em máquinas diferentes
Alfa(2,3) = 0.6



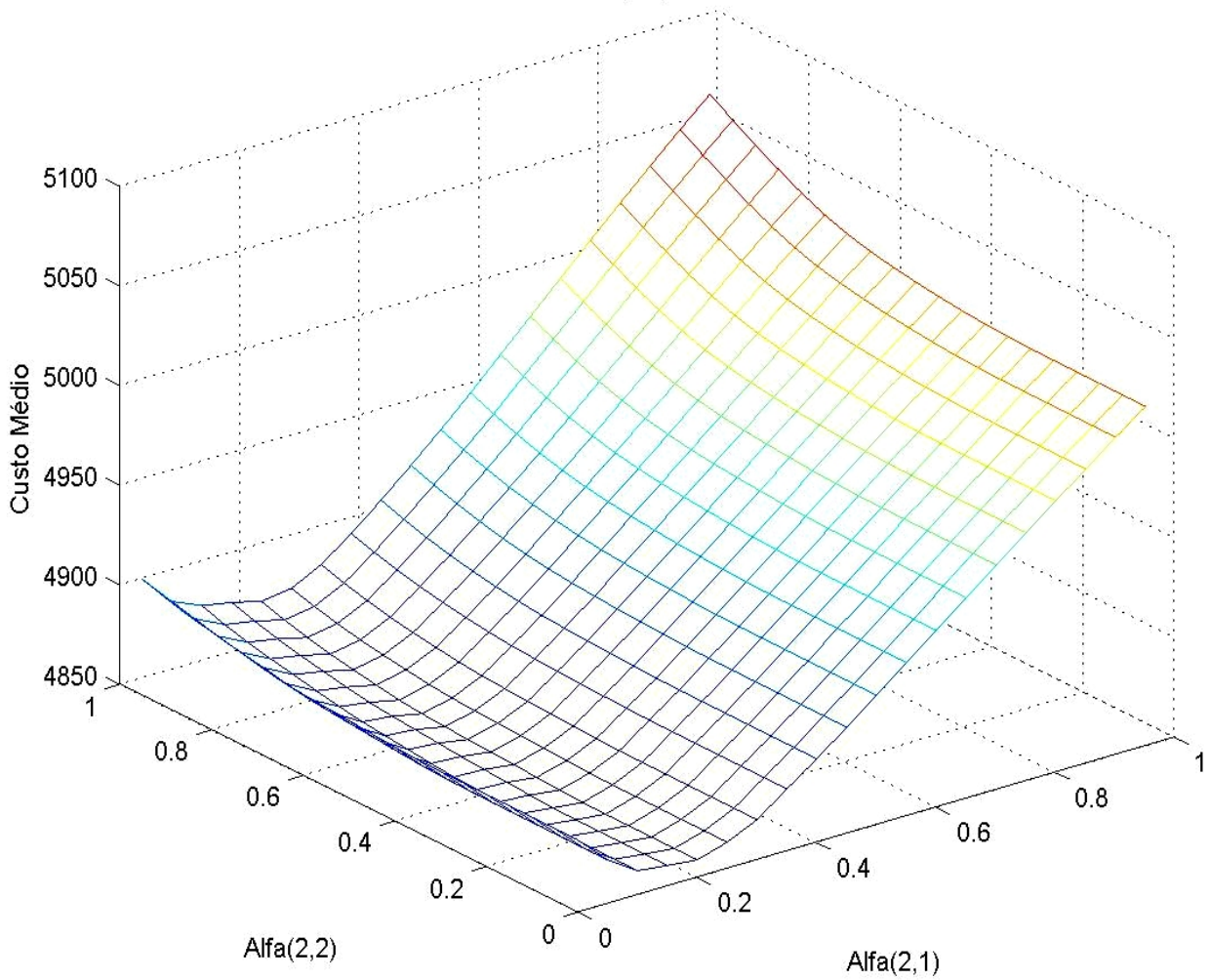
B.18. Mesmo Produto Final: Alfas em várias máquinas (Alfa(2,3) = 0.8)

Mesmo produto final: Alfas das fases em máquinas diferentes
Alfa(2,3) = 0.8



B.19. Mesmo Produto Final: Alfas em várias máquinas (Alfa(2,3) =1)

