

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Otimização / Simulação em Sistemas Industriais

Pedro Rafael Dinis Sousa Ferreira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor Jorge Pinho de Sousa

Co-orientador: Professor Doutor Samuel Oliveira Moniz

24 de Julho de 2017

Ao meu Pai,

Resumo

Atualmente a eficiência operacional é vista como uma vantagem competitiva e uma área de pesquisa e investimento. O crescente aumento da necessidade de produção em altas quantidades aumentou a importância do planejamento de produção, tendo como consequência o desenvolvimento tecnológico permitindo o aparecimento de métodos híbridos de otimização e simulação, que comprovaram que após a interação entre os dois métodos melhores soluções eram originadas.

Nesta dissertação um conjunto de métodos relacionados com o planejamento a nível industrial são estudados e analisados. Na abordagem de *Byrne e Bakir* [1] é apresentado um maior foco ao nível do modelo matemático, visto que usa a mesma formulação matemática desenvolvida no modelo de otimização deste projeto.

Resultados da simulação servirão como entrada do modelo MIP ajustando alguns parâmetro e restrições com a intenção de produzir planos de produção mais satisfatórios e realistas. A metodologia desenvolvida nesta dissertação foi aplicada a um caso de estudo genérico, onde a produção é desencadeada pela quantidade de produtos exigida pelos clientes. São propostos dois modelos: Modelo Base e Modelo Base Modificado, onde estes se diferenciam no quanto restringem as possibilidades de soluções.

A interação entre os métodos de otimização e simulação é desenvolvida em linguagem de programação C # e tem como auxílio o uso da *Application Program Interface* (API), permitindo uma rápida e fácil alocação de dados tanto no sistema a simular como a otimizar.

Os resultados presentes no projeto apoiam a metodologia proposta como uma alternativa viável para os problemas de planejamento de produção.

Abstract

Nowadays operational efficiency is seen as a competitive and investment research area. The need of high production quantities in today's industry increased the importance of production planning and scheduling. The technological development allowed the emergence of hybrid methods of optimization and simulation to execute their interaction with much better results.

In this dissertation, a set of methods related to industrial planning are studied and analysed. *Byrne and Bakir* [1] approach presents an important focus in the mathematical model proposal, since the same mathematical formulation is used in this paper.

The simulation results will serve as input to the MIP model, adjusting some parameters and constraints with the intention to produce more satisfactory and realistic production and scheduling plans. The methodology developed was applied to a generic case study where the production is driven by the product quantities required by the costumers. The search for the best possible solution is made through two proposed models: Base Model and Modified Base Model, where their difference is in how much they restrict the solution possibilities. The interaction between optimization and simulation methods is developed in C # programming language and it has the use of *Application Program Interface* (API), allowing easy and fast data allocation in both systems, to be simulated and optimized.

Results prove the proposed methodology to be a viable alternative to production planning problems.

Agradecimentos

Ficam aqui registados os agradecimentos a todos aqueles que, direta e indiretamente, contribuíram para o sucesso deste projeto.

Em primeiro lugar queria deixar um agradecimento especial pelo apoio e dedicação do meu orientador Professor Doutor Samuel Moniz, que sem a sua orientação e disponibilidade a conclusão desta dissertação seria impossível.

Agradeço a todos os meus colegas de faculdade que me acompanharam neste longo percurso académico, e à minha família pelo seu apoio incondicional.

Um especial apreço à equipa de trabalho do INESC TEC pela ajuda prestada no desenvolvimento desta dissertação.

Pedro Rafael Ferreira

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Contexto e Enquadramento	2
1.3	Objetivos	3
1.4	Abordagem Metodológica	3
1.5	Estrutura da Dissertação	4
2	Revisão de Literatura	5
2.1	Abordagens Híbridas de Otimização e Simulação	6
2.2	Meta-Heurísticas	9
2.3	Clearing Functions	11
2.4	Simulação	12
2.5	Regras de Despacho – Sequenciamento de produção	14
2.6	Análise Crítica da Literatura	14
3	Metodologia baseada em métodos de Otimização-Simulação	17
3.1	Descrição do Problema	17
3.1.1	Caso de Estudo	18
3.2	Metodologia Simulação Otimização	19
3.2.1	Otimização - Modelo Matemático	19
3.2.2	Modelo de Simulação Concetual	22
3.2.3	Abordagem Simulação - Otimização	23
4	Desenvolvimento do Modelo	29
4.1	Dados da Instância Modelo	29
4.2	Implementação do Modelo Matemático	30
4.2.1	Optimizador – Estrutura de Dados	30
4.3	Simulação	32
4.3.1	Entidades	32
4.3.2	Construção do Modelo	33
4.4	SIMIO API	47
4.5	Modelos de Interação entre Métodos	49
4.5.1	Modelo Base	49
4.5.2	Modelo Base Modificado	53
4.6	Regras de Sequenciamento	61

5	Discussão de Resultados	65
5.1	Impacto das diferentes Regras de Sequenciamento	65
5.2	Comparar Abordagens	67
5.3	Impacto das diferentes durações dos Períodos	68
6	Conclusão	71
6.1	Trabalhos Futuros	72
A		73
A.1	Impacto das diferentes Regras de Sequenciamento	73
A.1.1	Gráfico de Desempenho	73
B		75
B.1	Comparação dos Modelos Abordados	75
B.1.1	<i>Byrne e Bakir</i> [1]	75
B.1.2	<i>Kim e Kim</i> [11]	76
B.1.3	Modelo Base	76
B.1.4	Modelo Base	77
C		79
C.1	Resultados das várias janelas temporais	79
C.1.1	Modelo Base – 3 Períodos	79
C.1.2	Modelo Base Modificado – 3 Períodos	80
C.1.3	Modelo Base – 6 Períodos	81
C.1.4	Modelo Base Modificado – 6 Períodos	82
C.1.5	Modelo Base – 15 Períodos	83
C.1.6	Modelo Base Modificado– 15 Períodos	84
	Referências	85

Lista de Figuras

2.1	Procedimento normal de Otimização e Simulação Híbrida	7
3.1	Esquema do Sistema Genérico	19
3.2	Metodologia desenvolvida no método de Simulação	23
3.3	Esquema Abordagem Simulação - Otimização	24
3.4	Esquema Abordagem ROSA	26
4.1	Entidades específicas para cada produto	33
4.2	Objeto <i>Source</i>	34
4.3	Propriedades <i>Source</i>	34
4.4	Objeto <i>Server</i>	35
4.5	Propriedades <i>Server1</i>	35
4.6	Propriedades <i>Server2 e 4</i>	36
4.7	Modelo final com todos os objetos	36
4.8	Representação SIMIO tabela <i>demand</i>	37
4.9	Representação SIMIO tabela <i>operationTimes</i>	38
4.10	Representação SIMIO tabela <i>operationSequence</i>	39
4.11	Processo Roteamento de Entidades	40
4.12	Processo Critical Ratio + Shortest Processing Time	42
4.13	Processo Buffer Load Rule	43
4.14	Processo Entrada de Entidades	44
4.15	Processo Saída de Entidades	45
4.16	Executar um cenário com diferentes experiências – Modo de Experiência	46
4.17	<i>Flowchart</i> interação entre Optimizador e Simulador	48
4.18	Restrições definidas para o Modelo Base - Exemplo	49
4.19	Modelo Base – Passo 1	50
4.20	Modelo Base – Passo 2	50
4.21	Modelo Base – Passo 3	51
4.22	Modelo Base – Passo 4	51
4.23	Modelo Base – Passo 5	52
4.24	Modelo Base – Passo 7	52
4.25	Restrições definidas para o Modelo Base Modificado – Exemplo	53
4.26	Modelo Base Modificado – Passo 1	54
4.27	Modelo Base Modificado – Passo 2	55
4.28	Modelo Base Modificado – Passo 3	55
4.29	Modelo Base Modificado – Passo 4	56
4.30	Modelo Base Modificado – Passo 5	57
4.31	Modelo Base Modificado – Passo 6	58

4.32	Modelo Base Modificado – Passo 7	58
4.33	Modelo Base Modificado – Passo 8	59
4.34	Modelo Base Modificado – Passo 9	60
4.35	Modelo Base Modificado – Passo 10	61
4.36	Esquema representativo do sequenciamento FIFO/EDD	62
4.37	Esquema representativo do sequenciamento CR + SPT	63
4.38	Esquema representativo do sequenciamento BLR	63
4.39	Esquema representativo do sequenciamento Parallel Machines Cr + SPT + BLR	64
A.1	Desempenho das diferentes regras de sequenciamento para o primeiro ciclo de execução	73
C.1	Resultados obtidos pela abordagem Modelo Base – 3 Períodos	79
C.2	Resultados obtidos pela abordagem Modelo Base Modificado – 3 Períodos	80
C.3	Resultados obtidos pela abordagem Modelo Base – 6 Períodos	81
C.4	Resultados obtidos pela abordagem Modelo Base Modificado– 6 Períodos	82
C.5	Resultados obtidos pela abordagem Modelo Base – 6 Períodos	83
C.6	Resultados obtidos pela abordagem Modelo Base Modificado– 6 Períodos	84

Lista de Tabelas

2.1	Palavras-chave pesquisa	5
2.2	Divisão dos vários modelos de Simulação Otimização segundo <i>Figueira e Lobo</i> [2]	6
2.3	Comparação de vários Artigos	9
2.4	Aplicação de Projetos de Simulação	12
3.1	Elementos de Formulação – <i>Sets</i>	20
3.2	Elementos de Formulação – Índices	20
3.3	Elementos de Formulação – Parâmetros	20
3.4	Elementos de Formulação – Variáveis de Decisão	21
3.5	Elementos de Formulação Exemplo dos Modelos – Índices e Parâmetros	25
4.1	Pedido de produção de cada produto	29
4.2	Tempo de Processamento de cada produto na respetiva <i>Workcenter</i>	29
4.3	Custos associados à produção, inventário e <i>backorder</i> de produtos	30
4.4	Relação dos índices com os dados do Modelo	31
4.5	Replicação entre janelas temporais – Períodos	31
4.6	Propriedades das Entidades	33
4.7	Elementos de Monitorização	43
4.8	Representação valores R_t^{j+1} - Passo 2	54
4.9	Representação valores R_t^{j+1} - Passo 4	56
4.10	Representação valores R_t^{j+1} - Passo 6	57
4.11	Representação valores R_t^{j+1} - Passo 8	59
4.12	Representação valores R_t^{j+1} - Passo 10	60
5.1	Desempenho das diferentes regras de sequenciamento para o ciclo de execução ótimo	66
5.2	Relação entre as regras BLR e Parallel Machines CR + SPT + BLR	66
5.3	Comparação dos diferentes tipos de Abordagem	67
5.4	Desempenho das diferentes janelas temporais	69
5.5	Relação entre a quantidade de variáveis inteiras e a quantidade de períodos	70
B.1	Resultados obtidos pela abordagem de <i>Byrne e Bakir</i> [1]	75
B.2	Resultados obtidos pela abordagem de <i>Kim e Kim</i> [11]	76
B.3	Resultados obtidos pela abordagem Modelo Base	76
B.4	Resultados obtidos pela abordagem Modelo Base Modificado	77

Abreviaturas e Símbolos

AME	Analytical Model Enhancement
AG	Algoritmo Genérico
API	Application Program Interface
BLR	Buffer Load Rule
BOM	Bill of Materials
CF	Clear Functions
CR	Critical Ratio
DES	Discrete Event Simulation
EDD	Earliest Due Date
FIFO	First In First Out
GRASP	Greedy Randomized Adaptive Search Procedure
LP	Linear Programming
MILP	Mixed Integer Linear Programming
MTO	Make to Order
PLE	Personal Learning Edition
ROSA	Recursive Optimization-Simulation Approach
RSM	Responsive Surface Methodology
SE	Solution Evaluation
SG	Solution Generation
SPT	Shortest Processing Time
WIP	Work In Process

Capítulo 1

Introdução

1.1 Motivação

Hoje em dia para obter níveis elevados de rendimento e desempenho operacional dos sistemas de produção é necessário aumentar a confiabilidade e a qualidade dos processos de tomada de decisão. Este aumento de rendimento pode ser alcançado através do desenvolvimento de processos sistemáticos de monitorização e de planeamento e escalonamento da produção, recorrendo a abordagens e estratégias de simulação-otimização. Para o desenho dessas estratégias, o mapeamento das relações entre as variáveis do sistema em estudo, bem como a interligação entre os modelos de otimização e simulação, são aspetos fundamentais para que as soluções propostas representem com elevada fidelidade o sistema de produção e conduzam a um aumento real da eficiência.

Movidas pelo o aumento da competitividade, as empresas industriais procuram soluções que façam uma gestão otimizada dos seus recursos. As soluções passam, tipicamente, por estruturar e sistematizar os processos de planeamento de produção, usando métodos analíticos (i.e., modelos de otimização e/ou simulação) para lidar com um elevado número de produtos finais e reduzir os custos unitários. Contudo, com o aumento do mix de produção e a complexidade dos sistemas de produção, tem aumentado também os desafios de integrar e aplicar métodos analíticos nos sistemas de produção.

Este obstáculo introduziu no mundo dos sistemas de produção métodos e ferramentas de simulação, que em casos de complexidade ou impressibilidade dos sistemas, não pudessem ser resolvidos por métodos analíticos presentes até à data.

Os métodos de simulação apresentam-se como uma forte ferramenta de análise de soluções e fluxo dos processos. No entanto, as soluções ótimas não podem ser obtidas com recurso à simulação. Foi então necessário haver uma ligação das técnicas de otimização e simulação, que até hoje se consideravam alternativas ou incompatíveis.

Vários estudos, apresentados mais à frente nesta dissertação (Capítulo 2), vieram a ser feitos e comprovaram que a ligação entre os dois métodos trouxe melhorias na área do planeamento de produção. A metodologia abordada no desenvolvimento de métodos híbridos será apresentada na secção 1.4.

Ao longo do meu percurso académico áreas de decisão e gestão operacional sempre foram encaradas como áreas de curiosidade. Na possível exploração de diferentes perspetivas e interpretações, o tema então defendido nesta dissertação apresentou-se como um desafio interessante.

1.2 Contexto e Enquadramento

Planeamento significa atribuição de recursos limitados a tarefas de longo prazo com o objetivo básico de assegurar um uso eficaz e eficiente dos recursos disponíveis. Uma área “problemática” e de bastante importância é o planeamento de produção a nível industrial, nos quais as estações de trabalho (recursos) precisam de ser alocadas especificamente aos trabalhos (tarefas) da melhor maneira possível, minimizando ou maximizando o objetivo definido pela empresa, que para as condições do mercado e economia atual têm de se colocar em constante desenvolvimento, não só nos produtos e serviços oferecidos, mas como em estruturas, processos e produção interna.

Segundo *Gelders e Wassenhove* [3], planeamento de produção é atribuir um maior alcance aos objetivos de forma eficiente, rápida e económica, onde cada etapa de produção é prevista sugerindo correção de erros e respondendo as seguintes questões:

1. Em que ordem essa tarefa deve ser feita?
2. Quanto tempo deve ser gasto para essa tarefa?

Planear uma produção não é propriamente fácil e de rápida decisão, já que a ocorrência de imprevistos a nível do pedido, estações de trabalho em processamento, falhas no sistema, lotação de armazenamento, produção com falhas e *lead times* excessivos podem ocorrer. Para a abordagem defendida nesta dissertação, estes fatores complicam a resolução do planeamento de produção, contudo a ligação entre os modelos de otimização e simulação atenuam os seus efeitos. Termos como *lead time*, produto, *Workcenter*, sequenciamento, armazenamento e *bottleneck* são utilizados nesta dissertação, sendo a análise do sequenciamento, armazenamento e *bottleneck* bastante importantes.

A dissertação descreverá a aplicação do modelo de simulação com auxílio do modelo de otimização, gerando um planeamento de recursos a serem utilizados numa linha de produção, que posteriormente será comparado com a análise feita por projetos anteriores propondo assim uma nova solução. De todas as áreas existentes, este projeto incide na área de planeamento operacional em ambientes característicos de *Jobshop*.

O método de otimização procura encontrar o sequenciamento de tarefas o mais preciso possível, tendo como funcionalidade a procura pela melhor função objetivo de acordo com os dados de entrada do sistema – Secção 4.2.

O método de simulação, por sua vez, encontra o sequenciamento de tarefas que melhor se adapta ao sistema em estudo, podendo este não ser o ideal – Secção 4.3.

1.3 Objetivos

O planeamento de produção apresenta-se como uma área importante nos sistemas de produção de hoje em dia, levantando questões de como se pode reduzir custos operacionais, melhoria de desempenho e sustentabilidade de recursos. Nos últimos tempos a acessibilidade e competitividade a estes recursos provocou um desenvolvimento exponencial na procura e investimento nesta área.

Os principais objetivos desta dissertação são a investigação e aplicação de métodos de otimização e simulação que melhorem o planeamento de produção de sistemas e quais as possíveis iterações entre o otimizador e o simulador que poderiam beneficiar esses mesmos métodos. É de notar que o estudo ou abordagens a estes objetivos apresenta maior importância nas iterações entre os tipos de modelos.

1.4 Abordagem Metodológica

Como já referido na secção anterior, este projeto tem como foco principal obter a solução ótima no problema do planeamento de produção, mesmo quando estamos perante sistemas de grande complexidade e de tempos de análise e solução impraticáveis.

Modelos matemáticos conhecidos não são capazes de tratar a incerteza, enquanto que modelos de simulação, por outro lado, analisam sistemas complexos não lineares em tempo praticável. No entanto, métodos de simulação não são aptos a encontrar a solução ideal nem de provar quão ótima é uma solução. Analisando os dois métodos podemos então afirmar que as vantagens e desvantagens do método de otimização servem como soluções às incapacidades do método de simulação e vice-versa.

Para abordar o planeamento de produção e os problemas de programação, foi escolhida uma abordagem baseada na otimização. Com base em *Figueira e Lobo* [2], a metodologia proposta é categorizada como *Recursive Optimization-Simulation Approach* (ROSA) da categoria *Analytical Model Enhancement* (AME), como é resumidamente apresentado no capítulo 2. Quanto ao modelo matemático analisado neste projeto, este segue a formulação presente em *Byrne e Bakir* [1].

Este tipo de abordagem consiste em executar recursivamente um modelo analítico tipicamente linear e um modelo de simulação mais detalhado. Quanto à simulação, esta utiliza uma solução gerada previamente pelo algoritmo otimizador e determina medidas de desempenho específicas para o planeamento a ser estudado. Posteriormente os valores obtidos pela simulação são de novo testados pelo otimizador de modo a refinar as variáveis em uso e assim obter uma solução ótima. O processo recursivo termina quando o melhor valor é atingido ou quando se mantém um certo padrão de proximidade ou igualdade.

Incorporar uma lógica inovadora num modelo de simulação é uma tarefa bastante desafiadora. Para executar e obter uma melhor e rápida resposta, a ferramenta Simio fornece uma *Application Program Interface* (SIMIO API), sendo esta a principal razão pelo qual decidi selecionar o SIMIO

como o "*Simulation Manager*". Esta capacidade permite que os usuários sejam extremamente produtivos, oferecendo possibilidades ilimitadas de personalização e ampliação do modelo desejado. Neste projeto foi criado um *add-on* para o SIMIO usando a linguagem de programação C#, como podemos confirmar na secção 4.4. Ao executá-lo é possível criar e colocar automaticamente conjuntos de objetos Simio, que coletivamente formam o *Warehouse* pretendido, onde as propriedades e quantidades são criadas de forma automática. Para especificar todos os recursos, o utilizador apenas precisa de inserir os respetivos dados de quantidades de produção, períodos, tempos e características numa Folha Excel.

A solução proposta nesta dissertação aplica-se a um caso de estudo genérico de um sistema de produção, onde a demanda de produtos impulsiona a produção dos mesmos. O cenário de produção foi aplicado segundo *Byrne e Bakir* [1], contudo este cenário foi replicado para diferentes janelas temporais para ser possível analisar resultados de forma mais crítica e coerente. No modelo apresentado os produtos seguem uma determinada rota particular e apresentam tempos de processamento e custos específicos.

A estratégia abordada neste estudo soluciona problemas existentes em alguns dos modelos desenvolvidos até à data e impulsiona novas ideias que possam ser aplicadas ao problema Otimização-Simulação. *Bottleneck* do sistema, tempos de produção, sequenciamento das linhas, aumento de output e redução de custos são alguns dos fatores críticos em estudo.

1.5 Estrutura da Dissertação

A dissertação encontra-se dividida em 6 capítulos.

O presente capítulo apresenta o contexto e enquadramento do projeto, objetivos pretendidos desta pesquisa, motivação e metodologia abordada.

Este capítulo introdutório antevê uma revisão de literatura onde as técnicas existentes e usadas para resolver problemas de planeamento de produção são analisadas. São apresentadas breves descrições das metodologias de estruturação e ferramentas de apoio para este tipo de problemas. No capítulo 3 é dado a conhecer o caso de estudo, apresentação do modelo matemático que suporta o modelo de otimização e exposição à metodologia proposta. Quanto ao desenvolvimento da metodologia defendida, esta é explicada no capítulo 4. A análise de resultados das várias instâncias e a fiabilidade dos mesmos são apresentados e discutidos no capítulo 5. Por fim, no capítulo 6, a dissertação encerra com a apresentação das conclusões, contribuições e trabalhos futuros que possam ser aplicados à proposta e problema estudado.