Mesmo produto final: Alfas das fases em máquinas diferentes
Alfa(2,3) = 1



C. Resultados de Optimização com Vários Custos

Parametros de Custo				Parametros PS				Parametros prio1		Parametros prio2		Custo Médio		
h1	b1	h2	b2	z1	z2	alfa1	alfa2	z1	z2	z1	z2	PS	prio1	prio2
1	1	1	1	51.592	53.995	0.612	0.564	30.360	58.795	60.865	30.474	93.071	98.632	98.210
1	1	1	6	60.871	88.051	0.000	1.000	30.354	164.989	60.868	88.051	144.022	185.944	144.022
1	1	1	11	56.305	117.487	0.000	0.618	30.367	208.258	60.877	110.141	164.079	223.955	167.443
1	1	1	16	56.204	132.637	0.000	0.612	30.358	233.804	60.894	125.987	179.202	247.207	182.840
1	1	6	1	58.393	9.034	0.000	0.785	30.346	12.991	60.859	6.765	97.684	108.313	98.386
1	1	6	6	60.868	30.479	0.000	1.000	30.361	58.816	60.871	30.476	241.884	377.374	241.884
1	1	6	11	60.875	45.366	0.000	1.000	30.345	89.927	60.820	45.366	335.452	546.753	335.452
1	1	6	16	60.882	57.023	0.000	1.000	30.347	112.037	60.798	57.023	405.149	669.253	405.149
1	1	11	1	59.715	5.070	0.000	0.891	30.336	7.458	60.859	3.902	99.414	110.999	99.786
1	1	11	6	60.878	19.407	0.000	1.000	30.361	37.023	60.826	19.406	269.576	430.895	269.576
1	1	11	11	60.868	30.479	0.000	1.000	30.360	58.781	60.876	30.475	395.558	666.115	395.558
1	1	11	16	60.883	39.493	0.000	1.000	30.376	77.686	60.798	39.492	496.180	849.989	496.180
1	1	16	1	60.880	2.866	0.000	1.000	30.329	5.260	60.803	2.867	100.384	112.088	100.384
1	1	16	6	60.879	13.957	0.000	1.000	30.340	27.002	60.805	13.956	283.229	456.937	283.229
1	1	16	11	60.874	23.066	0.000	1.000	30.364	44.398	60.922	23.066	428.239	729.741	428.239
1	1	16	16	60.874	30.478	0.000	1.000	30.359	58.817	60.870	30.479	549.232	954.856	549.232
1	6	1	1	97.498	53.483	0.484	0.000	85.518	58.784	166.529	30.479	140.880	145.067	184.529
1	6	1	6	124.986	120.129	0.084	0.142	85.518	164.989	166.494	88.051	218.787	242.379	240.341
1	6	1	11	129.077	144.541	0.327	0.441	85.518	208.258	166.506	110.141	245.358	280.390	263.762
1	6	1	16	143.407	143.827	0.000	0.382	85.514	233.804	166.493	125.987	262.525	303.642	279.159
1	6	6	1	85.518	12.991	1.000	0.000	85.518	12.974	166.494	6.765	164.748	164.748	194.705
1	6	6	6	158.070	34.960	0.000	0.719	85.518	58.817	166.529	30.474	334.939	433.809	338.203
1	6	6	11	166.529	45.366	0.000	1.000	85.504	89.927	166.592	45.366	431.771	603.189	431.771
1	6	6	16	166.529	57.023	0.000	1.000	85.505	112.037	166.557	57.023	501.468	725.688	501.468
1	6	11	1	85.518	7.460	1.000	0.000	85.500	7.470	166.510	3.902	167.434	167.434	196.105
1	6	11	6	162.801	21.310	0.000	0.855	85.505	37.025	166.487	19.370	364.076	487.331	365.895
1	6	11	11	166.529	30.480	0.000	1.000	85.518	58.784	166.529	30.479	491.877	722.550	491.877
1	6	11	16	166.529	39.493	0.000	1.000	85.495	77.681	166.488	39.492	592.499	906.424	592.499
1	6	16	1	85.518	5.260	1.000	0.000	85.522	5.260	166.497	2.867	168.523	168.523	196.703
1	6	16	6	166.529	13.957	0.000	1.000	85.513	26.999	166.490	13.957	379.548	513.372	379.548
1	6	16	11	166.529	23.066	0.000	1.000	85.514	44.399	166.487	23.066	524.558	786.176	524.558
1	6	16	16	166.529	30.480	0.000	1.000	85.518	58.814	166.529	30.479	645.551	1011.290	645.551
1	11	1	1	116.848	55.167	0.618	0.000	110.583	58.828	210.156	30.474	166.105	169.877	221.371
1	11	1	6	144.064	126.930	0.084	0.000	110.583	164.998	210.156	88.049	245.232	267.189	277.184
1	11	1	11	152.594	151.099	0.456	0.492	110.583	208.258	210.156	110.159	272.752	305.200	300.605
1	11	1	16	154.692	165.833	0.269	0.327	110.583	233.817	210.242	125.987	289.191	328.452	316.002
1	11	6	1	116.848	12.631	0.618	0.000	110.583	12.995	210.156	6.763	184.611	189.558	231.548
1	11	6	6	195.927	36.057	0.000	0.665	110.583	58.817	210.156	30.479	370.700	458.618	375.046
1	11	6	11	203.863	48.270	0.000	0.812	110.567	89.927	210.249	45.366	465.079	627.998	468.614
1	11	6	16	210.156	57.023	0.000	1.000	110.585	112.037	210.181	57.023	538.311	750.498	538.311
1	11	11	1	116.848	7.246	0.618	0.000	110.583	7.464	210.259	3.898	187.212	192.244	232.948
1	11	11	6	204.829	21.542	0.000	0.832	110.475	37.023	210.103	19.388	400.456	512.140	402.737
1	11	11	11	204.296	33.212	0.000	0.823	110.583	58.789	210.156	30.474	526.357	747.360	528.720
1	11	11	16	210.156	39.493	0.000	1.000	110.645	77.686	210.156	39.492	629.342	931.234	629.342
1	11	16	1	116.848	5.132	0.618	0.000	110.583	5.260	210.253	2.867	188.273	193.333	233.546
1	11	16	6	206.044	15.723	0.000	0.871	110.476	26.987	210.252	13.955	414.730	538.182	416.391
1	11	16	11	206.233	24.640	0.000	0.877	110.600	44.399	210.158	23.065	559.842	810.986	561.401
1	11	16	16	210.156	30.480	0.000	1.000	110.583	58.781	210.156	30.478	682.394	1036.100	682.394
1	16	1	1	133.534	55.167	0.618	0.000	127.413	58.788	233.687	30.474	181.624	185.691	243.393
1	16	1	6	157.256	129.914	0.238	0.113	127.413	165.011	233.687	88.051	261.520	283.002	299.206
1	16	1	11	168.003	153.306	0.385	0.381	127.602	208.258	233.723	110.141	289.806	321.013	322.626
1	16	1	16	173.131	167.004	0.324	0.366	127.439	233.804	233.859	125.987	306.379	344.266	338.023
1	16	6	1	133.535	12.633	0.618	0.000	127.413	12.972	233.687	6.764	200.129	205.371	253.570
1	16	6	6	219.357	36.604	0.000	0.644	127.413	58.812	233.687	30.475	392.340	474.432	397.067
1	16	6	11	220.884	51.401	0.000	0.671	127.477	89.927	233.687	45.366	485.634	643.812	490.635
1	16	6	16	220.450	63.595	0.000	0.660	127.413	112.048	233.687	57.025	555.036	766.311	560.333
1	16	11	1	133.535	7.246	0.618	0.000	127.437	7.460	233.688	3.893	202.730	208.057	254.970
1	16	11	6	227.091	21.581	0.000	0.830	127.417	37.021	233.870	19.394	422.311	527.954	424.759
1	16	11	11	226.943	33.065	0.000	0.828	127.413	58.793	233.687	30.475	548.224	763.173	550.742
1	16	11	16	233.683	39.492	0.000	1.000	127.454	77.686	233.690	39.490	651.363	947.047	651.364
1	16	16	1	133.535	5.132	0.618	0.000	127.413	5.260	233.898	2.865	203.791	209.146	255.568
1	16	16	6	228.782	15.801	0.000	0.869	127.413	26.990	233.897	13.955	436.629	553.995	438.413
1	16	16	11	228.906	24.774	0.000	0.870	127.475	44.399	233.687	23.065	581.742	826.799	583.423
1	16	16	16	233.685	30.480	0.000	1.000	127.413	58.784	233.687	30.475	704.415	1051.910	704.416