Capítulo 2

Revisão de Literatura

Neste capítulo é realizada a revisão bibliográfica dos estudos que se focaram no tema abordado. É de referenciar que o método de pesquisa recorreu ao Google Scholar e a um conjunto de palavras chave.

Inicialmente foram encontrados documentos de primeira análise e conhecimento sobre a área em estudo, contudo foi numa fase em que os documentos se apresentavam repetitivos que se realizou uma análise detalhada, exaustiva e crítica. As referências consideradas com maior relevância foram agrupadas de acordo com a natureza dos métodos propostos.

Tabela 2.1: Palavras-chave pesquisa

. Optimization	. Production Scheduling	. Job shop	. Simulation Framework
. Simulation	. Production Planning	. Clear Functions	. Dispatching Rules

Os modelos de planeamento de produção podem ser caracterizados de muitas formas, mas a maioria das categorias sugeridas são específicas para as aplicações em consideração em *Figueira e Lobo* [2], *Juan e Figueira* [4] e *Azadivar* [5]. Esses modelos podem ser classificados de acordo com a dificuldade do problema, a sua relação com a realidade, a sua finalidade, a sua estimativa sobre a certeza dos seus dados e o seu domínio temporal. Segundo *Figueira e Lobo* [2], cada nível de abstração do modelo é diferente, isto é, o nível de detalhe no método de otimização é inferior ao método de simulação. Esta opção de modelagem é usada para evitar o aumento da complexidade na matemática da otimização. No entanto, o modelo de simulação compreenderá os recursos (variáveis) não incluídos no modelo de otimização.

Figueira e Lobo [2] apresentam uma análise detalhada e crítica sobre o problema dos resultados de modelos de planeamento de produção, apresentando-se como um documento de bastante importância para o estruturamento deste capítulo. Os autores dividem os problemas de modelos de simulação e otimização em três partes principais, sendo estes *Solution Evaluation* (SE), *Analytical Model Enhancement* (AME) e *Solution Generation Approaches* (SG).

A primeira abordagem corresponde à mais popular entre os modelos de simulação, onde a simulação é usada para avaliar as soluções e perceber até que ponto a resposta a uma escala temporal é correta. As outras duas abordagens combinam modelos analíticos com modelos de simulação, as quais têm grande interesse visto que a abordagem definida no meu projeto segue a mesma ideologia, podendo ser classificados como modelos de simulação-otimização híbrida. Estes são na sua maioria adotados dentro da comunidade de otimização.

Tabela 2.2: Divisão dos vários modelos de Simulação Otimização segundo *Figueira e Lobo* [2]

Solution Evaluation Approaches	
. Statistical Selection Methods (SSM)	. Metaheuristics (MH)
. Memory-based Metaheuristics (MMH)	. Random Search (RS)
. Stochastic Approximation (SA)	. Sample-Path Optimization (SPO)
. Matemodel-based Methods	. Gradient Surface Methods (GSM)
. Surrogate Management Framework (SMF)	. Reverse Simulation Technique (RST)
. Approximate Dynamic Programming (ADP)	. Retrospective Simulation Response Optimization (RSRO)
Solution Generation Approaches	
. Solution Completion by Simulation (SCS)	. Iterative Optimization-based Simulation (IOS)
Analytical Model Enhancement Approaches	
. Stochastic Programming Deterministic Equivalent (SPDE)	. Recursive Optimization-Simulation Approach (ROSA)
. Function Estimation based Approach (FEA)	. Optimization-based Simulation with Iterative Refinement (OSIR)

2.1 Abordagens Híbridas de Otimização e Simulação

No caso de modelos analíticos, *Juan e Figueira* [4] afirmam que estamos perante modelos que são uma abstração de um sistema real em termos de relações quantitativas descritas por expressões matemáticas. Como referido anteriormente, em [2] são descritos tipos de modelos onde existem procedimentos formais de soluções analíticas disponíveis para a otimização. Estes modelos têm sido frequentemente utilizados como ferramentas de solução no planeamento de produção de uma fábrica, como técnicas de programação matemática que dão resultados ótimos ou como abordagens heurísticas e metaheurísticas baseadas em lógica simples para resolver o problema. A escolha é geralmente baseada na forma de funções, complexidade, linearidade ou não-linearidade [2].

Uma discussão de abordagens de programação matemática para sistemas de planejamento é dada em *Missbauer e Uzsoy* [6] e *Pochet e Wolsey* [7]. Nestas duas abordagens os planos de produção tendem a superestimar a capacidade efetiva do sistema de produção, especialmente quando os recursos do sistema de produção têm altas taxas de utilização.

Em *Pahl* [8] para resolver este problema os modelos utilizam parâmetros adicionais tentando captar o comportamento não linear entre a carga de trabalho a que o sistema está sujeito e o tempo de produção.

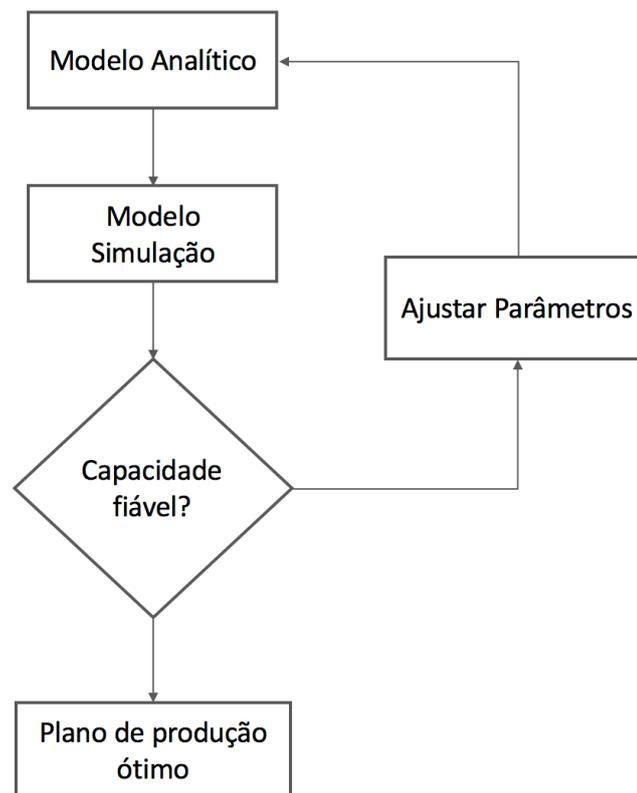


Figura 2.1: Procedimento normal de Otimização e Simulação Híbrida

Hung e Leachman [9] propuseram uma metodologia de planejamento de produção automatizada para o fabrico de semicondutores com base na otimização iterativa de um modelo de programação linear (LP), que usando estimativas dos tempos de fluxo determina um modelo padrão da saída dos produtos maximizando o lucro final, e um modelo de simulação de eventos discretos (DES) que toma como referência o planejamento determinado pelo modelo LP retornando os valores reais dos tempos de fluxo à saída de cada produto. Os valores reais medidos dos tempos de fluxo são então inseridos num novo modelo LP e o procedimento repete-se até que o critério de convergência seja satisfeito. Mais tarde um novo artigo defendido por *Hung e Hou* [10] substituí o modelo de simulação DES por um modelo de filas para reduzir o peso e tempo computacional gerado pelo modelo de simulação.

Este tipo de abordagem foi proposto por vários autores apresentando melhores resultados tanto a nível da ligação entre os dois modelos como a nível de soluções finais. *Byrne e Bakir* [1] propõem uma abordagem em que a utilização de recursos observada a partir do modelo de simulação em um dado período é usada para modificar o modelo LP que gere o planeamento de produção. Caso os resultados apontem para valores muito diferentes entre os dois métodos, o modelo de simulação ajustará um conjunto específico de parâmetros do modelo de otimização e repetirá o processo com condições mais restritas.

Mais tarde um modelo melhorado é apresentado por *Kim e Kim* [11]. Neste artigo são estendidos os dois procedimentos apresentados por [1] e [9] propondo uma nova medida de capacidade do modelo LP que simultaneamente estima os prazos e a capacidade real dos centros de produção (máquinas) ao longo de todos os períodos. O algoritmo base das variáveis de envio são estudadas em *Irdem e Uzsoy* [12], chegando à conclusão que o método converge de forma consistente apenas para um número baixo de iterações.

Posteriormente *Byrne e Hossain* [13] modificam o modelo analítico alterando as restrições de capacidade do produto, operação e probabilidade do centro de trabalho. *Moniz e Marques* [14] defendem o seguimento de [11] apresentando um modelo que efetua uma média dos produtos a entrar em produção em qualquer período desde que começa a produção.

Bang e Kim [15] abordam o planeamento de produção propondo uma solução iterativa de três fases. Em cada iteração os valores de tempo de espera e o nível de inventário a cada máquina da produção são atualizados como parâmetros de entrada do modelo de otimização. É sugerido um método de planeamento de produção hierárquica de dois níveis, onde a decisão de alto nível acontece para o planeamento de produção e a decisão de baixo nível para o agendamento.

Em [15] é apresentado um modelo idêntico a [9] modificando os parâmetros de envio entre o modelo de otimização e simulação, onde ao acréscimo dos tempos de fluxo atualizados é apresentado um reagrupamento de tipos de produtos podendo interromper a simulação caso o plano de produção seja considerado indesejável. O método foi desenvolvido para o contexto de produção de semicondutores caracterizado pela alta taxa de produção e variedade. Uma das considerações dos autores foi a ocorrência de eventos inesperados e o seu efeito sobre o método, uma vez que a abordagem de modelagem LP assume situações determinísticas na instalação.

Mahdavi e Solimanpur [16] expõem um sistema de suporte à decisão baseado em simulação. O modelo avalia entradas a tempo real e compara a saída do mesmo com a saída esperada. Um modelo analítico baseado nos métodos de superfície de resposta (RSM – *Response Surface Methodology*) calcula as alterações necessárias ao sistema para obter o resultado esperado. Essas alterações são executadas pelo modelo de simulação de forma iterativa até que a saída corresponda à saída esperada.

Estes procedimentos, onde o modelo de programação LP acoplado a um modelo de simulação cujo o objetivo é estimar decisões de planeamento, são testados mais tarde em diferentes grandezas dos sistemas de produção apresentando desempenhos bastante reduzidos em *Irdem e Uzsoy* [12]. Abordagens baseadas nos métodos de simulação e otimização híbrida analítica apresentam fortes fundamentos matemáticos e são garantidos para apresentar uma solução ideal. No entanto, o tempo

de execução pode tornar-se impraticável com o aumento das variáveis em estudo. A solução usual é incluir os parâmetros difíceis de modelar e de detalhes importantes no modelo de simulação, e aumentar o nível de abstração.

Tabela 2.3: Comparação de vários Artigos

Artigo	Estrutura	Planeamento	Regras de Despacho	Planeamento por Escalonamento	Escalonamento por Planeamento
<i>Hung e Leachman 1996</i>	Iterativa	Ordens de Produção	FIFO	Cronograma de lançamento de Produtos	<i>Lead Time</i>
<i>Byrne e Bakir 1999</i>	Iterativa	Ordens de Produção	FIFO	Cronograma de lançamento de Produtos	<i>Lead Time</i>
<i>Kim e Kim 2001</i>	Iterativa	Ordens de Produção	FIFO	Cronograma de lançamento de Produtos	<i>Lead Time, Utilização dos Recursos</i>
<i>Bang e Kim 2010</i>	Iterativa	Ordens de Produção	ES/RW2, MDBH	Cronograma de lançamento de Produtos, <i>Output</i>	<i>Lead Time, Crescimento de Produção</i>
<i>Moniz e Marques 2015</i>	Iterativa	Ordens de Produção	FIFO	Cronograma de lançamento de Produtos	<i>Lead Time, Utilização dos Recursos</i>

2.2 Meta-Heurísticas

Das abordagens referidas até agora é de notar que o autor *French* [17] afirma que com o aumento da complexidade dos sistemas de produção as suas soluções apresentam um mau resultado em relação ao tempo de execução do modelo de otimização. Uma forma de solucionar o problema é minimizar a informação presente no modelo, o que por vezes não é indicado. Assim, problemas de alta complexidade tendem a não encontrar rapidamente uma solução real ótima devido à incapacidade da sua computação em tempos admissíveis.

Para abordar problemas de otimização combinatoria e tempos de execução, *French* [17] e *Adams* [18] iniciaram o desenvolvimento e aplicação de meta-heurísticas em problemas de simulação – otimização. No entanto, para modelar um planeamento de sistema de produção devem ser incluídos parâmetros não-lineares, o que impossibilita o uso sozinho de meta-heurísticas. A combinação de meta-heurísticas com modelos de simulação é necessária.

As meta-heurísticas beneficiam de diferentes paradigmas de busca aleatória e paralelização, mas consideram que os *inputs* do problema e o conjunto de restrições do problema são determinísticos [4]. O funcionamento destas abordagens é idêntico ao dos modelos de otimização, onde as soluções enviadas através do modelo de simulação são afetadas pela solução heurística.

Nowicki e Smutnicki [19] com o objetivo de minimizar o tempo de produção num sistema que considera a existência de tempo de Setup das máquinas, aplica um algoritmo *Busca Tabu*. Nesta mesma abordagem, *Liu* [20] propõe um algoritmo genético (AG) que procura uma melhoria entre as fases de crossover e de mutação.

Shen e Buscher [21] apresentam uma abordagem onde o tempo de preparação da máquina é dependente da sequência de produção. A primeira fase do problema é agrupar todas as tarefas com um próximo grau de semelhança, diminuindo assim o grau de complexidade do problema, apresentando um número considerável de tarefas de planeamento. A segunda fase é aplicar um algoritmo *Busca Tabu* para chegar à solução.

Davoudpor e Ashrafi [22] apresentam uma solução para minimizar penalidades por atraso, por adiantamento, por tempo de conclusão e datas de entrega. Na formulação da solução cada fase é constituída por máquinas paralelas idênticas, os produtos apresentam diferentes datas de libertação de máquina e os *Setups* das máquinas são dependente da sequência planeada. [22] usa a metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) para resolver o problema.

Jeong e Kim [23] propuseram uma abordagem híbrida com a implementação de um AG. O AG é utilizado para gerar a melhor otimização de esquemas e a simulação para minimizar o tempo máximo de produção com horários fixos definidos no modelo. Os resultados experimentais indicam que o tempo de operação ajustado afeta significativamente o planeamento de produção. Trabalhos idênticos são também defendidos por *Zhou* [24] onde o AG determina qual a tarefa a ser processado inicialmente em cada máquina de produção e uma abordagem heurística determina o resto da sequência do plano de produção e por *Jeong* [23], onde é considerado que o tempo de conclusão é igual ao tempo de simulação, sendo este o tempo total gasto para executar todas as operações com base na solução gerada pelo AG. São gerados novos horários de simulação até que a diferença entre o tempo de simulação anterior e o tempo de execução atual esteja dentro dos valores definidos.

Liu [25] apresenta uma solução onde interliga uma função de otimização multiobjectivo, média e variação do custo total, baseada num AG com um modelo de simulação para apresentar uma sequência de produção. O método proposto por [25] aplica-se a situações em que há a necessidade de ponderar a compensação entre o custo médio e o risco associado a esse mesmo custo.

Sels [26] analisou um AG de procedimento de busca dispersa com o alargamento do problema para duas populações, onde efetuava trocas de indivíduos entre as populações obtendo maior diversidade e qualidade de informação.

AG e Meta-Heurística em geral, tendem a encontrar soluções mais rápidas do que os métodos analíticos comuns. No entanto, tais métodos não garantem a solução ideal a ser encontrada.

2.3 Clearing Functions

Recentemente, vários autores propõem a abordagem de modelos que relacionam a não linearidade entre a utilização de recursos e o número de produtos a entrar em produção. Estes métodos incluem a adição na função objetivo de um termo não-linear correspondendo ao custo do WIP, ou restrições baseadas em *Clear Functions* (CF) que relacionam o nível de WIP num determinado período com a produção esperada segundo Missbauer [27].

Irtem e Uzsoy [12] definem que as CF representam a relação entre a saída esperada de um produto num período do planeamento definido com o nível de WIP ou produtos a entrar em produção - *workload*.

Karmarkar [28] e *Graves* [29] apresentam um dos primeiros trabalhos com CF. Mais tarde, segundo *Kacar e Uzsoy* [12, 30], e *Purgstaller e Missbauer* [31] estes modelos foram estendidos e testados com resultados promissores, no entanto todos os testes foram desenvolvidos em pequenos sistemas com um máximo de 20 operações por produto, colocando em dúvida se estes modelos podem ser aplicados a sistemas de maior complexidade.

Segundo [28] as CF dependem da carga de trabalho esperada, da soma do WIP disponível e do material a entrar em produção no início de cada período. Em [30] os autores comparam o desempenho de vários modelos de regressão tanto para CF baseadas em *product-based* e *load-based*, não encontrando grandes diferenças entre as regressões apresentadas, contudo afirmam que CF baseadas em *product-based* geram consistentemente melhores desempenhos do que CF baseadas em *load-based* em um sistema sem *Setups*.

Em [12] e [32] é estudado se as CF servem de boa base para a resolução dos problemas de otimização e simulação de sistemas, chegando à conclusão que o problema principal das CF é a sua estimativa. As diferentes abordagens de regressão são comparadas e baseadas em resultados da simulação e resultados computacionais. Com os valores obtidos os autores puderam concluir que CF apresentam melhores resultados do que os métodos até à data, contudo são difíceis de estimar e demonstram uma forte ausência de bases matemáticas para apoiar os seus procedimentos.

Em [31] os autores analisam a integração de CF no ordenamento da solução de otimização com os métodos baseados em controlo de entrada de produtos, chegando à conclusão que os valores obtidos por esta abordagem são bem superiores aos obtidos até à data por abordagens idênticas.

Em [27] é apresentada uma secção de análise às limitações dos modelos CF, onde é possível concluir que as CF oferecem uma boa capacidade de incorporar características estocásticas para problemas de planeamento de produção, mas apresentam bastantes questões em relação à estimação de resultados e à sua formulação quanto a bases matemáticas

2.4 Simulação

Como referido anteriormente, os métodos de simulação são desenvolvidos em ferramentas de simulação onde são estudados e trabalhados os principais aspetos aplicados a sistemas de produção. Uma grande vantagem é acrescentada com este auxílio, passando pela análise a tempo real do comportamento de todo o sistema, deteção de falhas imprevisíveis ao coordenar todo um sistema complexo, monitorização de variáveis mais críticas e a possível análise de várias soluções estatísticas a tempo real.

Negahban [33] apresenta estudos sobre simulações de operações e projetos de sistemas de produção, onde afirma que este tipo de ferramenta tem tido um papel fundamental na área de análise, design e otimização de gestão industrial, diminuindo o risco e fortalecendo as decisões e desenvolvimento de cenários de estudo. Na tabela 2.4 estão representadas as áreas típicas de aplicação e alguns projetos desenvolvidos [34].

Tabela 2.4: Aplicação de Projetos de Simulação

<p>Fábrica de Produção</p> <ul style="list-style-type: none"> . Determinar tamanho ótimo de lotes a produzir para uma fábrica de semicondutores . Otimização do tempo de ciclo e a sua utilização no fabrico de semicondutores . Investigação em cadeia de abastecimento dinâmica orientada aos serviços . Análises estratégicas de armazenamento e recuperação em um armazém . Comparação de regras de despacho para o fabrico de semicondutores usando modelos . Comparação das regras de despacho usando modelos de pequenas dimensões
<p>Logística, Transporte e Distribuições</p> <ul style="list-style-type: none"> . Análise do fluxo de passageiros num terminal de aeroporto . Problemas logísticos em sistemas de produção de alimentos autónomos para exploração espacial de longa duração . Distribuição de produtos da indústria de informação -Jornais . Avaliar os potenciais benefícios de um algoritmo de planeamento para uma linha ferroviária
<p>Simulação de Processos de Negócios</p> <ul style="list-style-type: none"> . Planeamento estratégico da força de trabalho . Reconciliação da modelagem de negócios e sistemas
<p>Engenharia de Construção</p> <ul style="list-style-type: none"> . Construção especial de túneis utilitários . Renovação de infraestruturas urbanas subterrâneas . Planeamento e construção de um dique - barragem

O sucesso do sistema a ser desenvolvido pode ser influenciado tanto pela linguagem de programação como pela escolha do simulador a usar. Atualmente existem vários simuladores no mercado, contudo, segundo os seus fabricantes, apenas alguns se adequam à simulação de sistemas e

processos de produção de materiais.

O *Simio* é uma ferramenta destinada à interligação de objetos inteligentes [35]. É um programa que apresenta um vasto armazenamento de bibliotecas possibilitando assim que o utilizador desenvolva um projeto completamente gráfico, sem a necessidade de recorrer a códigos de programação [36]. Similar às ferramentas do mesmo género, um indivíduo, uma máquina ou outra entidade presente no modelo são exemplos de objetos, e a ligação entre eles pode ser um sistema em análise. A animação, processo utilizado para representar as mudanças de estado dos objetos com a fácil alteração entre 2D e 3D, e a lógica dos sistemas em uso são construídas simultaneamente tornando o processo de modelação bastante intuitivo. Esta ferramenta apresenta dois modelos de execução: o modelo experimental e o modelo interativo. No primeiro modelo é possível alterar as propriedades do sistema observando o impacto que este gera. O segundo modelo é útil para condições onde se está a construir e validar o modelo. Este estilo de simulação e modelação orientada a objetos torna-se bastante intuitiva durante a construção de um todo modelo, o que é um facto de mais valia para o utilizador [36].

O *Anylogic* é uma ferramenta destinada a simulações dinâmicas [37]. Segundo [37, 38, 39] este programa une eventos discretos, sistemas dinâmicos e métodos de agente *based Modeling* num só tipo e ambiente de programação. A linguagem é muito flexível e permite desenvolver complexos sistemas de negócios, económicos e sociais ao nível de detalhe desejado para alcançar uma melhor perceção dos processos constitutivos de uma empresa ou organização. Segundo [40] é possível usar e desenvolver bibliotecas pré-desenvolvidas de modo a utilizar projetos já feitos com uma estrutura orientada a objetos, tal como o *Simio*. Esta ferramenta apresenta uma interface gráfica e livrarias de objetos os quais permitem a modelação de diferentes áreas, desde o comportamento de entidades (consumidores, máquinas, produtos...), logística, manufatura e negócios. *Anylogic* apresenta um difícil uso, contudo é um programa de escolha quando queremos desenvolver projetos com mais complexidade de ligação entre estruturas e obter maior flexibilidade do projeto por completo.

O *Arena* é uma ferramenta mais destinada à ajuda de previsão, medição e demonstração de consequências de determinadas escolhas, permitindo aumentar a eficiência e otimização do projeto em estudo. Este software apresenta recursos para a modelagem de processos, análise de dados estatísticos, design de animações e por final análise de resultados. Disponibiliza um vasto conjunto de livrarias que se encontram já integradas no sistema, permitindo a fácil modelação de projetos nas áreas de manufatura, logística e reengenharia [41]. Essa modelação é feita construindo um modelo matemático-lógico que reflete a dinâmica dos processos e sistema, recorrendo a objetos orientados à simulação não sendo necessário conhecimentos aprofundados de linguagens de programação. Tal como o *Simio* e *Anylogic* dispõe de ferramentas de apoio à análise de resultados e comportamento global do sistema.

2.5 Regras de Despacho – Sequenciamento de produção

O sequenciamento de operações apresenta um papel importante em qualquer modelação de processos a nível da simulação, onde maximizar ou minimizar o nível de produção dos recursos definidos é uma prioridade. Segundo *Chase* [42] sequenciamento é o processo que deve ser definido à priori, onde se indica o tipo de tarefa ou atividade que deve ser iniciada em primeiro lugar, priorizando assim as tarefas a serem produzidas num determinado recurso.

De modo geral as regras de sequenciamento mais usadas pelos recursos são *First In First Out* (FIFO), *Earliest Due Date* (EDD), *Critical Ratio* (CR) e *Shortest Processing Time* (SPT). Segundo *Slack* [43] a regra de sequenciamento FIFO define que a primeira tarefa a entrar no sistema será a primeira a ser produzida não havendo quebras de produção ao longo do ciclo produtivo.

EDD define que a primeira tarefa a ser executada é a que apresenta uma data de entrega mais próxima à data atual. *Slack* [43] afirma que esta regra melhora a confiabilidade e a média de rapidez de entrega de um produto, contudo pode não introduzir no sistema uma produtividade ótima, diminuindo a eficiência deste e aumentando os custos totais. Em regra geral, EDD deve ser aplicado quando se pretende minimizar o máximo atraso das atividades do sistema.

SPT tem como prioridade a tarefa cujo tempo total de produção é mais pequeno. *Chase* [42] defende que esta regra deve ser acompanhada por alguma regra de atraso para que não haja tarefas a não serem produzidas. SPT apresenta melhores resultados no que diz respeito a critérios de tempo total de conclusão, isto é, médias de atrasos, esperas e tempos de conclusão.

CR define a diferença entre a data de entrega e a data atual dividida pelo tempo de processamento. Segundo *Nahmias* [44] e *Slack* [43] CR serve para balancear as regras SPT e EDD, que consideram apenas o tempo de processamento e data de entrega, respetivamente. Ao utilizar uma regra de sequenciamento CR as tarefas que apresentem um menor valor CR são priorizadas, tendo como características a data de entrega mais próxima e o tempo de processamento mais longo. O numerador pode tornar-se negativo, o que significa que a tarefa está em atraso, passando assim a ser prioritária. Caso exista mais do que uma tarefa em atraso é recomendável aplicar a regra SPT [43].

2.6 Análise Crítica da Literatura

Apresentados os vários métodos possíveis que melhor respondem ao problema de planeamento de produção, conclui-se que cada método distingue-se pelas suas vantagens e desvantagens.

Métodos de Otimização Híbrida, ao contrário das CF, apresentam um forte fundamento de modelos matemáticos para apoiar a modelagem analítica e garantir a solução ideal apesar do seu longo tempo de resposta, que por vezes pode tornar-se impraticável com o aumento da complexidade do modelo em estudo. Para solucionar esta desvantagem usa-se as ferramentas de simulação, onde detalhes e parâmetros importantes são incluídos no modelo.

Métodos Meta-Heurísticos tendem, em geral, encontrar soluções mais rápidas do que os métodos analíticos, apesar de não garantirem a solução ideal.

Métodos *Clearing Functions*, apesar de indicarem melhores soluções do que os métodos referidos em cima e proporcionar uma boa capacidade de incorporar características estocásticas nas abordagens escolhidas para cada modelo, apresentam uma difícil estimação de resultados e uma forte ausência de bases matemáticas que possam suportar tais soluções.

De entre todos os artigos referenciados é de salientar que *Figueira e Lobo* [2] disponibilizam uma vasta organização de conteúdos e esclarecimento sobre os vários métodos existentes, tendo contribuído em grande parte da estruturação desta dissertação, e *Byrne e Bakir* [1] e *Kim e Kim* [11] que por apresentarem um modelo matemático claro e bem estruturado servem de modelo base para solução presente neste projeto.

Quanto às ferramentas de simulação qualquer uma das ferramentas mencionadas é um forte candidato para o modelo em estudo. O *Simio*, como programa complementar ao método analítico, apresenta-se como a melhor escolha visto que fornece uma *Application Program Interface* (API), tornando fácil a interação entre os dois modelos para o projeto defendido nesta dissertação.

Quanto à escolha de regras de sequenciamento para o caso em estudo, todas as regras vão servir para complementar o modelo de simulação, uma vez que todas apresentam valores diferentes podendo ter fatores decisivos na escolha de produtos a apresentarem prioridade de produção.

No capítulo seguinte, a abordagem desenvolvida pelo modelo de otimização é explicada em detalhe, juntamente com as adaptações feitas pelas possíveis regras de sequenciamento escolhidas para o projeto e a interação entre os modelos de otimização e simulação.

Capítulo 3

Metodologia baseada em métodos de Otimização-Simulação

Cada um dos métodos de otimização e simulação apresenta diferentes vantagens e limitações quanto à sua natureza. Como já referido, a abordagem proposta nesta dissertação é categorizada como híbrida já que consiste no uso iterativo entre os diferentes métodos de otimização e simulação, onde a abordagem procura atenuar as suas limitações tirando partido dessas mesmas atenuações.

Quanto aos métodos de otimização, estes são bem conhecidos em garantias de soluções ideais e apresentam uma forte formulação matemática que se adapta ao problema em estudo, para além de poderem apresentar como desvantagem um aumento da complexidade e tamanho do modelo, e um longo tempo de execução.

Quanto aos métodos de simulação, estes contêm um elevado nível de detalhe, onde fatores não lineares podem ser aplicados sem comprometer a complexidade e tempo de execução. No entanto, apesar destas ferramentas apresentarem grandes capacidades determinísticas, a busca por uma solução ótima não pode ser feita apenas com recursos à simulação, dependendo assim da interação entre os dois métodos.

Neste capítulo é apresentada a descrição do problema resolvido, seguido pelo esclarecimento detalhado dos métodos de otimização e simulação, o estudo do problema, a interação entre os dois modelos, as possíveis regras de sequenciamento abordadas no problema e o critério de paragem pela abordagem dos métodos sugeridos.

3.1 Descrição do Problema

As decisões de planeamento de produção estão, grande parte das vezes, relacionadas com previsões temporais de médio longo prazo, onde fatores como melhoria de desempenho, redução de custos e maior *output* de produtos são críticos para a indústria de hoje, juntamente com aspetos

relacionados ao sequenciamento de ordenamento e operação. Sendo assim, o problema principal divide-se em duas fases – encontrar um plano de produção otimizado e, posteriormente, agendar o ciclo de produção de acordo com o plano obtido.

A escolha de qual o produto a produzir para um determinado cenário de produção influencia a decisão do tipo de sequenciamento que mais beneficia cada período de produção, sendo este um dos exemplos mais comuns para o agendamento de produção. Estes exemplos podem ser brevemente definidos como um conjunto de problemas onde, para cada produto, é necessário determinar a quantidade e o momento em que são libertados para a linha de produção.

Pelo exemplo anterior é possível a percepção da interdependência entre as duas fases, sendo a solução de uma a vantagem da outra.

Decisões como tamanho de lotes, flexibilidade de produção, *setup* de máquinas, horas de trabalho dos operários, *lead time* das linhas de produção e a quantidade de WIP no sistema podem classificar-se como fatores críticos, sendo os últimos dois impossíveis de modelar com restrições lineares. Quanto a restrições de capacidade, estas tendem a ser lineares, não se apresentando como revelantes para a modelagem do sistema.

Como mencionado na secção 1.4, a metodologia proposta visa solucionar problemas de planeamento de produção, tendo em conta responder às duas considerações apresentadas nesta secção.

3.1.1 Caso de Estudo

A solução proposta nesta dissertação aplica-se a um caso de estudo genérico de um sistema de produção, onde a demanda de produtos impulsiona a produção de equipamentos industriais.

A unidade de produção está composta por 4 áreas principais, onde cada produto tem a sua ordem de fabrico, das quais o *Workcenter* inicial e final indica, respetivamente, a entrada e saída dos produtos em linha.

A estratégia de produção é *Make-to-Order* (MTO), o que significa que o produto final só entra em produção quando existe a entrada de um pedido feito pelo cliente. Este tipo de estratégia de produção é caracterizado como do tipo *Pull* visto que os produtos são produzidos quando existe um pedido.

A alocação de recursos é feita de forma automática e é realizada durante a fase de planeamento. Eventos inesperados podem acontecer, tais como defeitos de produção e falhas dos *Workcenters*. Estes eventos não são considerados nas tarefas de planeamento de produção, o que leva a que haja uma nova execução de tarefas.

Os fluxos dos processos da instalação em estudo podem ser categorizados como um *Job-Shop*.

O sistema de produção é composto por diferentes *Workcenters* e produtos. Cada *Workcenter* apresenta a sua capacidade de produção e cada produto apresenta as suas características e tempos de fabrico. A inclusão de buffers em cada *Workcenter* é permitida assegurando o fluxo de materiais entre eles.

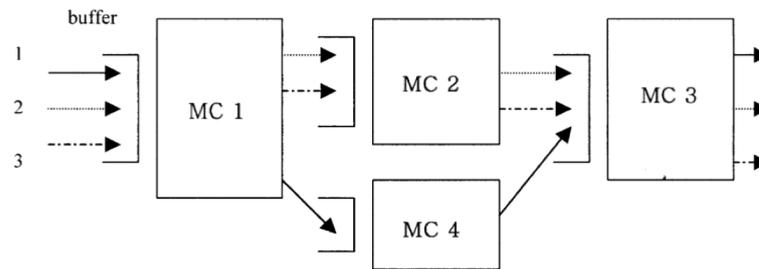


Figura 3.1: Esquema do Sistema Genérico

Nesta abordagem não existe *Bill-of-Materials* (BOM) o que permite que cada produto seja produzido de acordo com a ordem sequencial de fabrico, não havendo a necessidade de espera de produção de produtos para finalizar um produto final. Quanto à instalação, o armazenamento não se apresenta como um fator limitante no entanto o *WIP* excessivo pode significar um aumento de transporte de materiais e tempos de *setup* dos *Workcenters*, causando problemas logísticos internos.

Os tempos de operação e alocação de recursos variam conforme as características dos produtos a serem produzidos. Requisitos de material variam conforme o pedido exigido no início de produção. Quanto à capacidade do sistema de produção, esta é definida pela disponibilidade de unidades para processar, contudo apresenta um limite – Capacidade do recurso.

O *Workcenter* MC3 apresenta-se como o *bottleneck* do sistema devido aos longos tempos de processamento e ao facto de todos os produtos serem aí finalizados. Um dos métodos de estudo é a escolha da regra de sequenciamento aplicada ao sistema, apresentada na secção 4.6, para reduzir o tempo de espera do *Workcenter* MC3. A não procura da melhoria de desempenho por este *Workcenter* leva a que todo o sistema fique comprometido. Assim, o desenvolvimento de ferramentas e métodos que consideram não apenas as dependências entre alocações de recursos, mas também as regras de sequenciamento a usar, beneficiariam o desempenho global do sistema.

3.2 Metodologia Simulação Otimização

3.2.1 Otimização - Modelo Matemático

Como comprovado no capítulo 2, não é computacionalmente possível resolver problemas MIP envolvendo todas as informações de um sistema real por mais ou menos complexo que seja. Antes da formulação do modelo de otimização é necessário haver abstrações do problema real para que não haja uma formulação matemática “pesada”, reduzindo o tempo de execução do mesmo.

Para a formulação matemática foi usada a mesma abordagem utilizada por *Byrne e Bakir* [1]. Na secção 3.2.1.1 e 4.2 são apresentadas, respetivamente, as restrições feitas por [1] para

modelar o problema em estudo e o método de implementação do modelo matemático com recurso ao software de otimização *Gurobi*.

Para um sistema de produção, o lucro e custos estão relacionados de forma dependente. A função objetivo da formulação presente é a minimização de custos. Os custos estão presentes em três categorias: custo de produção, inventário e *backorder*. Cada custo está relacionado, respetivamente, com os custos de produção de um determinado produto da linha, armazenamento de um produto e o não cumprimento de uma ordem de produção em um determinado período. Para o caso em estudo, a matéria prima é considerada disponível após o pedido de produção ser efetuado, não necessitando assim de qualquer restrição para a espera do início de produção. Quanto aos *Workcenters*, são considerados iguais e o tempo de transporte entre as máquinas é considerado nulo, bem como os seus *Setups*.

3.2.1.1 Formulação Matemática

A estrutura do modelo é definida nos termos das seguintes tabelas, onde são apresentados *Sets*, índices, parâmetros e variáveis estudadas por [1].

Tabela 3.1: Elementos de Formulação – *Sets*

T	Número Total de Períodos
N	Tipo de Produtos
K	Tipo de estações de trabalho - <i>Workcenters</i>
J	Número total de iterações do modelo

Tabela 3.2: Elementos de Formulação – Índices

$i \in N$	índice do produto i
$t \in T$	índice do período t
$k \in K$	índice do <i>Workcenter</i> k

Tabela 3.3: Elementos de Formulação – Parâmetros

a_{ik}	tempo de processamento do produto i no <i>Workcenter</i> k
C_{kt}	capacidade (horas) do <i>Workcenter</i> k no período t
d_{it}	demanda do produto i no período t
c_{it}	custo de produção do produto i no período t
h_{it}	custo de inventário do produto i no período t
π_{it}	custo de <i>backorder</i> do produto i no período t

Tabela 3.4: Elementos de Formulação – Variáveis de Decisão

X_{it}	quantidade do produto i produzido no período t
I_{it}	quantidade de inventário do produto i produzido no período t
B_{it}	quantidade de <i>backorder</i> do produto i produzido no período t
E_{kt}	elasticidade do <i>Workcenter</i> k no período t

Conforme mencionado, a função objetivo deste problema é a minimização de custos totais, que engloba custos de produção, inventário e *backorder* dos produtos a cada período 3.1.

$$\min \sum_{t=1}^T \sum_{i=1}^N (c_{it} * X_{it} + h_{it} * I_{it} + \pi_{it} * B_{it}) \quad (3.1)$$

Quanto às restrições do modelo matemático, como qualquer sistema de produção, a utilização global dos recursos não pode exceder a capacidade máxima dos mesmos. Na seguinte restrição é medido o tempo que cada produto demora a executar a sua operação (produzir) em cada *Workcenter*, tendo este que ser menor do que o tempo máximo permitido para um dado *Workcenter* a cada período 3.2.

$$\sum_{i=1}^N a_{ik} * X_{it} \leq C_{kt} \quad (3.2)$$

Para garantir que haja um fluxo correto e equilibrado do material, é necessário criar uma restrição que relacione as quantidades de materiais entre os períodos de produção. Os produtos estão associados entre si, no entanto, as quantidades iniciais de produção são “espalhadas” pelos diferentes períodos, visto que os tempos de espera e a carga de trabalho dependem do estado em que se encontra o sistema 3.3.

$$I_{it} - B_{it} = I_{it-1} - B_{it-1} + X_{it} - d_{it} \quad (3.3)$$

Caso o sistema se encontre num período $t = 1$, os valores das variáveis B_{it-1} e I_{it-1} são nulas.

Na secção 4.1 será apresentado e discutido as várias instâncias para o mesmo cenário definas neste projeto. Para uma instância em específico - três semanas de trabalho com cinco períodos de oito horas em cada semana – a restrição 3.2 é reformulada em 3.4 e é acrescentada uma nova

restrição 3.5, que diz respeito à introdução de elasticidade no nosso modelo.

$$\sum_{i=1}^N a_{ik} * X_{it} \leq C_{kt} + E_{kt} - E_{kt-1} \quad (3.4)$$

$$E_{kt} \leq C_{maxk} \quad (3.5)$$

Uma vez que estamos na presença de períodos de trabalho muito pequenos, por vezes não é possível produzir ou finalizar a produção de determinados produtos causando um tempo neutro no fim do período. Para garantir que caso a capacidade de um *Workcenter* num determinado período t seja aproximadamente suficiente para produzir um determinado produto i , as restrições acrescentadas em cima garantem que esse produto seja lançado num determinado período t e possa exceder o tempo permitido pela capacidade do *Workcenter*, finalizando assim o produto no período $t+1$. Este novo fator foi restringido por 3.5, não podendo assim exceder o tempo máximo de produção mais longa de todos os *Workcenters*.

Quanto às variáveis de decisão, estas estão classificadas de acordo com a não-negatividade das variáveis inteiras 3.6.

$$X_{it}, I_{it}, B_{it}, E_{kt} \geq 0, \forall i, t, k \quad (3.6)$$

3.2.2 Modelo de Simulação Concetual

Várias técnicas de modelação de processos têm sido usadas ao longo dos anos. No entanto, a maioria dessas técnicas fornece pouco suporte na área de programação, tornando a lógica usada nos modelos de simulação completamente diferente nas várias técnicas de modelação.

Nesta dissertação a modelação de processos segue a mesma lógica de interpretação usada no caso de estudo (secção 3.1). Desta forma, pode notar-se um aumento na utilidade do modelo de simulação concetual, facilitando a sua programação, verificação e validação de cenários.

A figura 3.2 representa a ideologia do modelo base seguindo os critérios definidos no caso de estudo. Quanto aos elementos e as suas respetivas funções, estas são apresentadas na secção 4.3.

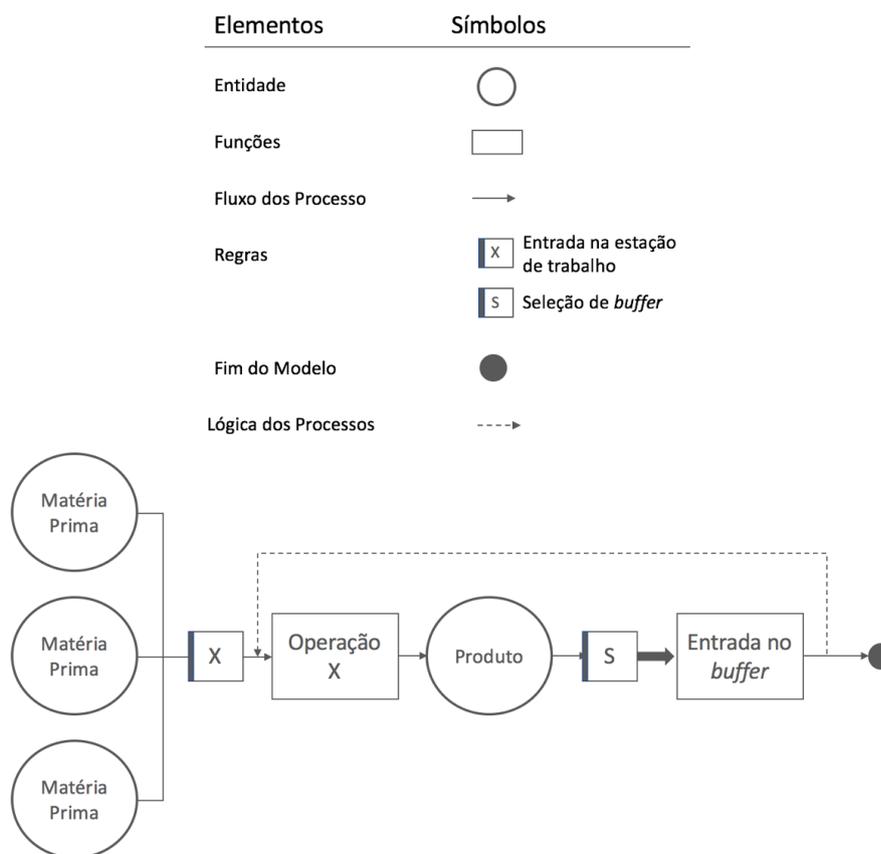


Figura 3.2: Metodologia desenvolvida no método de Simulação

Usando a simbologia presente na figura 3.2, a função Operação X ocorre quando uma específica entidade é pedida para processamento. Em seguida a peça processada é enviada para a seguinte estação de trabalho passando pelo *buffer* de entrada, uma vez que caso haja uma peça em processamento, esta é colocada em espera. A escolha da estação de trabalho depende do produto a ser produzido, já que estes apresentam uma rota definida.

Este processo é repetido até que todas as peças pedidas pela ordem de produção sejam satisfeitas. Toda esta interligação é feita com a definição de propriedades e processos.

3.2.3 Abordagem Simulação - Otimização

Como já referido nas secções anteriores, este projeto apresenta uma abordagem híbrida entre a otimização e simulação, onde, após a primeira execução os resultados da otimização são as entradas de ordens do simulador e os resultados da simulação são dados de análise do modelo de otimização.

A figura 3.3 traduz, de forma resumida, todo o processo que ocorre desde a entrada dos dados estruturais do sistema no modelo de otimização até à obtenção de um plano de produção ótimo.

As restrições impostas a cada iteração e o critério de paragem são apresentados nas duas próximas secções.

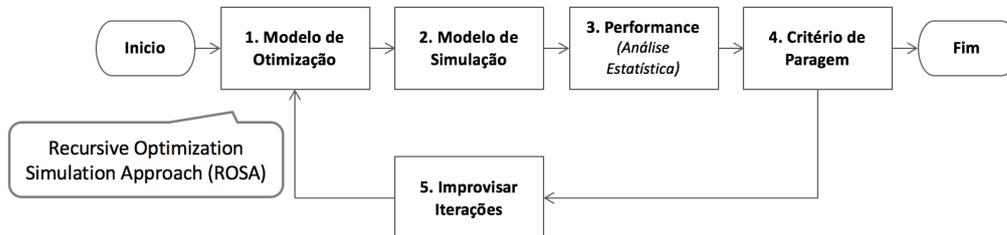


Figura 3.3: Esquema Abordagem Simulação - Otimização

O objetivo desta abordagem é melhorar o plano de produção lançado na Etapa 1 através da introdução de novos parâmetros e restrições no modelo.

- **Etapa 1** – Consiste na execução do Modelo Matemático para determinar o plano de lançamento por parte do simulador.
- **Etapa 2** – É executado o Modelo de Simulação com o plano lançado na Etapa 1.
- **Etapa 3** – As estatísticas sobre o número de produtos produzidos no plano inicial e final são guardadas, sendo comparada a diferença de produção entre os dois planos.
- **Etapa 4** – O algoritmo decide se dever ser novamente executado com bases na Etapa 3.
- **Etapa 5** – As restrições do Modelo Matemático são ajustadas conforme as diferenças apresentadas na análise das soluções obtidas.

3.2.3.1 Interação entre métodos

Uma vez implementados os dois métodos, é necessário definir algumas restrições ao longo do modelo que modifiquem o sistema de forma a que este chegue ao critério de paragem (secção 3.2.3.2), disponibilizando os melhores resultados ocorridos ao longo da execução do programa.

Na execução de ambos os métodos a estrutura de dados tende a ser a mesma. A quantidade de pedido de produtos, capacidades das estações de trabalho e prazos de entrega mantêm-se. A leitura de pedidos é direta e semelhante para os dois modelos, ou seja, o período em que o plano de otimização liberta um produto para a linha de produção é o mesmo período em que o plano de simulação liberta a mesma ordem de produção.

Para alcançar o critério de paragem e serem apresentados os resultados de cada iteração são propostas duas abordagens, denominadas de Modelo Base e Modelo Base Modificado. Estas duas

abordagens são quase idênticas, diferenciando-se no quanto cada modelo restringe as possibilidades de soluções – secção 4.5.

Na implementação dos modelos foi necessário definir novos parâmetros que definem a quantidade de produto lançado pelo modelo de otimização e produto produzido pelo modelo de simulação.

Tabela 3.5: Elementos de Formulação Exemplo dos Modelos – Índices e Parâmetros

Índices	
$i \in N$	índice do produto i
$t \in T$	índice do período t
$j \in J$	índice da iteração j
Parâmetros	
P_{it}^j	produção do produto i no tempo t na iteração j
R_{it}^j	lançamento do produto i no tempo t na iteração j

O Modelo Base e o Modelo Base Modificado seguem uma ideologia em que a ordem de produção indicada pelo método de otimização deve ir de encontro à ordem de produção obtida pelo método de simulação, uma vez que este apresenta maior aproximação da linha implementada no estudo genérico.

Sendo assim, uma das formas de aproximar os dois métodos aplicados a cada um dos modelos é restringir a diferença de quantidades de produção na iteração seguinte ($j+1$) 3.7.

$$P_{it}^j \leq R_{it}^{j+1} \leq R_{it}^j, \forall i \in N, j \in I, t \in T^* \quad (3.7)$$

, onde T^* é

$$\begin{cases} \text{se } j = 1, T^* = 1 \\ \text{se } j = n, T^* = 1, \dots, n \end{cases} \quad (3.8)$$

É de referir que o número de iterações nunca pode ser menor ou igual ao número de períodos definidos no início da execução do problema, uma vez que só é possível obter o critério de paragem com um número de iterações maior que o número de períodos (Capítulo 5).

Para o Modelo Base Modificado é necessário acrescentar uma nova restrição 3.9 e aplicar-lhe o somatório de um erro que aumenta as possibilidades de ordens de produção para o mesmo período numa iteração seguinte ($j+1$).

$$R_{it}^{j+1} = \frac{P_{it}^j + R_{it}^j}{2} + erro \quad (3.9)$$

O erro tem como características ser um valor aleatório e pertence a um intervalo de valores:

$$erro \in \{-2; -1,5; -0,5; 0; 0,5; 1; 1,5; 2\}$$

Na seguinte figura 3.4 está representada a abordagem ROSA com os parâmetros de retorno a serem alterados pelo modelo matemático.

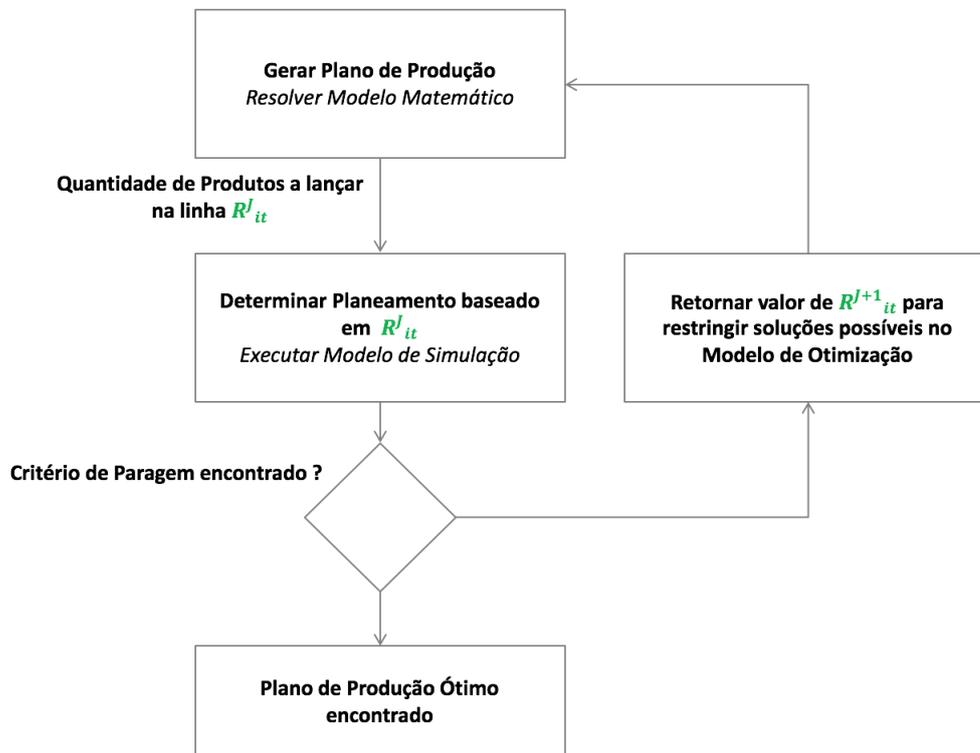


Figura 3.4: Esquema Abordagem ROSA

3.2.3.2 Critério de Paragem

A ligação entre os dois métodos em estudo tem de terminar sob a presença de alguma condição.

Na literatura de estudos feitos semelhantes ao projeto apresentado eram defendidas condições de paragem dependentes do resultado cíclico das soluções obtidas no final de cada execução, sequências de iterações sem diferenças significantes nas suas soluções e número de produtos finalizados de acordo com o pedido feito no início da execução do programa. Apesar dessas condições

serem válidas e poderem apresentar bons resultados, neste projeto o critério de paragem é o alcance de $\approx 0\%$ de erro entre o número de produtos lançados pelo otimizador e o número de produtos produzidos no simulador, garantindo uma convergência entre os dois modelos.

Como podemos comprovar através dos resultados apresentados no capítulo 5, a convergência dos métodos ocorre passadas algumas iterações, valor que aumenta com o aumento do número de períodos, como era de esperar. Nas iterações iniciais a percentagem de diferença entre o número de *output* de produtos dos dois modelos é bastante significativa, contudo após os ajustes que ocorrem ao longo do modelo de otimização este valor tende a aproximar-se de zero.

Condições relacionadas com valores temporais não foram implementadas, pois o tempo de execução está relacionado com a utilização média do *bottleneck* e da duração de cada período, tornando difícil a generalização de uma condição temporal.

Os resultados obtidos à abordagem proposta neste projeto são analisados no capítulo 5.

Capítulo 4

Desenvolvimento do Modelo

4.1 Dados da Instância Modelo

Para o desenvolvimento deste projeto foi usado como referência os valores apresentados por *Byrne e Bakir* [1].

Segundo [1] o cenário de estudo é constituído por 3 semanas e cada semana tem um período de quarenta horas de trabalho. Para cada período de trabalho é apresentado um pedido de produção constituído pelos seus custos de produção, inventário e *backorder*, que alteram de valor para diferentes cenários. Quanto à quantidade do pedido de produção para cada período, tempos de processamento e custos, estes são apresentados nas tabelas 4.1, 4.2 e 4.3, respetivamente.

Tabela 4.1: Pedido de produção de cada produto

Pedido Produto	Período		
	1	2	3
	1	150	125
2	100	150	150
3	125	165	125

Tabela 4.2: Tempo de Processamento de cada produto na respetiva *Workcenter*

Tempo Processamento (min) Produto	<i>Workcenter</i>			
	MC1	MC2	MC3	MC4
	1	5		4
2	7	7	5	
3	7	6	10	

Tabela 4.3: Custos associados à produção, inventário e *backorder* de produtos

Custo de produtos									
Custos	Custo Produção			Custo Inventário			Custo <i>Backorder</i>		
Períodos	1	2	3	1	2	3	1	2	3
Produto									
1	100	100	100	25	25	100	400	400	400
2	150	150	150	30	30	150	450	450	450
3	125	125	125	35	35	200	500	500	500

4.2 Implementação do Modelo Matemático

Uma vez definido o modelo matemático com a sua função objetivo e as suas restrições é necessário escolher um *software* otimizador. A escolha foi feita pelos critérios de desempenho, disponibilidade e fácil implementação. Existem vários optimizadores possíveis para solucionar o modelo, como *XPRESS*, *MIDALGO*, *CPLEX*, *GUROBI*, entre outros.

Para o modelo apresentado por *Byrne e Bakir* [1], é necessário um optimizador com capacidades de solucionar um problema LP e MILP. Sendo assim, qualquer optimizador seria uma opção viável, contudo *Gurobi Optimizer* [45] foi escolhido pela sua facilidade de programação.

O *Gurobi* suporta uma variedade de linguagens de programação, das quais está englobado C#, a linguagem escolhida para o desenvolvimento deste projeto, que para além da fácil interpretação de erros, apresenta-se como uma mais valia no desenvolvimento da interface API.

4.2.1 Optimizador – Estrutura de Dados

A escolha do tipo de dados a estruturar e a forma como este interagem entre si é fundamental para apresentar um modelo rápido, flexível e compreensível.

Como qualquer modelo de otimização é necessário definir previamente os dados do sistema, isto é, quantidades de pedidos, tempos de processamento, rotas, entre outros. No projeto desenvolvido, a leitura e escrita de dados, mais tarde a ser usada pela ferramenta de simulação, foi realizada com recurso a arquivos *Excel* – biblioteca *Microsoft Excel 11 Object Library Interop* [46]. Quanto aos dados de entrada, estes foram relacionados às informações relevantes para os índices e parâmetros apresentados na secção 3.2.1.1. Os dados de saída, por sua vez consistiam em ordens de produção e quantidades de inventário e *backorder* de cada produto para um determinado período.

Durante a construção do modelo a preferência por estruturas de listas e classes de objetos foi notória, uma vez que estes facilitavam a compreensão e utilização dos dados.

Outra forma de tornar rápida e fácil a compreensão de dados e detecção de erros (*debugging*) foi a indexação de produtos, *Workcenters* e períodos em números inteiros. Esta indexação permitiu também verificar que caso houvesse uma combinação de índices inexistente, em vez de atribuir um valor nulo a essa operação do modelo, esta era simplesmente ignorada, aumentando o desempenho do modelo.

Tabela 4.4: Relação dos índices com os dados do Modelo

Produto			<i>Workcenter</i>			
O1	O2	O3	C01	C02	C03	C04
0	1	2	0	1	2	3

Um dos estudos deste projeto foi a resposta do sistema à variação da quantidade e duração dos períodos de trabalho, que fazendo parte dos dados de entrada alteravam as ordens e quantidades de produção. A janela temporal inicial foi fixada segundo *Byrne e Bakir [1]*, que correspondia a três semanas de trabalho com um período de quarenta horas em cada semana.

De forma a estudar o impacto da alteração do mesmo cenário, foi então replicada a janela temporal para três semanas de trabalho com dois períodos de vinte horas em cada semana e para três semanas de trabalho com cinco períodos de oito horas em cada semana.

Tabela 4.5: Replicação entre janelas temporais – Períodos

Semana 1					Semana 2					Semana 3				
40 h					40 h					40 h				
20 h		20 h			20 h		20 h			20 h		20 h		
8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h

Esta variação de janelas temporais é importante para ver qual a relação e consequência entre os parâmetros e variáveis do modelo com o aumento e diminuição de períodos de trabalho, que apresenta as suas vantagens e desvantagens.

Um aumento do número de períodos de trabalho implica um aumento de parâmetros e variáveis, o que traduz um acréscimo na complexidade e tamanho do modelo. Portanto, o aumento do número de períodos causa um grande impacto no tempo necessário para resolver o modelo analítico, que se torna ainda mais crítico numa abordagem múltipla de iterações. Contudo, o aumento do número de períodos tem as suas vantagens. O aumento de períodos implica uma diminuição do respetivo período de trabalho, que por sua vez gera soluções mais precisas, onde o resultado obtido no modelo de simulação é tão próximo ou igual à solução do modelo de otimização.

Quanto às alterações de valores entre os cenários, é necessário explicar a relação entre os diferentes custos presentes no modelo e capacidades dos *Workcenters*. Para qualquer período de trabalho se considerarmos x , y e z o custo de produção, inventário e *backorder*, respetivamente, é de esperar que o valor de x se mantenha igual para qualquer cenário e que y e z se dividam pelo número de períodos por semana – $y/2$ e $z/2$ para o cenário de três semanas de trabalho com dois períodos de vinte horas em cada semana e $y/5$ e $z/5$ para o cenário de três semanas de trabalho com cinco períodos de oito horas em cada semana.

O mesmo acontece com as capacidades dos *Workcenters* – Caso w seja a capacidade de cada *Workcenter* (horas), este muda conforme a duração de cada período para os diferentes cenários, podendo w ser 40, 20 ou 8.

Quanto à quantidade a produzir, o pedido é considerado a cada início de período de trabalho e tal como a capacidade dos *Workcenters*, sofre uma alteração de valor para o estudo das diferentes janelas temporais. Na secção 4.1 é possível ver a quantidade do pedido de produção seguindo como referência *Byrne e Bakir* [1].

4.3 Simulação

Com o avanço tecnológico nas áreas da simulação cada vez mais são as ferramentas em constante evolução que permitem trabalhar com maior proximidade ao mundo da indústria.

Como referido no capítulo 2, o *software* disponível no mercado é vasto e equipado com as mais diversas ferramentas, que para além de fornecer uma licença livre *Personal Learning Edition* (PLE), é um dos softwares mais completos de pesquisa e permite a personalização completa de um modelo a partir de uma linguagem interna de programação.

Neste projeto é desenvolvido uma simulação por eventos discretos, o que implica que o estado do sistema mude em tempos discretos, e com um número finito de mudanças de variáveis em um determinado tempo.

O próximo passo é encontrar a melhor maneira de abordar o problema em estudo no método de simulação. Nas seguintes secções o sistema real é recriado, incluindo todos os tipos de variáveis que até agora não eram representadas.

4.3.1 Entidades

Em *Simio*, as peças, partes, operadores ou produtos são representadas por Entidades.

Para o projeto em estudo cada entidade terá as suas próprias características, seguindo os próprios tempos de processamento e roteamento em cada *Workcenter*. Visto que o modelo em estudo apresenta três produtos é necessário criar três entidades, denominadas de *O1*, *O2* e *O3*. De forma a ser possível a distinção de cada entidade ao longo da execução do modelo, estas são construídas com uma cor característica individual.

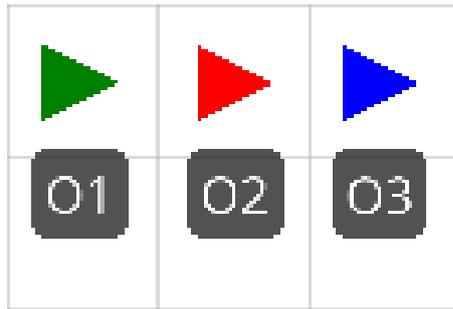


Figura 4.1: Entidades específicas para cada produto

No desenvolvimento das entidades é necessário definir as suas propriedades, que estão em constante alteração enquanto o modelo é executado.

Tabela 4.6: Propriedades das Entidades

Propriedade	Tipo	Descrição
idCode	Inteiro	Contém código do produto
Starttime	String	Guarda data e hora com que a entidade foi criada
CurrentMachine	String	Contém referência do <i>Workcenter</i> em que se encontra
periodo	Inteiro	Contém referência do período em que foi criada a entidade
Destino	Inteiro	Contém a referência do <i>Workcenter</i> para o qual se desloca
proctime	Real	Contém tempo de operação a realizar
DueDate	DateTime	Contém o tempo de entrega da ordem de produção
totaltimeRemaining	Real	Contém o tempo que falta até ocorrer o final das operações
Operation	String	Contém a referência da operação a cada estado do modelo

4.3.2 Construção do Modelo

4.3.2.1 Objetos

Mencionado no capítulo 2, *Simio* é uma ferramenta orientada a objetos, sugerindo assim que o primeiro passo deva ser a criação de objetos do modelo. Para a maioria dos sistemas desenvolvidos neste tipo de ferramentas é por norma necessário que haja pelo menos uma fonte de lançamento de entidades para o sistema, uma estação de trabalho para processar os mesmos e um coletor final de entidades, denominados *Source*, *Server* e *Sink*, respetivamente.

No desenvolvimento do modelo cerca de 19 objetos foram criados, dividindo-se nas seguintes categorias:

- *Entidades*: Como já referido, é um objeto que representa os produtos. Cada Entidade tem o seu nome e características.

- *Source*: Objeto que cria entidades. A criação de entidades é definida seguindo a ordem *O1*, *O2* e *O3*, contudo poderia ser definida aleatoriamente, já que o *Workcenter MC1* apresenta um *buffer* de entrada com capacidade ilimitada, não havendo necessidade de distinção entre entidades na fonte.

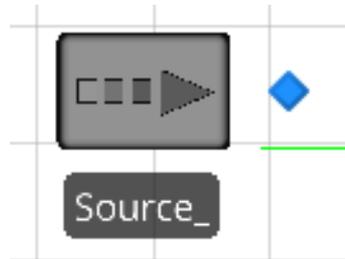


Figura 4.2: Objeto *Source*

Cada entidade é criada segundo propriedades definidas previamente na fonte em conjunto com as tabelas de estruturação de dados para o modelo em estudo. Uma das informações principais que deve ser aplicada à fonte é a quantidade de entidades que esta deve criar, e a partir de que data e hora as deve introduzir na linha de produção. Para tal é fundamental que o objeto *Source* esteja ligado com a tabela *demand* (secção 4.3.2.2).

À saída da fonte é introduzido o processo de Roteamento de Entidades (secção 4.3.2.3).

Properties: Source1 (Source)	
<input checked="" type="checkbox"/> Show Commonly Used Properties Only	
Entity Arrival Logic	
Entity Type	demand.idDemand
Arrival Mode	Arrival Table
Arrival Time Property	demand.releaseDate
Entities Per Arrival	demand.amount
Repeat Arrival Pattern	False
Stopping Conditions	
Maximum Arrivals	Infinity
State Assignments	
Before Exiting	1 Row
General	
Name	Source1
Description	

Figura 4.3: Propriedades *Source*

- *Server*: Objeto onde as operações das entidades são executadas. Cada *Server* recebe informação sobre os tempos de processamento de cada entidade. Inicialmente todos os *Servers* têm um *buffer* de entrada e saída com capacidade ilimitada, contudo dependendo da regra de sequenciamento aplicada, a capacidade do *buffer* de entrada do *Workcenter MC4* pode ser limitada. Isto permite que quando uma nova entidade chega à estação de trabalho e existe um

produto em processamento, ela é colocada em espera até que o produto em processamento passe para o *buffer* de saída.

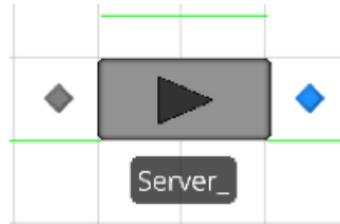


Figura 4.4: Objeto *Server*

Na implementação do modelo foram usados quatro servidores denominados por *Server1*, *Server2*, *Server3* e *Server4*, tendo cada um as suas características.

A primeira estação de trabalho do modelo é o *Server1*, que para além de processar todos os produtos do sistema apresenta as propriedades mais difíceis de implementar. Como referido na secção 4.6, algumas das regras de sequenciamento necessitam de maior complexidade para escolher o produto a ser processado.

Descartando a implementação base do FIFO, quando a entidade que se encontra em processamento termina a sua operação é necessário consultar o *buffer* de entrada e escolher a entidade com menor tempo de processamento. Para tal é fundamental definir qual o valor da expressão aplicada à posição de entidades no *buffer* de entrada, que segundo a Figura 4.5 se implementa na propriedade *Dynamic Selection Rule*. Outro fator importante é a adição de *add-on triggers*, que passam por ser os processos criados no modelo (secção 4.3.2.3).

Properties: Server 1 (Server)	
<input checked="" type="checkbox"/> Show Commonly Used Properties Only	
Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
<input checked="" type="checkbox"/> Dynamic Selection Rule	Smallest Value First
Value Expression	$Candidate.ModelEntity.procTime + ((Candidate.ModelEntity.DueDate - TimeNow) / Candidate.ModelEntity.totalTimeRemaining)$
Filter Expression	
<input checked="" type="checkbox"/> Processing Time	ModelEntity.procTime
Buffer Logic	
<input checked="" type="checkbox"/> Input Buffer	
Capacity	0
<input checked="" type="checkbox"/> Output Buffer	
Capacity	0
Add-On Process Triggers	
Before Processing	Server1_BeforeProcessing
Processing	RegistryFirstTimeValue
After Processing	RoutingAndProcTimeServers
General	
Name	Server 1
Description	

Figura 4.5: Propriedades *Server1*

Seguindo-se do *Server1* são criados os *Servers2*, *3* e *4*, sendo estes menos complexos e de fácil desenvolvimento.

Nesta abordagem os *Server2* e *4* são idênticos, diferenciando-se apenas no tipo de produto que produzem. Ambos apresentam uma produção FIFO, contudo é aplicada uma regra de ordenação das entidades presentes no *buffer*. O produto escolhido depende do seu tempo de processamento e do tempo em espera no *buffer*.

Quanto ao *Server3*, idêntico ao *Server1*, processa todos os produtos e para além de ser a estação de trabalho final, apresenta-se como o *bottleneck* do sistema. A propriedade *Ranking Rule* desta estação de trabalho é definida com *Earliest Due Date*, uma vez que não é vantajoso que os produtos sejam produzidos depois da sua *due date* definida.

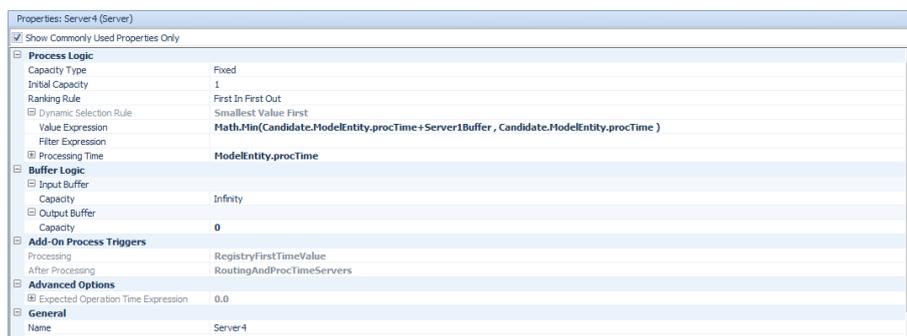


Figura 4.6: Propriedades *Server2* e *4*

- *Paths*: Objeto que interliga todos os outros objetos, definindo uma rota entre o início e o fim de produção. Cada *Path* foi definido como unidirecional, já que não existem produtos a serem produzidos mais do que uma vez na mesma estação de trabalho.
- *Sink*: Objeto que “destrói” as entidades.

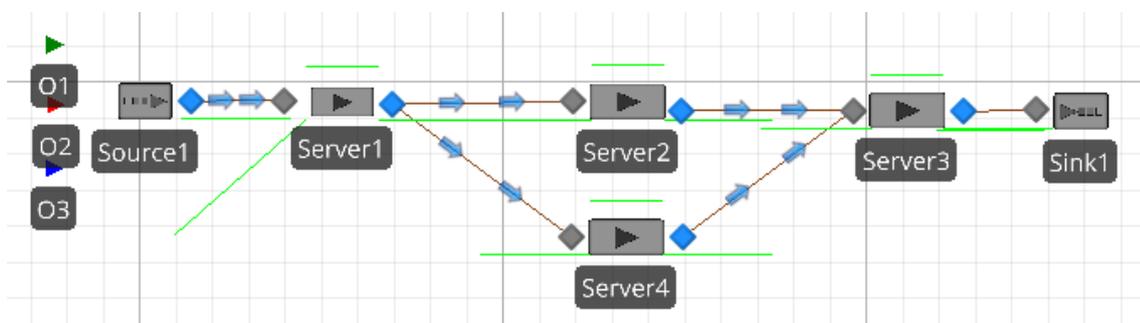


Figura 4.7: Modelo final com todos os objetos

4.3.2.2 Simulação – Estrutura de Dados

Para construir um modelo com alguma complexidade, o utilizador não pode fazê-lo apenas com recursos à *Standard Library* fornecida pelo *Simio*. Além das propriedades das entidades e objetos do modelo, a identificação de dados de entrada no modelo de simulação é bastante importante não só para a estruturação de processos, variáveis e propriedades, mas como também para a implementação do SIMIO API (secção 4.4).

Na implementação do modelo foram criadas três tabelas diferentes que foram posteriormente atualizadas pelo modelo de programação desenvolvido. Cada tabela é definida ao início da execução do programa contendo informação sobre a quantidade do pedido de cada produto por parte do cliente, tempos de processamento e sequencias de produção, sendo denominadas, respetivamente, por *demand*, *operationTimes* e *operationSequence*.

A tabela *demand* tem como objetivo definir a quantidade de cada produto a produzir ao longo dos períodos de trabalho. Apresenta também a data e hora a que o produto deve entrar e sair do sistema apesar de esta não ser real, já que é natural que ocorram atrasos na produção traduzindo-se numa entrada e saída tardia do produto na linha de produção. Cada operação tem o respetivo *idMaterial* (entidade) associado à quantidade a produzir “*amount*” e o seu *release Date* e *due Date*.

Esta tabela está associada ao processo de Roteamento de Entidades e à propriedade do objeto *Source* do modelo, uma vez que este processo e propriedade necessitam de informações relativas às quantidades a produzir e tempos a respeitar.

demand	operations Times		operation Sequence		release Date	due Date
	id Demand	id Material	amount	lot Size		
1	O1	O1	130	1	17/04/2017 00:00:00	18/04/2017 16:00:00
2	O2	O2	86	1	17/04/2017 00:00:00	18/04/2017 16:00:00
3	O3	O3	124	1	17/04/2017 00:00:00	18/04/2017 16:00:00
4	O1	O1	123	1	18/04/2017 16:00:00	20/04/2017 08:00:00
5	O2	O2	89	1	18/04/2017 16:00:00	20/04/2017 08:00:00
6	O3	O3	126	1	18/04/2017 16:00:00	20/04/2017 08:00:00
7	O1	O1	140	1	20/04/2017 08:00:00	22/04/2017 00:00:00
8	O2	O2	78	1	20/04/2017 08:00:00	22/04/2017 00:00:00
9	O3	O3	121	1	20/04/2017 08:00:00	22/04/2017 00:00:00
*						

Figura 4.8: Representação SIMIO tabela *demand*

A tabela *operationTimes* tem como objetivo definir o tempo de processamento que cada entidade apresenta ao longo das estações de trabalho que a produzem. Para a identificação do tempo de processamento é adicionada uma coluna *idOperation* que garante que a leitura é correta para a estação de trabalho em que se encontra a entidade.

Esta tabela está associada aos processos de Roteamento de Entidades e *Critical Ratio + Shortest Processing Time*, e às propriedades de cada estação de trabalho, uma vez que estes processos e propriedades definem o tempo que cada estação de trabalho deve gastar a produzir certa entidade.

demand	operations Times	operation Sequence			
	processingTime (Minutes)	id Resource	id Product	id Operation	
1	5	Server 1	O1	P01	
2	10	Server4	O1	P02	
3	4	Server3	O1	P03	
4	0	Sink1	O1	P00	
5	7	Server 1	O2	P01	
6	7	Server2	O2	P02	
7	5	Server3	O2	P03	
8	0	Sink1	O2	P00	
9	7	Server 1	O3	P01	
10	6	Server2	O3	P02	
11	10	Server3	O3	P04	
12	0	Sink1	O3	P00	
▶					

Figura 4.9: Representação SIMIO tabela *operationTimes*

A tabela *operationSequence* tem como objetivo definir quais as estações de trabalho que devem ser ordenadamente respeitadas pelas entidades a produzir.

É de mencionar que a tabela apresenta uma linha final de cada produto com o objeto *Sink* referenciado. Este objeto não apresenta tempo de processamento de entidade, contudo é necessário apresentá-lo uma vez que cada produto tem de ser eliminado do sistema, não sendo o ultimo objeto *Server* capaz de o fazer.

Esta tabela, juntamente com a tabela *operationTimes*, está associada aos processos de Roteamento de Entidades e *Critical Ratio + Shortest Processing Time*, e ás propriedades de cada estação de trabalho, uma vez que estes processos e propriedades definem ordenadamente as estações de trabalho em que cada produto deve ser produzido.

demand	operations Times		operation Sequence	
	id Operation	id Next Operation	id Product	id Resource
1	P00	P01	O1	Server1
2	P01	P02	O1	Server4
3	P02	P03	O1	Server3
4	P03	P00	O1	Sink1
5	P00	P01	O2	Server1
6	P01	P02	O2	Server2
7	P03	P03	O2	Server3
8	P04	P00	O2	Sink1
9	P00	P01	O3	Server1
10	P01	P02	O3	Server2
11	P02	P04	O3	Server3
12	P04	P00	O3	Sink1
▶				

Figura 4.10: Representação SIMIO tabela *operationSequence*

4.3.2.3 Processos

O *Simio* tende a facilitar a construção do modelo por parte do utilizador, onde basta arrastar os objetos para o local indicado definindo as suas propriedades, e o modelo deve estar pronto a ser executado. Este tipo de abordagem deve ser capaz de trabalhar nos modelos mais simples. Contudo, apesar de esta ser uma rápida e fácil construção de modelos, não garante flexibilidade ao modelo quando executado.

Quando se pretende ter complexidade e situações de estudo em que cada recurso está sujeito a várias tarefas e regras a respeitar ao mesmo tempo, é necessário desenvolver um modelo com recurso ao *add-on* processos da *Standard Library*. Um *add-on* processos é um conjunto de lógicas que são inseridas dentro dos objetos da *Standard Library* que, em certos pontos seleccionados, executa uma lógica personalizada alterando várias propriedades.

Os processos são criados como fluxogramas sem recorrerem a qualquer linguagem de programação. Assim, o *Simio* combina benefícios de modelagem baseado tanto no objeto como no processo gráfico.

Outra vantagem do uso de processos é a possibilidade de inserir num objeto diferentes definições conforme a instância em que se encontra, sem alterar as definições do objeto principal.

Os processos são compostos por *Steps*, *Elements* e *Tokens*. Os *Tokens* fluem através de um processo executado que altera um ou mais *Elements*. Nesta implementação todos os processos são considerados como *add-on* processos, o que significa que são executados por alguns *triggers* pré-definidos.

Em seguida são apresentados os processos mais relevantes para a construção do modelo.

Roteamento de Entidades

Para o roteamento de entidades ao longo de todo o modelo é necessário criar um conjunto de processos. Este conjunto de processos é acionado sempre que uma entidade entre e sai de uma estação de trabalho a partir dos *add-on triggers* definidos em cada bloco de propriedades das respectivas estações de trabalho. Isto significa que sempre que um produto entra ou sai de uma das estações de trabalho é criado um *Token* associado a esse mesmo produto, que segundo a figura 4.11, corresponde à identificação *begin* de cada processo.

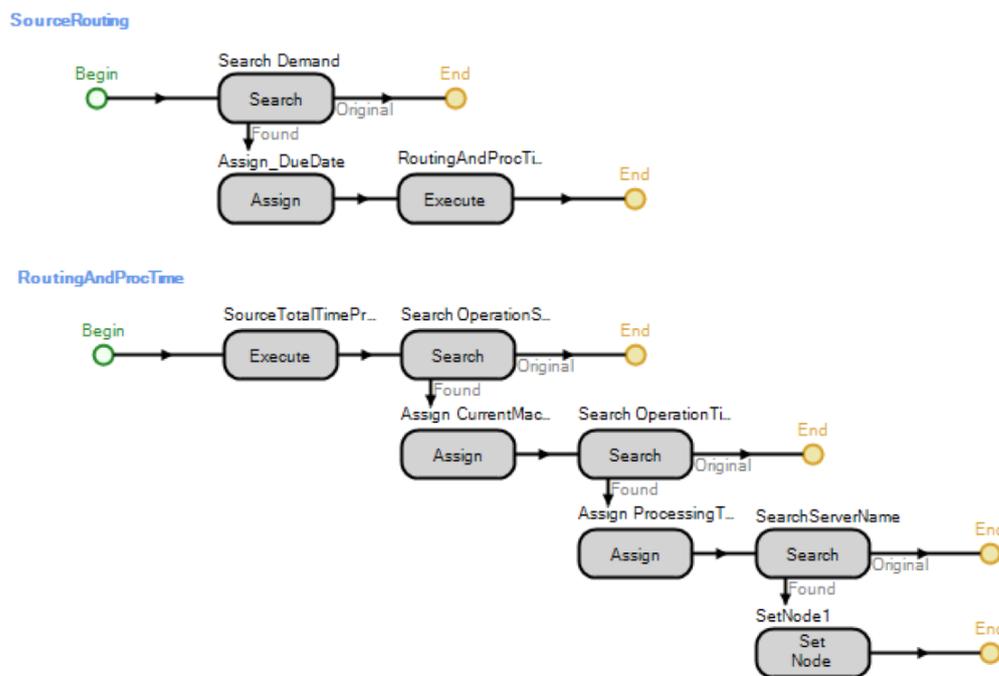


Figura 4.11: Processo Roteamento de Entidades

SourceRouting – Este processo é iniciado no objeto output *buffer Source* e tem como objetivo principal atribuir a cada entidade criada um valor de *due date* e a sua respetiva rota de produção.

- **Search Demand** – Esta etapa percorre a tabela *demand* configurada na estrutura de dados definida na secção 4.3.2.2 e encontra a entidade correspondente ao produto que acabou de ser produzido pelo objeto *Source*.

Em primeiro lugar é comparado o tipo de entidade associado ao processo com os tipos de entidades presentes na tabela. Em seguida, de forma a comprovar que não houve erro de

atribuições e que a tabela seja corrida de forma correta, é comparado o tempo presente da execução do programa com o tempo *due date* de cada produto, já que existe uma evolução crescente do tempo de execução do programa e do valor *due date* de cada produto.

- *Assign Due Date* – Este passo atribui à variável *DueDate* do modelo entidade o valor correspondente do *due date* definido nos dados estruturais do problema. Este valor é guardado até à destruição da entidade pelo objeto *Sink*.
- *Execute Block* – Após identificar e guardar o valor do *due date* de cada entidade produzida é necessário atribuir o sequenciamento de estações de trabalho que esta deve seguir. Nesta etapa é executado o segundo processo que engloba o Roteamento de Entidades – *RoutingAndProcTime*.

RoutingAndProcTime – Este processo é iniciado logo após a criação das entidades para a linha de produção e tem como objetivo atribuir a cada produto a sua rota definida antes de executar o modelo.

- *Execute Block – RoutingAndProcTime* inicia-se com a execução em paralelo de um outro processo – *SourceTotalTimeProcEntity* – que é desenvolvido na próxima estruturação de processos e é necessário para atribuir ao modelo regras de sequenciamento (secção 4.6).
- *Search OperationSequence* – Esta etapa percorre a tabela *operation Sequence* configurada na estrutura de dados definida na secção 4.3.2.2 e procura a operação que dá seguimento à operação atual. A correspondência encontrada depende da posição do produto em relação à estação de trabalho que se encontra.
- *Assign CurrentMachine and Operation* – É atribuído à variável *CurrentMachine* e *Operation* do modelo entidade o valor correspondente da estação de trabalho e operação em que a entidade se encontra. Este valor é guardado até que o produto seja produzido pela respetiva estação de trabalho em que se encontra.
- *Search OperationTimes* – Esta etapa percorre a tabela *operation Times* configurada na estrutura de dados definida na secção 4.3.2.2 e procura ao valor do tempo de processamento para a estação de trabalho em que a entidade se encontra.
- *Assign ProcessingTime* – É atribuído à variável *CurrentMachine* e *Operation* do modelo entidade o valor correspondente da estação de trabalho e operação em que a entidade se encontra.

Critical Ratio + Shortest Processing Time

Para a implementação da regra de sequenciamento CR + SPT (secção 4.6) é necessário criar um conjunto de processos. Este conjunto de processos é acionado ao longo de todo o modelo, uma vez que esta regra de sequenciamento avalia o tempo total que ainda existe de execução do

programa e escolhe a entidade que mais benefícios apresenta para diminuir o tempo de produção e *idle time* do *bottleneck*.

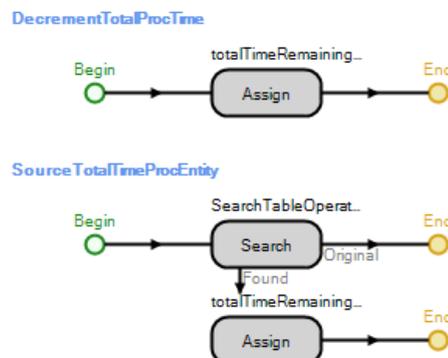


Figura 4.12: Processo Critical Ratio + Shortest Processing Time

DecrementTotalProcTime – Este processo é iniciado logo após a produção de cada entidade nas suas respectivas estações de trabalho e tem como objetivo decrementar o tempo total de produção. Está associado a um *add-on trigger* definido no objeto *output buffer* de cada estação de trabalho. Neste processo é aplicado apenas um *Step*:

- *Assign totalTimeRemaining* – É atribuído à variável *totaltimeRemaining* o valor que esta variável possuía anteriormente menos o tempo de produção que foi gasto por uma certa entidade na estação de trabalho que acabou de produzi-la. Este valor é atualizado cada vez que uma entidade é produzida.

SourceTotalTimeProcEntity – Como visto anteriormente, este processo é chamado durante a execução do processo *RoutingAndProcTime* e tem como objetivo determinar qual a contribuição de cada entidade criada para o tempo total que ainda existe de execução do programa. Este processo é aplicado no Roteamento de Entidades uma vez que é inicializado em primeiro lugar, contribuindo assim para definição do valor do *totaltimeRemaining*.

- *Search OperationTimes* – Esta etapa percorre a tabela *operation Times* configurada na estrutura de dados definida na secção 4.3.2.2 e procura ao valor do tempo de processamento para a estação de trabalho em que a entidade se encontra.
- *Assign totalTimeRemainingActualize* – É atribuído à variável *totaltimeRemaining* o valor que esta variável possuía anteriormente mais o tempo de produção que será gasto até que a entidade seja destruída. Este valor é avaliado a cada criação e produção de uma certa entidade.

Buffer Load Rule

Para a implementação da regra de sequenciamento BLR (secção 4.6) é necessário criar um conjunto de processos. Este par de processos é iniciado no *buffer* de entrada do *Workcenter* MC1 com o objetivo de analisar a capacidade do *buffer* de entrada de MC4. Como referido na secção 4.6, ao usar a regra de sequenciamento BLR é necessário definir um valor limite para o *buffer* de entrada de MC4. Caso esse valor atinja o seu limite é necessário ativar a produção de peças com não produção em MC4, inicializando assim a produção em MC2.

Na implementação destes processos foi necessário criar uma variável de estados denominada *Aceita* que tem como objetivo fornecer a informação de se o *buffer* de entrada do *Workcenter* MC4 chegou ao valor limite, ativando a produção em MC2. Esta variável é do tipo *Integer State Variable* e foi dimensionada como um vetor de 3 dimensões. Para além dessa variável foram introduzidos dois elementos de monitorização que juntamente com o par de processos apresentados, ativa ou desativa a produção em MC4.

Tabela 4.7: Elementos de Monitorização

Monitor Elements	Tipo de Objeto	Estação de Trabalho
Monitor4 OK	Monitor Element	Server4
Monitor4 FULL	Monitor Element	Server4

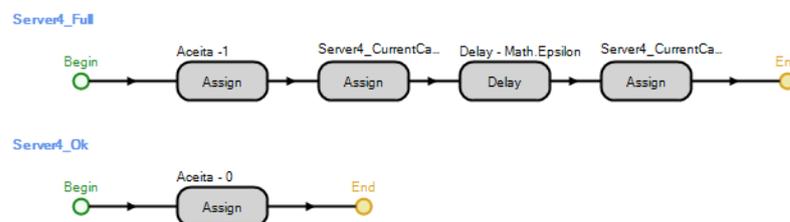


Figura 4.13: Processo Buffer Load Rule

Server4 OK – Este processo avalia a quantidade de entidades presentes na estação de trabalho *Server4* com o objetivo de desativar a produção no *Server2* e voltar a produzir no *Server4*. Neste processo é aplicado apenas um *Step*:

- *Assign - 0* – É atribuído à variável *Aceita* o valor de zero. Este valor é atualizado sempre que uma entidade entra no *Server4*.

Server4 Full – Este processo avalia a quantidade de entidades presentes no *Server4*, mas ao contrário do processo anterior, tem como objetivo ativar a produção no *Server2*. Quando o valor do limitador do *buffer* de entrada da estação de trabalho *Server4* é atingido, este processo é ativado.

- *Assign - 1* – É atribuído à variável *Aceita* o valor de um. Este valor é atualizado sempre que uma entidade entra no *Server4*.
- *Assign Server4 CurrentCapacity* – É atribuído à variável *CurrentCapacity* do *Server4* o valor de zero, informando o sistema que o *buffer* do *Server4* está cheio.
- *Delay Math.Epsilon* – Neste *Step* é implementado o intervalo mais curto de espera que a ferramenta SIMIO consegue medir. Ao usar este *delay* é garantido que todos os eventos a acontecer num determinado instante (imediatamente antes do *delay*) sejam executados.
- *Assign Server4 CurrentCapacity* – É atribuído à variável *CurrentCapacity* do *Server4* o valor de um.

Entrada de Entidades no Modelo

Para a entrada de entidades no modelo é necessário criar apenas um processo. Este processo é acionado, com auxílio aos *add-on triggers* definidos na propriedade de cada bloco, sempre que uma entidade entra no modelo e em cada objeto *Server*.



Figura 4.14: Processo Entrada de Entidades

RegistryFirstTimeValue – Este processo avalia e regista os valores da data e hora da entrada de cada entidade no sistema e a produção em cada estação de trabalho.

- *Execute Block* – Este processo inicia-se com a execução em paralelo de um outro processo – *DecrementTotalProcTime* – que tem como objetivo decrementar o tempo total de produção.
- *Assign StartTime* – São atribuídos valores às variáveis *StartTime* e *CurrentMachine* da data e hora retirados da posição da entidade ao longo do processo e das estações de trabalho em que se encontra aquando o registo da variável *StartTime*.

Saída de Entidades no Modelo

Para a saída de entidades no modelo é necessário criar apenas um processo, que é acionado no final de cada entidade ser produzida ao longo de todos os objetos do modelo. Este processo tem como objetivo principal registar todos os valores pertinentes para a análise de dados no final da execução do programa.

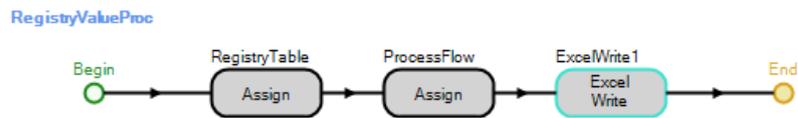


Figura 4.15: Processo Saída de Entidades

RegistryValueProc – Este processo avalia e regista os valores da data e hora da entrada de cada entidade no sistema e produção em cada estação de trabalho.

- *Assign RegistryTable* – É atribuído às variáveis *Name*, *Current Machine*, *StartTime* e *ProcTime* do modelo entidade o valor das suas características ao longo do modelo.
- *Assign ProcessFlow* – Permite a que os valores registados no *Assign* anterior não sejam reescritos, não havendo perda de informação.
- *ExcelWrite 1* – São registados todos os valores que são gravados numa folha Excel no momento final da execução do programa. Este *Step* cria uma tabela com os registos dos nomes das entidades criadas, estações de trabalho por onde passaram e o respetivo tempo de entrada da entidade no sistema e *due date* real.

4.3.2.4 Executar o Modelo

Uma vez construído o modelo com todas as propriedades definidas, o próximo passo é executá-lo. No *Simio* é possível executar o modelo de duas formas possíveis:

1. Modo Interativo
2. Modo de Experiência

Modo Iterativo

Este modo permite ao utilizador ver a evolução do seu modelo em “tempo real”, tanto os objetos, gráficos dinâmicos e parcelas que resumem o comportamento do sistema. Deste modo é possível obter uma análise crítica e validar o modelo conforme a sua evolução, obtendo uma noção básica de como as entidades criadas se movem ao longo do sistema criado.

Este tipo de execução é vantajosa para obter uma primeira análise visual do sistema, mas quando se pretende determinar ou executar cenários específicos, onde algumas propriedades são alteradas de modo a diferenciar os cenários e o foco é determinar a diferença no desempenho dos diferentes modelos, observar o modelo torna-se quase irrelevante já que para os olhos do utilizador é quase impossível determinar tais diferenças. A solução passa por replicar cada cenário na procura das diferenças implementadas no sistema e chegar a conclusões válidas a partir dos dados estatísticos fornecidos ao longo e final da execução.

Modo de Experiência

Este modo permite ao utilizador executar várias simulações com diferentes propriedades para que seja possível reconhecer e analisar as diferentes reações do sistema a estas alterações.

No modo de experiência é possível definir uma ou mais propriedades do modelo a alterar para ver qual o impacto no desempenho deste. Essas propriedades, expostas na experiência como *Controlos*, podem ser usadas para variar o número de *Workcenters* disponíveis, velocidades no transporte de produtos ou regras de decisão sobre qual a entidade a produzir referenciadas em um ou mais objetos do modelo.

Para ver as diferenças causadas no modelo é necessário adicionar Respostas, não sendo mais do que os principais indicadores de desempenho do modelo. Por vezes é necessário classificar dinamicamente as colunas criadas na estatística.

O utilizador pode também adicionar *Restrições* que serão aplicadas automaticamente antes ou após a execução do modelo de forma a prevenir uma má performance deste.

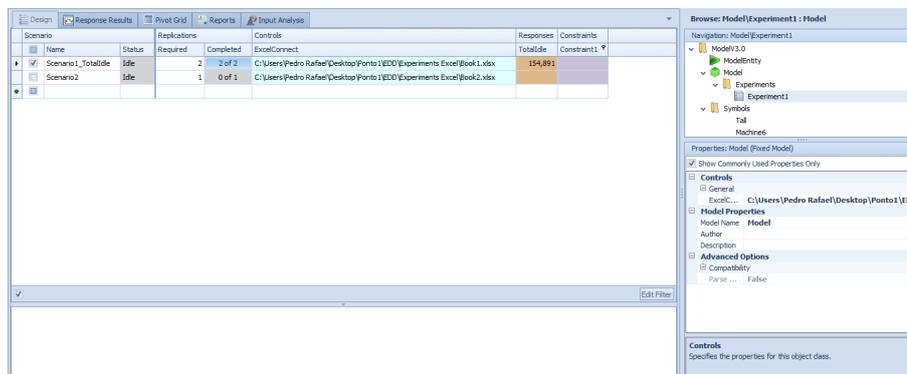


Figura 4.16: Executar um cenário com diferentes experiências – Modo de Experiência

4.4 SIMIO API

Encontrados e desenvolvidos os modelos de otimização e simulação é necessário interligar os dois. Como referido na secção 1.4, a ligação entre os dois modelos é desenvolvida segundo a linguagem de programação C# e tem como auxílio final o uso de uma API. Este *add-on* permite criar e alocar automaticamente cada entidade presente no sistema em estudo, que juntamente com os dados de entrada fixados no modelo matemático do otimizador gera de forma autónoma o planeamento de produção ideal.

Todo código de programação foi escrito a partir da biblioteca fornecida pelo simulador *Simio* que é disponibilizada na instalação deste. Mencionado na secção 4.2.1, a estrutura de dados passa por ser de apenas listas e classes de objetos. Quanto aos resultados obtidos, são guardadas as quantidades de produto, inventário e *backorder* em cada período de trabalho, custo total de produção do pedido e custos individuais de cada parcela em estudo para cada período t e produto i .

O próximo *flowchart*, Figura 4.17, representa, resumidamente, as interações e passos que são seguidos desde a escolha dos dados da instância a simular até á quantidade de produção final ótima. Como é necessário recorrer à biblioteca *SIMIO DLL*, o projeto base a desenvolver, já criado previamente no simulador, é aberto logo no início do programa e só depois são alocados recursos e entidades, carregamento de dados estruturais, corrida a otimização e simulação de forma iterativa e representação de resultados finais.

Outro fator que deve ser mencionado na construção do programa é o processo de inicialização do simulador através da API. *SimioProjectFactory* é uma classe estática na montagem do arquivo do projeto. Esta classe expõe o método *LoadProject* que leva uma sequência de informação contendo o arquivo a carregar, retornando uma referência para o *ISimioProjecto* que permitem aceder ao *IModel* demonstrando propriedades de acesso às tabelas, tabelas de funções e tabelas de taxas definidas no modelo, que, para este estudo, terá como valores os dados obtidos pelo modelo de otimização. Para testar várias possibilidades e encontrar diferenças entre alterações nos modelos implementados da *Warehouse* é utilizada a função *IExperiments*. Esta funcionalidade facilita não só a inicialização do projeto, como também a monitorização crítica das soluções do problema.

É de dar especial atenção à ligação entre os nós 5 e 9 (figura 4.17), já que este passo é realizado iterativamente até ser encontrada a melhor solução, que é definida pela percentagem de 0 % entre a diferença do *output* de produtos dado pelo otimizador e simulador. Este critério de paragem é referido e explicado na secção 3.2.3.2.

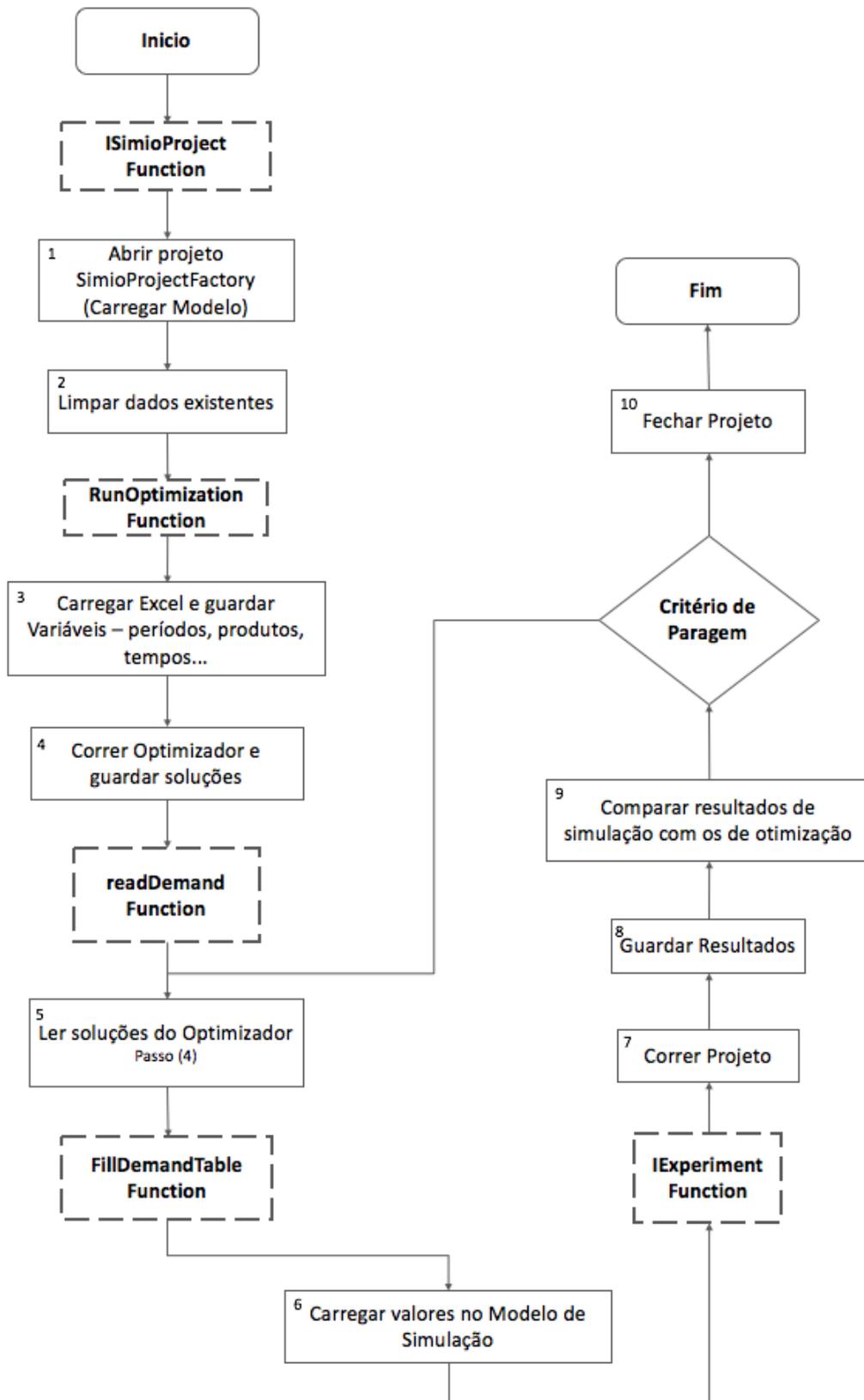


Figura 4.17: *Flowchart* interação entre Optimizador e Simulador

4.5 Modelos de Interação entre Métodos

Ambos os modelos desenvolvidos foram aplicados e analisados às várias janelas temporais presentes na secção 4.1, contudo para explicar cada abordagem foi escolhido o cenário temporal de três semanas de trabalho com um período de quarenta horas em cada semana.

Como mencionado na secção 3.2.3.2, para alcançar o critério de paragem e serem apresentados os resultados de cada iteração são propostas duas abordagens, denominadas de Modelo Base e Modelo Base Modificado. Os dois modelos respeitam a restrição 3.7, tendo o Modelo Base Modificado a adição da restrição 3.9.

4.5.1 Modelo Base

O estudo Modelo Base consiste, de forma geral, na restrição de produtos a produzir no método de otimização dependendo do número de produtos possíveis a produzir pelo método de simulação. A cada iteração que ocorra no sistema, valores do número de produtos a produzir na iteração seguinte são restringidos possibilitando a ocorrência do critério de paragem.

Na figura 4.18 está representado a resolução da instância pré-definida respeitando a abordagem do Modelo Base, encontrando o critério de paragem à iteração 4.

		Iteration Number		2		3		4	
Period	Product	Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador
1	1	150	129	129	129	129	129	129	129
	2	111	89	89	89	89	89	89	89
	3	124	124	124	124	124	124	124	124
2	1	125	105	144	129	129	129	129	129
	2	127	69	116	99	99	99	99	99
	3	126	126	124	124	124	124	124	124
3	1	160	79	162	106	177	127	127	127
	2	105	52	104	71	94	86	86	86
	3	123	123	123	123	122	122	122	122
Total Output		1151	896	1115	994	1087	1018	1018	1018
Total Cost		187925		255075		277200		302800	
Critério de Paragem		23,89		10,85		6,35		0	

Figura 4.18: Restrições definidas para o Modelo Base - Exemplo

Processo

Passo 1)

Inicialmente é corrido o método de otimização dependendo na estrutura de dados de entrada no sistema, como o pedido de produtos por período, quantidades e valores dos períodos, tempos de processamento, capacidades das estações de trabalho, entre outros.

Na figura 4.19 são apresentados os valores de output de cada produto em cada um dos períodos – R_{it}^j – para a execução do método de otimização. O próximo passo é executar o método de simulação para comparar a diferença entre os valores de output dos dois métodos.

		Iteration Number	
Period	Product	1	
		Otimizador	Simulador
1	1	150	
	2	111	?
	3	124	
2	1	125	
	2	127	?
	3	126	
3	1	160	
	2	105	?
	3	123	
Total Output		1151	?
Total Cost		187925	
Critério de Paragem		23,89	

Figura 4.19: Modelo Base – Passo 1

Passo 2)

Ocorrida a primeira iteração e encontrado o melhor plano de produção por parte dos dois métodos, figura 4.20, é necessário aplicar a restrição 3.7 para bloquear o número de produtos a produzir no período 1 da iteração 2.

Uma vez bloqueadas as quantidades a produzir no modelo matemático, é necessário encontrar as quantidades de *output* – R_{it}^j – do método de otimização para os períodos 2 e 3, visto que as quantidades bloqueadas correspondiam ao período 1.

		Iteration Number			
Period	Product	1		2	
		Otimizador	Simulador	Otimizador	Simulador
1	1	150	129		
	2	111	89		
	3	124	124		
2	1	125	105		
	2	127	69		?
	3	126	126		
3	1	160	79		
	2	105	52		?
	3	123	123		
Total Output		1151	896		
Total Cost		187925			
Critério de Paragem		23,89			

Figura 4.20: Modelo Base – Passo 2

Passo 3)

Adquiridos os valores de lançamento ótimos de produção é necessário estudar o comportamento do simulador – P_{it}^j . Como era de esperar os valores de output para o período 1 mantêm-se

iguais uma vez que a entrada dos valores bloqueados no modelo matemático para esse período eram a saída do simulador na iteração 1.

		Iteration Number			
Period	Product	1		2	
		Otimizador	Simulador	Otimizador	Simulador
1	1	150	129	129	129
	2	111	89	89	89
	3	124	124	124	124
2	1	125	105	144	?
	2	127	69	116	?
	3	126	126	124	?
3	1	160	79	162	?
	2	105	52	104	?
	3	123	123	123	?
Total Output		1151	896		
Total Cost		187925			
Critério de Paragem		23,89			

Figura 4.21: Modelo Base – Passo 3

Passo 4)

Semelhante ao passo 2, uma vez ocorrida a segunda iteração e encontrado o melhor plano de produção por parte dos dois métodos, figura 4.22, é necessário aplicar a restrição 3.7 para bloquear o número de produtos a produzir no período 2 para a iteração 3.

É de salientar que a quantidade de produção bloqueada para o período 1 (passo 2) mantém-se bloqueada, executando assim o método de otimização com as quantidades de $output - R_{it}^j$ – dos períodos 1 e 2 já definidas.

		Iteration Number					
Period	Product	1		2		3	
		Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador
1	1	150	129	129	129	129	129
	2	111	89	89	89	89	89
	3	124	124	124	124	124	124
2	1	125	105	144	129	129	129
	2	127	69	116	99	99	99
	3	126	126	124	124	124	124
3	1	160	79	162	106	?	?
	2	105	52	104	71	?	?
	3	123	123	123	123	?	?
Total Output		1151	896	1115	994		
Total Cost		187925				255075	
Critério de Paragem		23,89				10,85	

Figura 4.22: Modelo Base – Passo 4

Passo 5)

Uma vez obtidos os valores de lançamento ótimos de produção é necessário executar o simulador para analisar a quantidade de produção real – P_{it}^j .

Como era de esperar os valores de output para o período 1 e 2 mantêm-se inalterados uma vez que a entrada dos valores bloqueados no modelo matemáticos para esses mesmo períodos era a saída do simulador na iteração 2.

		Iteration Number	
Period	Product	1	
		Otimizador	Simulador
1	1	150	129
	2	111	89
	3	124	124
2	1	125	105
	2	127	69
	3	126	126
3	1	160	79
	2	105	52
	3	123	123
Total Output		1151	896
Total Cost		187925	
Critério de Paragem		23,89	

		2	
		Otimizador	Simulador
		129	129
		89	89
		124	124
		144	129
		116	99
		124	124
		162	106
		104	71
		123	123
Total Output		1115	994
Total Cost		255075	
Critério de Paragem		10,85	

		3	
		Otimizador	Simulador
		129	129
		89	89
		124	124
		129	129
		99	99
		124	124
		177	?
		94	
		122	

Figura 4.23: Modelo Base – Passo 5

Passo 6)

Semelhante ao passo 2 e 4, corrida a terceira iteração e encontrado o melhor plano de produção por parte dos dois métodos, figura 4.24, é aplicada a restrição 3.7 para bloquear o número de produtos a produzir no período 3 para a iteração 4.

Neste momento todos os valores já se encontram bloqueados, executando o método de otimização com as quantidades de *output* definidas até agora nos períodos 1, 2 e 3.

Como era de esperar chegamos ao final das iterações possíveis. O critério de paragem foi atingido à iteração 4, uma vez que os valores lançados pelo otimizador são iguais aos valores produzidos pelo simulador.

		Iteration Number	
Period	Product	1	
		Otimizador	Simulador
1	1	150	129
	2	111	89
	3	124	124
2	1	125	105
	2	127	69
	3	126	126
3	1	160	79
	2	105	52
	3	123	123
Total Output		1151	896
Total Cost		187925	
Critério de Paragem		23,89	

		2	
		Otimizador	Simulador
		129	129
		89	89
		124	124
		144	129
		116	99
		124	124
		162	106
		104	71
		123	123
Total Output		1115	994
Total Cost		255075	
Critério de Paragem		10,85	

		3	
		Otimizador	Simulador
		129	129
		89	89
		124	124
		129	129
		99	99
		124	124
		177	127
		94	86
		122	122
Total Output		1087	1018
Total Cost		277200	
Critério de Paragem		6,35	

		4	
		Otimizador	Simulador
		129	129
		89	89
		124	124
		129	129
		99	99
		124	124
		127	127
		86	86
		122	122
Total Output		1018	1018
Total Cost		302800	
Critério de Paragem		0	

Figura 4.24: Modelo Base – Passo 7

4.5.2 Modelo Base Modificado

Tal como o Modelo Base, defendido na secção anterior, o estudo do Modelo Base Modificado consiste na restrição de produtos a produzir no método de otimização dependendo do número de produtos possíveis a produzir pelo método de simulação.

A cada iteração que ocorra no sistema, valores do número de produtos a produzir na iteração seguinte são restringidos possibilitando a ocorrência do critério de paragem. Diferenciando-se do modelo anterior de forma a obter uma maior janela de soluções possíveis, este modelo respeita uma restrição adicional que implica que o valor de *output* de produtos a bloquear no lançamento da iteração seguinte seja mais ou menos o valor médio entre o plano lançado pelo otimizador e simulador. A esse valor é acrescentado um erro para variar a restrição anterior.

Na figura 4.25 está representado a resolução da instância pré-definida, respeitando a abordagem do Modelo Base Modificado, encontrando o critério de paragem à iteração 6.

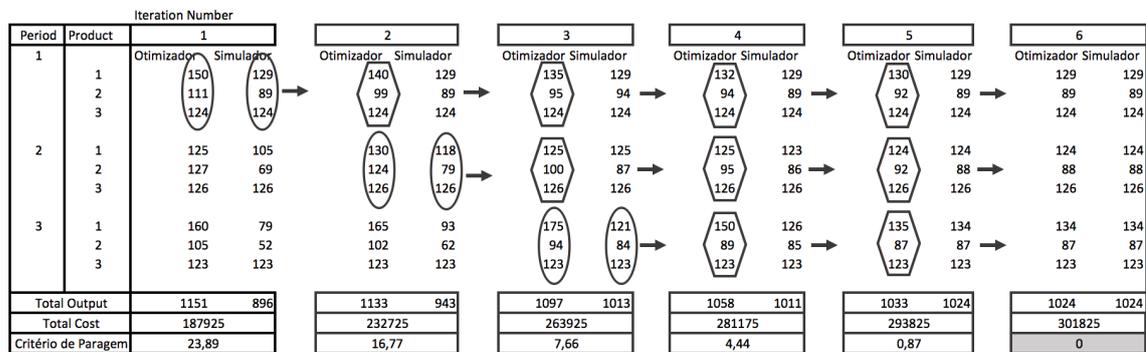


Figura 4.25: Restrições definidas para o Modelo Base Modificado – Exemplo

Processo

Passo 1)

Conforme indicado no primeiro passo do Modelo Base, inicialmente é corrido o modelo de otimização dependendo na estrutura de dados de entrada no sistema, como o pedido de produtos por período, quantidades e valores dos períodos, tempos de processamento, capacidades de estações de trabalho, entre outros.

Na figura 4.26 são apresentados os valores de output de cada produto em cada um dos períodos – R_{it}^j – para a execução do método de otimização. O próximo passo é executar o método de simulação para comparar a diferença entre os valores de lançamento e a produção dos métodos.

		Iteration Number	
Period	Product	1	
1	1	Otimizador	150
	2		111
	3		124
2	1		125
	2		127
	3		126
3	1		160
	2		105
	3		123
Total Output			1151
Total Cost			187925
Critério de Paragem			23,89

Figura 4.26: Modelo Base Modificado – Passo 1

Passo 2)

Ocorrida a primeira iteração e encontrado o melhor plano de produção por parte dos dois métodos, 4.27, é necessário aplicar a restrição 3.7 para bloquear o número de produtos a produzir no período 1 da iteração 2. Na abordagem deste modelo é necessário adicionar a restrição 3.9 para calcular o valor de R_{it}^{j+1} . De forma a simplificar a adição desta nova restrição, a tabela 4.8 demonstra a escolha dos valores para a iteração 2.

Tabela 4.8: Representação valores R_{it}^{j+1} - Passo 2

t	i	P_{it}^j	R_{it}^j	Erro $\in [-2;2]$	Equação 3.9 - R_{it}^j
1	1	129	150	0,5	140
1	2	89	111	1	99
1	3	124	124	0	124

Uma vez bloqueadas as quantidades a produzir no modelo matemático é necessário encontrar as quantidades de *output* - R_{it}^j - do método de otimização para os períodos 2 e 3, visto que as quantidades bloqueadas correspondiam ao período 1.

		Iteration Number			
Period	Product	1		2	
		Otimizador	Simulador	Otimizador	Simulador
1	1	150	129	140 99 124	
	2	111	89		
	3	124	124		
2	1	125	105		?
	2	127	69		
	3	126	126		
3	1	160	79		?
	2	105	52		
	3	123	123		
Total Output		1151	896		
Total Cost		187925			
Critério de Paragem		23,89			

Figura 4.27: Modelo Base Modificado – Passo 2

Passo 3)

Adquiridos os valores de lançamento ótimos de produção é necessário estudar o comportamento do simulador – P_{it}^j .

		Iteration Number				
Period	Product	1		2		
		Otimizador	Simulador	Otimizador	Simulador	
1	1	150	129	140 99 124		
	2	111	89			?
	3	124	124			
2	1	125	105	130		
	2	127	69	124	?	
	3	126	126	126		
3	1	160	79	165		
	2	105	52	102	?	
	3	123	123	123		
Total Output		1151	896	1133	?	
Total Cost		187925		232725		
Critério de Paragem		23,89		16,77		

Figura 4.28: Modelo Base Modificado – Passo 3

Passo 4)

Semelhante ao passo 2, uma vez ocorrida a segunda iteração e encontrado o melhor plano de produção por parte dos dois métodos, figura 4.29, volta-se a aplicar a ambas restrições e bloqueia-se o número de lançamento de produção no período 2 para a iteração 3.

A formulação de R_{it}^{j+1} é idêntica ao passo 2, contudo os valores de R_{it}^{j+1} para o período 1 voltam a ser recalculados.

Tabela 4.9: Representação valores R_{it}^{j+1} - Passo 4

t	i	P_{it}^j	R_{it}^j	Erro $\in [-2;2]$	Equação 3.9 - R_{it}^j
1	1	129	140	0,5	135
1	2	89	99	1	95
1	3	124	124	0	124
2	1	118	130	1	125
2	2	79	124	-1,5	100
2	3	126	126	0	126

Uma vez bloqueadas as quantidades a produzir no modelo matemático é necessário encontrar as quantidades de output – R_{it}^j – do método de otimização para o período 3, visto que as quantidades bloqueadas correspondiam ao período 1 e 2.

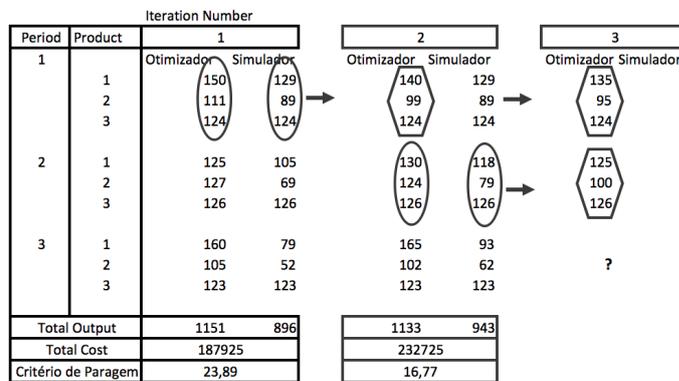


Figura 4.29: Modelo Base Modificado – Passo 4

Passo 5)

Obtidos os valores de lançamento ótimos de produção é necessário executar o simulador para analisar a quantidade de produção real – P_{it}^j .

		Iteration Number					
Period	Product	1		2		3	
		Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador
1	1	150	129	140	129	135	?
	2	111	89	99	89	95	?
	3	124	124	124	124	124	?
2	1	125	105	130	118	125	?
	2	127	69	124	79	100	?
	3	126	126	126	126	126	?
3	1	160	79	165	93	175	?
	2	105	52	102	62	94	?
	3	123	123	123	123	123	?
Total Output		1151	896	1133	943	1097	?
Total Cost		187925		232725		263925	
Critério de Paragem		23,89		16,77		7,66	

Figura 4.30: Modelo Base Modificado – Passo 5

Passo 6)

Idêntico aos passos 2 e 4, uma vez ocorrida a terceira iteração e encontrado o melhor plano de produção por parte dos dois métodos, figura 4.31, volta-se a aplicar ambas restrições e bloqueia-se o número de lançamento de produção no período 3 para a iteração 4.

A formulação de R_{it}^{j+1} é idêntica aos passos anteriores, contudo os valores R_{it}^{j+1} de para o período 1 e 2 voltam a ser recalculados.

Tabela 4.10: Representação valores R_{it}^{j+1} - Passo 6

t	i	P_{it}^j	R_{it}^j	Erro $\in [-2; 2]$	Equação 3.9 - R_{it}^j
1	1	129	135	0	132
1	2	94	95	-0,5	94
1	3	124	124	0	124
2	1	125	125	0	125
2	2	87	100	1,5	95
2	3	126	126	0	126
3	1	121	175	2	150
3	2	84	94	0	89
3	3	123	123	0	123

		Iteration Number		2		3		4	
Period	Product	Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador
1	1	150	129	140	129	135	129	132	?
	2	111	89	99	89	95	94	94	?
	3	124	124	124	124	124	124	124	?
2	1	125	105	130	118	125	125	125	?
	2	127	69	124	79	100	87	95	?
	3	126	126	126	126	126	126	126	?
3	1	160	79	165	93	175	121	150	?
	2	105	52	102	62	94	84	89	?
	3	123	123	123	123	123	123	123	?
Total Output		1151	896	1133	943	1097	1013	1058	1024
Total Cost		187925		232725		263925		281175	
Critério de Paragem		23,89		16,77		7,66		4,44	

Figura 4.31: Modelo Base Modificado – Passo 6

Passo 7)

Adquiridos os valores de lançamento ótimos de produção é necessário estudar o comportamento do simulador – P_{it}^j .

		Iteration Number		2		3		4	
Period	Product	Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador	Otimizador	Simulador
1	1	150	129	140	129	135	129	132	129
	2	111	89	99	89	95	94	94	89
	3	124	124	124	124	124	124	124	124
2	1	125	105	130	118	125	125	125	124
	2	127	69	124	79	100	87	95	88
	3	126	126	126	126	126	126	126	126
3	1	160	79	165	93	175	121	150	134
	2	105	52	102	62	94	84	89	87
	3	123	123	123	123	123	123	123	123
Total Output		1151	896	1133	943	1097	1013	1058	1024
Total Cost		187925		232725		263925		281175	
Critério de Paragem		23,89		16,77		7,66		4,44	

Figura 4.32: Modelo Base Modificado – Passo 7

Passo 8)

Semelhante aos passos 2, 4 e 6, ocorrida a quarta iteração e encontrado o melhor plano de produção por parte dos dois métodos, figura 4.33, volta-se a aplicar ambas as restrições e bloqueia-se o número de lançamento de produção de todos os períodos para a iteração 5. A formulação de está presente na seguinte tabela 4.11.

Tabela 4.11: Representação valores R_{it}^{j+1} - Passo 8

t	i	P_{it}^j	R_{it}^j	Erro $\in [-2; 2]$	Equação 3.9 - R_{it}^j
1	1	129	132	-0,5	130
1	2	89	94	0,5	92
1	3	124	124	0	124
2	1	123	125	0	124
2	2	86	95	1,5	92
2	3	126	126	0	126
3	1	126	150	-2	135
3	2	85	89	1	88
3	3	123	123	0	123

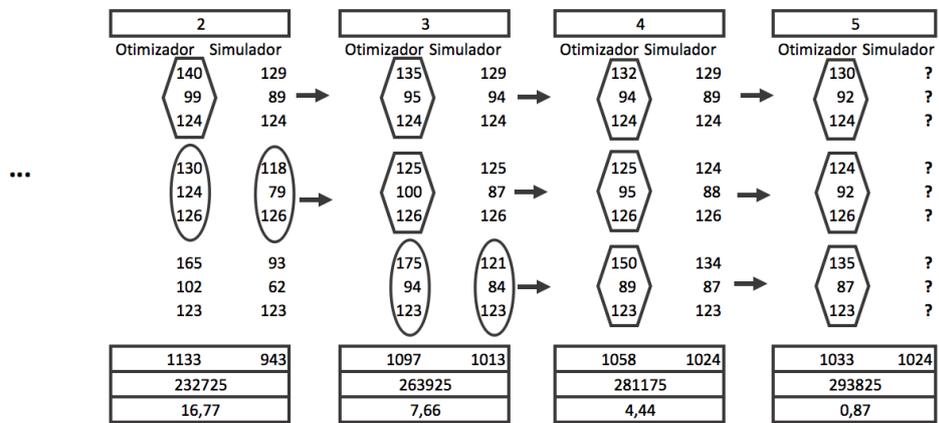


Figura 4.33: Modelo Base Modificado – Passo 8

Passo 9)

Adquiridos os valores de lançamento ótimos de produção é necessário estudar o comportamento do simulador – P_{it}^j .

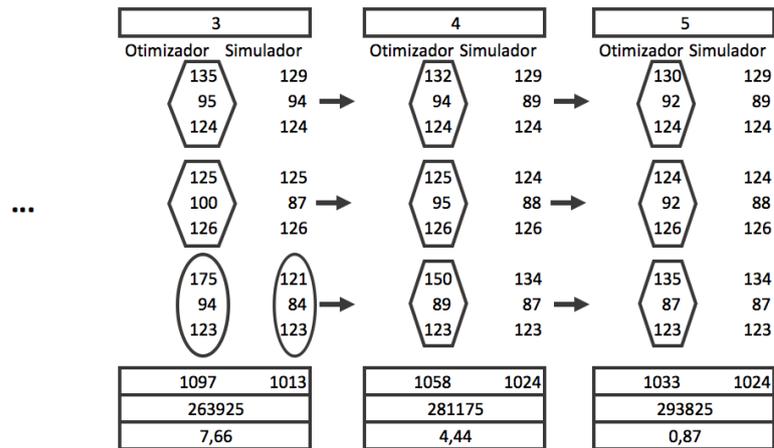


Figura 4.34: Modelo Base Modificado – Passo 9

Passo 10)

Semelhante aos passos 2, 4, 6 e 8, ocorrida a quinta iteração e encontrado o melhor plano de produção por parte dos dois métodos, figura 4.35, volta-se a aplicar ambas as restrições e bloqueia-se o número de lançamento de produção R_{it}^{j+1} de todos os períodos para a iteração 6. A formulação de R_{it}^{j+1} está presente na seguinte tabela 4.12.

Tabela 4.12: Representação valores R_{it}^{j+1} - Passo 10

t	i	P_{it}^j	R_{it}^j	Erro $\in [-2;2]$	Equação 3.9 - R_{it}^j
1	1	129	130	-0,5	129
1	2	89	92	-1,5	89
1	3	124	124	0	124
2	1	124	128	-2	124
2	2	88	92	-2	88
2	3	126	126	0	126
3	1	134	135	-0,5	134
3	2	87	87	0	87
3	3	123	123	0	123

Adquiridos os valores de lançamento ótimos de produção é necessário estudar o comportamento do simulador – P_{it}^j .

O critério de paragem foi atingido à iteração 6, uma vez que os valores lançados pelo otimizador são iguais aos valores produzidos pelo simulador.

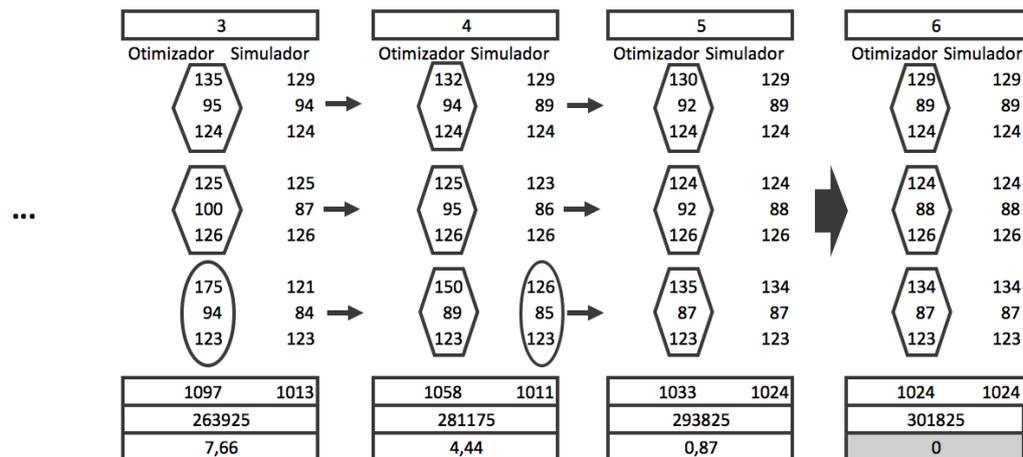


Figura 4.35: Modelo Base Modificado – Passo 10

4.6 Regras de Sequenciamento

Uma das formas de se manter o fluxo contínuo de produção é a atualização do sequenciamento dos produtos, que consiste em estabelecer uma programação sequencial destes produtos no fluxo produtivo, seguindo assim, uma sequência lógica de mais valia para o aumento de produtividade, redução de *lead time* e melhor fluxo de processos.

No capítulo 2 foram apresentadas as abordagens mais comuns de sequenciamento, contudo nesta secção vão ser sugeridas quatro regras de sequenciamento que mais se adaptam ao projeto em estudo.

Para as diferentes regras abordadas foram analisados os *delays* de cada estação de trabalho com especial atenção ao *bottleneck*, tempo total gasto na produção pedida ao arranque da linha de produção, e os custos de produção, inventário e *backorder*, presentes na secção 4.1.

A primeira regra a ser implementada no estudo deste projeto foi a *First in First out* (FIFO) e/ou *Earliest Due Date* (EDD). Neste caso, como o modelo está construído e a partir dos valores definidos nos tempos de processamento, não há diferença entre as duas regras. Ambos os modelos de sequenciamento apresentam os mesmos valores de tempo total de produção e *idle time*. Todas as *Workcenters* respeitam uma regra idêntica e capacidade de armazenamento infinita, à exceção de MC3 que respeita uma regra EDD para não intensificar os atrasos dos produtos relativos ao seu *due date*.

Para cada tarefa, a prioridade do produto i no período t é definida pelo tempo de chegada desse mesmo produto ao *Workcenter* k - 4.1.

$$Prioridade_i^t = TempoChegada_i^k \quad (4.1)$$

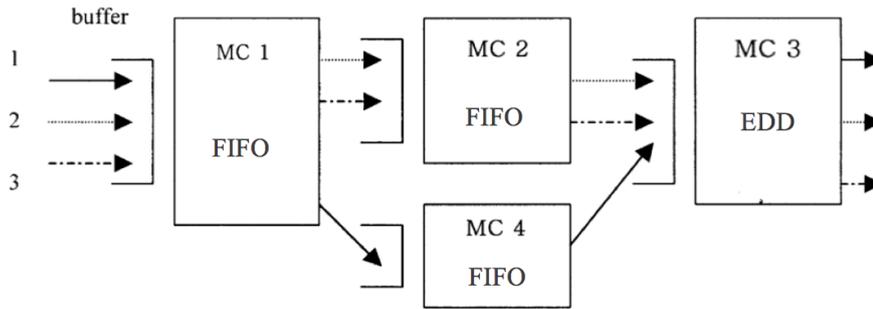


Figura 4.36: Esquema representativo do sequenciamento FIFO/EDD

Outra abordagem realizada sugeriu uma regra de sequenciamento que combina o *Critical Ratio* (CR) com *Shortest Processing Time* (SPT), denominada de CR + SPT. Se a tarefa de produção do produto tiver folga, a regra CR é aplicada à fila, caso contrário, a regra SPT é escolhida como prioritária.

Para cada tarefa é calculado o CR, e posteriormente é comparado ao tempo de processamento da mesma. Por fim, é escolhido o valor que dá maior prioridade à tarefa entre o CR e o tempo de processamento.

$$CR_i = \frac{Data de Entrega_i - Tempo Atual}{Tempo Processamento Restante} \quad (4.2)$$

$$Prioridade_i^t = \max\{CR_i * Tempo Processamento_i; Tempo Processamento_i\} \quad (4.3)$$

Nesta abordagem todos os *Workcenters* respeitam a mesma regra à exceção de MC3. Visto que MC3 é o *bottleneck* do modelo e os tempos de processamento de cada produto indicam um elevado valor para MC3, é de esperar que este *Workcenter* nunca pare. Sendo assim, EDD foi aplicado para que não houvesse espera de produtos que deveriam ser produzidos num período $t-1$.

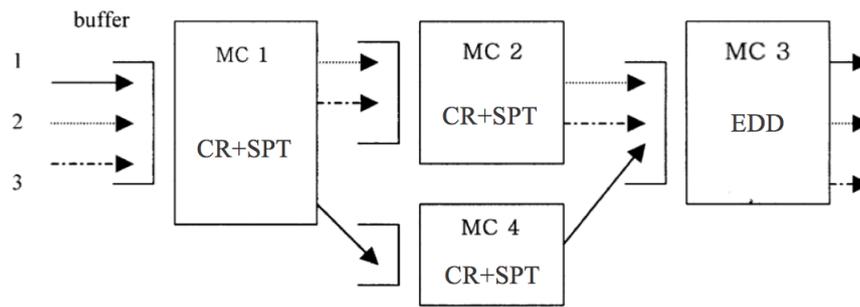


Figura 4.37: Esquema representativo do sequenciamento CR + SPT

Um dos problemas que sugeria modificações foi o tempo em que os *Workcenters* estavam parados à espera de entrada de produtos. MC2 e MC4, como *Workcenters* de ligação entre o início e fim de linha, necessitavam do acréscimo de uma nova regra de sequenciamento às regras já existentes. Sendo assim, foi desenvolvida uma regra *Buffer Load Rule* (BFR), que relacionava MC2 e MC4 permitindo um processamento de produtos em paralelo.

Na implementação de BFR é definido um limite de capacidade de armazenamento em MC4. O *buffer* limitador foi apenas definido numa dos *Workcenters*, visto que para o *flow* obtido e tempos de processamento do modelo estudado, o processamento de produtos é bastante rápido. A escolha do *Workcenter* MC4 como limitador do *buffer* é feita uma vez que, as regras desenvolvidas anteriormente (FIFO e CR + SPT) sugeriam que para os tempos de processamento definidos na secção 4.1, MC4 iniciasse a sua produção antes de MC2.

Segundo o esquema representativo seguinte, figura 4.38, BLR é aplicada a MC1 para esta poder escolher o produto a produzir de acordo com o valor do *buffer* em MC4. Na implementação foi definido para o *buffer* um limite máximo de 4 produtos, beneficiando a produção em paralelo para que MC3 trabalhe o maior tempo possível.

Quanto às restantes *Workcenters*, estas respeitam a regra FIFO, à exceção de MC3 que respeita uma regra EDD para não intensificar os atrasos dos produtos relativos ao seu *due date*.

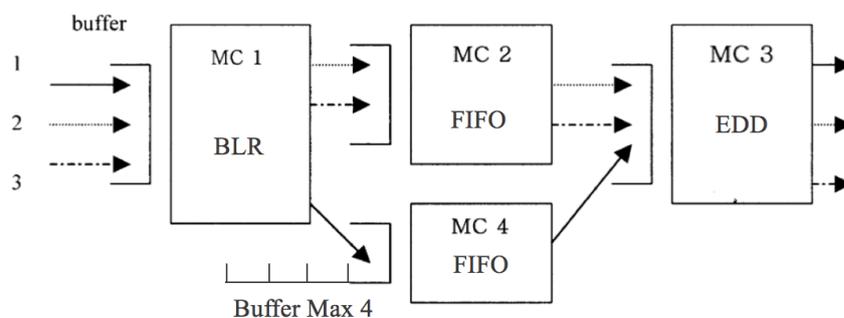


Figura 4.38: Esquema representativo do sequenciamento BLR

Outra abordagem realizada foi interação entre as regras de sequenciamento BLR e CR + SPT, que idêntica à regra anterior é aplicada a MC1 para esta poder escolher o produto a ser produzido de acordo com a prioridade atribuída por CR + SPT e o valor do *buffer* em MC4. O valor limite máximo do *buffer* foi mantido a 4 produtos.

Quanto à prioridade do produto, esta segue os mesmos cálculos apresentados em 4.2 e 4.3. Nesta abordagem todos os *Workcenters* respeitam a mesma regra à exceção de MC3, que aplica uma regra EDD para que não houvesse espera de produtos que deveriam ser produzidos num período $t-1$.

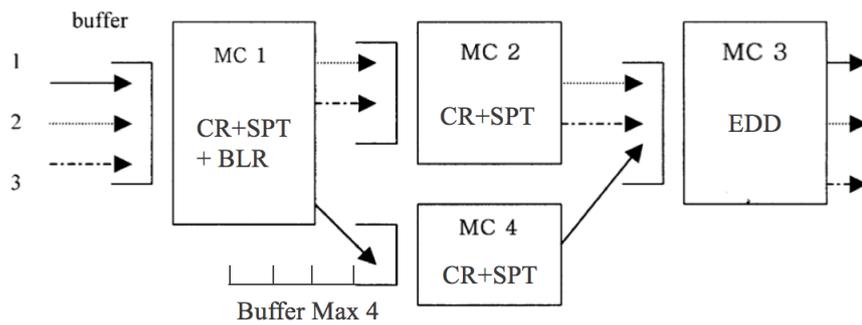


Figura 4.39: Esquema representativo do sequenciamento Parallel Machines Cr + SPT + BLR

Capítulo 5

Discussão de Resultados

Com a metodologia descrita e a dinâmica do sistema apresentado, a abordagem foi avaliada em termos de qualidade das soluções propostas e desempenho computacional, considerando o impacto das alterações da discretização do tempo no modelo matemático e as diferentes regras de sequenciamento.

5.1 Impacto das diferentes Regras de Sequenciamento

Quantidades de pedidos de produção e regras de sequenciamento foram projetadas de forma a provocar o máximo desempenho do sistema, forçando-o a trabalhar perto do seu limite. Para além da redução de custos, uma das observações feitas quando ocorria a execução do modelo de simulação era a importância do *idle time* correspondente ao *bottleneck*, valores dos tempos de processamento e rotas definidas. Uma vez que os tempos de processamento e rotas eram inalteráveis, era necessário definir qual a ordem de produção das estações de trabalho e o seu tempo de espera.

Identificadas algumas das várias regras de sequenciamento possíveis de implementar de acordo com a redução do *idle time* do *bottleneck* e máximo de estações de trabalho a produzir é necessário analisar o seu impacto no sistema de produção definido.

Para comparar o impacto de cada regra foram utilizados quatro indicadores de desempenho:

- Tempo total de execução
- Percentagem de produtos não entregues
- *Idle time* total
- *Idle time* do *bottleneck*
- Custo Total

Na tabela 5.1 estão representadas as diferentes regras abordadas para o ciclo de execução ótimo, como pode ser observado no anexo A. É de referir que o modelo de abordagem utilizado para esta análise é o Modelo Base.

Tabela 5.1: Desempenho das diferentes regras de sequenciamento para o ciclo de execução ótimo

Regra de Sequenciamento	Tempo total de Execução (horas)	% Produtos não Entregues	Total <i>idle time</i> (horas)	<i>Idle time Bottleneck</i> (%)	Produção 1 ^o iteração
FIFO	138,92	24,07	134,89	46,31	874
CR + SPT	132,65	23,89	129,83	1,13	876
BLR	133,8	16,07	134,77	2	966
Parallel Machines CR + SPT + BLR	131,4	22,15	125,23	0,25	896

Dos resultados obtidos é possível afirmar que existem regras com uma maior preferência de implementação. A regra de sequenciamento FIFO apresenta-se como a pior regra a implementar no sistema, registando uma baixa produção de produtos e uma percentagem de *idle time* do *bottleneck* quase de 50 %, o que indica que metade dos tempos de atraso correspondiam à estação de trabalho que deveria estar no seu limite de produção.

De todas as regras apresentadas, as de melhor implementação são a BLR e a Parallel Machines CR + SPT + BLR, que não só apresentam os dois melhores valores de produtos produzidos em relação a todas as regras estudadas, como apresentam uma boa percentagem de *idle time* do *bottleneck*, forçando o sistema a trabalhar no seu limite.

Na tabela 5.2 são analisadas estas duas regras de sequenciamento, onde é possível concluir que existe um maior custo por produção na regra BLR. Quanto à relação de custos e produção, a passagem da regra BLR para Parallel Machines CR + SPR + BLR apresenta uma redução de custos de 4,4 % e uma redução de produção de 1,2 % para a mesma abordagem, isto é, um planeamento de produção com produtos e máquinas idênticas.

Tabela 5.2: Relação entre as regras BLR e Parallel Machines CR + SPT + BLR

Regra de Sequenciamento	Custo \$	Produção	Custo/Produção
BLR	316055	1036	305,07
Parallel Machines CR + SPT + BLR	301825	1024	295,75

Já que as regras são quase idênticas para a escolha de qual a melhor regra a implementar, o utilizador deverá ter em conta as seguintes questões:

- É preferível obter maior número de produtos entregues?
- É preferível obter menores custos de produção?

Na análise deste projeto foi implementada a regra Parallel Machines CR + SPT + BLR, já que foi adotado uma ideologia de menor custos de produção para um bom e razoável valor de produtos produzidos.

5.2 Comparar Abordagens

Expostas e analisadas as regras de sequenciamento é necessário analisar a resposta dos dois diferentes modelos de interação entre o método de otimização e simulação (secção 4.5).

De forma a analisar os resultados obtidos pelos dois modelos, estes foram comparados aos resultados apresentados por *Byrne e Bakir* [1] e *Kim e Kim* [11], que nas suas implementações matemáticas utilizavam como restrição de melhoramento de iterações o ajuste da capacidade disponível nas estações de trabalho, correspondente ao lado direito da restrição de capacidade da equação 3.2, tendo *Kim e Kim* [11] adicionado dois novos parâmetro a essa mesma equação – as taxas de fluxo de carregamento de cada estação de trabalho são estimados iterativamente em relação a qualquer período de produção, em vez de obter taxas em períodos de tempo fixo.

Os resultados apresentados como comparação não só usam uma restrição diferente da abordada neste projeto, como também usam uma regra de sequenciamento diferente da implementada no modelo de simulação (tabela 5.3), contudo foram considerados como bons indicadores de comparação com as soluções obtidas neste projeto já que o objetivo era melhorar e encontrar um novo tipo de resolução para este género de problemas.

A janela temporal para analisar o comportamento dos modelos foi fixada segundo *Byrne e Bakir* [1], que correspondia a três semanas de trabalho com um período de quarenta horas em cada semana.

Conforme indicado na secção 3.2.3.1, os modelos desenvolvidos e defendidos neste projeto seguem uma ideologia em que a ordem de produção indicada pelo método de otimização deve ir de encontro com a ordem de produção indicada pelo método de simulação.

Nos anexos B estão representados os resultados das quatro abordagens comparadas, tendo como principal análise o número de iterações necessárias para atingir a melhor solução, o custo que essa solução implica e as taxas de produção ótimas resultantes de cada iteração.

Tabela 5.3: Comparação dos diferentes tipos de Abordagem

Modelo	Regra de Sequenciamento	Iterações	Output	Custos (\$ 000)
<i>Byrne e Bakir</i>	FIFO	7	660	350
<i>Kim e Kim</i>	FIFO	3	1126	241
Modelo Base	Parallel Machines CR + SPT + BLR	4	1018	302,8
Modelo Base Modificado	Parallel Machines CR + SPT + BLR	6	1024	301,825

Comparando as duas últimas abordagens é possível afirmar que dos modelos desenvolvidos nesta dissertação, o que apresenta melhores resultados é o Modelo Base Modificado, apesar de conter um tempo de convergência para o melhor resultado maior do que o Modelo Base devido ao número de iterações necessárias para alcançar o critério de paragem. Além dos menores custos de produção revelados, tem maior número de quantidade de produtos produzidos. A diferença do número de iterações pode parecer muito significativo, mas o valor do custo total e número de produtos produzidos é significativamente alto comparando as alterações que existem entre os dois modelos.

Da análise da tabela 5.2 comprovamos que os métodos desenvolvidos apresentam melhores valores em relação a *Byrne* e *Bakir* [1] e piores valores em relação a *Kim* e *Kim* [11]. Esta diferença de resultados era de esperar visto que ambos os artigos não apresentam a implementação do modelo no método de simulação. Deste modo, processos, prioridades de sequenciamento e limitadores de armazenamento podem ter sido implementados com dados não mencionados, possibilitando melhores soluções.

Relacionando os modelos desenvolvidos constata-se que os valores são significativamente afetados pelas restrições de entrada e pela configuração do sistema. Não só as restrições contribuem para um melhor resultado, mas também a implementação de fatores como as regras de sequenciamento e limitadores de *buffers*.

5.3 Impacto das diferentes durações dos Períodos

A identificação das várias janelas temporais e variação da duração dos períodos foi introduzida e analisada brevemente na secção 4.2.

A duração do período está associada ao número de períodos por intervalo de tempo, o que implica que uma quantidade maior de períodos por intervalo de tempo se traduza em períodos com valores de tempo mais pequenos e um aumento de precisão do modelo. No entanto, o aumento da quantidade de períodos diminuí o desempenho do modelo matemático, já que ocorre um aumento do número de variáveis. Esta vantagem e desvantagem relacionam-se num equilíbrio entre a precisão e o desempenho que o modelo deve alcançar.

Nesta secção é apresentada uma análise detalhada desta estratégia de solução.

Para além da conversão dos custos de produção, inventário e *backorder*, a discretização temporal requer a conversão dos prazos de entrega de cada unidade para os períodos de tempo. A conversão de cada período é feita de modo a que o tempo de execução seja igual ao quociente entre o valor do tempo definido na janela temporal pré-definida e o número de períodos aplicados à instância em estudo.

Para comparar o impacto da alteração do mesmo cenário com diferentes janelas temporais (diferem-se na duração de cada período), foram utilizados quatro indicadores de desempenho:

- Custo final

- Quantidade de produtos entregues
- Tempo total do plano de produção
- Número de iterações necessárias para atingir o critério de paragem

Tabela 5.4: Desempenho das diferentes janelas temporais

Períodos	Tempo do período (horas)	Modelo	Regra de Sequenciaamento	Custo (\$ 000)	Output	Tempo de Produção	Iterações
3	40	Modelo Base	Parallel Machines CR + SPT + BLR	302,8	1018	119,15	4
6	20	Modelo Base	Parallel Machines CR + SPT + BLR	278,8	1024	119,99	7
15	8	Modelo Base	Parallel Machines CR + SPT + BLR	293,4	1032	120,88	15
3	40	Modelo Base Modificado	Parallel Machines CR + SPT + BLR	301,82	1024	119,90	6
6	20	Modelo Base Modificado	Parallel Machines CR + SPT + BLR	280,37	1026	120,55	9
15	8	Modelo Base Modificado	Parallel Machines CR + SPT + BLR	292	1038	121,58	18

Como esperado, a qualidade de solução aumenta com a diminuição da duração de cada período, e consequentemente aumenta o número de iterações necessárias para chegar ao valor ótimo. O custo da solução diminui e a percentagem de produtos entregues aumenta. No entanto, tanto o tempo de execução do modelo matemático como do modelo de simulação aumenta com a duração dos períodos, já que existem mais períodos para o mesmo intervalo de tempo.

Na tabela 5.4 estão representados os dois modelos de abordagem desenvolvidos neste projeto, Modelo Base e Modelo Base Modificado, comprovando que o segundo apresenta melhores resultados para diferentes janelas temporais.

Na discretização do modelo de 40 horas para 20 horas e 8 horas o número de variáveis de decisão aumentam uma vez que o número de períodos aumenta e o modelo aumenta de tamanho e complexidade, observado no anexo C.

Analisando os resultados da melhor abordagem, para o cenário de 40 horas a solução final foi encontrada ao final de 6 iterações, o que traduz uma precisão mais baixa. O tempo de produção total de cada produto evita que a quantidade de produtos pedidos seja cumprida.

O melhor cenário estudado foi o de duração de 8 horas cada período. Apesar de apresentar o estudo com maior tempo de execução, fator que pode não ser prático em sistemas com situações reais e de maior complexidade, apresenta os melhores valores de produção e custos.

Ao longo da discretização do modelo, o tempo de processamento para encontrar a solução ideal tende a aumentar. O número de variáveis aumenta com o número de períodos, influenciando os tempos de execução do programa.

Na tabela 5.5 estão representados os valores das quantidades de variáveis inteiras correspondentes a cada período de tempo.

Tabela 5.5: Relação entre a quantidade de variáveis inteiras e a quantidade de períodos

Número de Períodos	Variáveis Inteiras
3	99
6	186
15	447

Capítulo 6

Conclusão

Esta dissertação centrou-se no desenvolvimento de uma metodologia de ajuda para sistemas industriais de alto nível de desempenho e tarefas, que tenham de tomar decisões e lidar com problemas de planeamento de produção. O objetivo alcançado propõe um método híbrido iterativo entre os métodos de otimização e simulação.

A abordagem híbrida foi examinada para um caso de estudo genérico apresentando bons resultados. O procedimento híbrido iterativo permitiu que as discrepâncias entre o método analítico e o método de simulação fossem atenuadas e uma solução ótima realisticamente viável fosse encontrada.

A proposta de estudo desta dissertação baseia-se em um modelo MIP simples e um modelo de simulação detalhado do sistema real, apesar deste ser um caso de estudo genérico. O primeiro é usado para obter planos de produção ideais para a função objetiva definida, enquanto que no segundo método a flexibilidade de modelagem da simulação permite que as características de operação probabilísticas e não-lineares sejam facilmente consideradas. O modelo de simulação foi capaz de englobar características do sistema de forma eficaz, onde a aplicação de várias regras de sequenciamento e limitadores de armazenamento revelaram-se boas abordagens.

Nesta dissertação a interação entre os dois métodos ocorre até que seja encontrado o valor critério paragem. A metodologia proposta mostrou-se bastante flexível quanto à discretização temporal do modelo, permitindo assim aplicar o método desenvolvido em sistemas de produção relaxados ou de alta eficiência.

O estudo efetuado foi projetado de forma a provocar o máximo desempenho do sistema, forçando-o a trabalhar no seu limite tendo todas as estações de trabalho com o menor tempo de inatividade possível, principalmente o *bottleneck* do sistema, que foi um dos fatores de maior importância de estudo neste projeto. Embora o método proposto tenha sido desenvolvido para um sistema com taxas de produção relativamente alta devido aos valores dos pedidos de produção e capacidades das estações de trabalho, a metodologia pode ser usada em sistemas com taxas de produção mais baixas e com menor diferenciação entre produtos. Nestas condições a metodologia defendida foi capaz de apresentar planos de produção detalhados e satisfatórios, contudo é de reconhecer que a eficácia deste método em outros sistemas pode não ser tão satisfatório.

Esta abordagem híbrida otimização-simulação força a convergência dos valores dos dois métodos, contudo ao contrário de outras abordagens híbridas esta não reduz a qualidade das soluções. O impacto da simulação na otimização afeta as quantidades de produtos que devem ser lançados na linha de produção a cada período, tendo posteriores consequências nos tempos de espera e taxas de utilização de cada estação de trabalho dependendo das quantidades de produtos a produzir.

O próximo passo seria a abordagem dos modelos desenvolvidos num caso de estudo mais complexo de forma a refinar a metodologia presente nesta dissertação.

6.1 Trabalhos Futuros

Apesar dos resultados satisfatórios produzidos pela metodologia defendida, apenas uma abordagem ao caso de estudo e análises em diferentes janelas temporais foi realizado.

Se o sistema não for determinista, ou seja, as estações de trabalho forem sujeitas a falhas, fatores adicionais devem ser contabilizados. Testar e avaliar os efeitos de todos esses fatores pode gerar um plano de produção verdadeiramente viável e com soluções mais realistas.

Recentemente as empresas reconhecem que o recurso a Data Mining é uma técnica importante com um forte impacto no desempenho de tarefas a realizar dentro desta. A organização e estudos de dados é cada vez mais um fator importante na evolução de processos. Quanto ao problema de planeamento de produção, muitas tarefas abordam pesquisas e dados passados de forma a contribuir a uma melhor análise de qual o produto a lançar com prioridade nas linhas. Um estudo aprofundado nesta área seria vantajoso, visto que hoje em dia ainda existe um grande problema na interação entre Data Mining e planeamento de produção devido à incerteza, pedidos de produção e sistemas operacionais variáveis. Assim, uma forte aposta em algoritmos de Data Mining em abordagens híbridas poderia contribuir para novas e ótimas soluções de produção.

Outra opção, que se caracteriza como mais futurista e inovadora, é o uso de algoritmos preditivos para planear uma linha de produção ou estações de trabalho de forma mais adaptativa em vez de um cronograma fixo. A tecnologia de produção moderna tem vindo a incorporar Machine Learning ao longo dos processos de produção. Analisando o desempenho anterior do sistema com recurso a Machine Learning é garantido que existe um conhecimento prévio de qual os produtos que devem ser produzidos a cada momento, obtendo os melhores resultados possíveis.

Anexo A

A.1 Impacto das diferentes Regras de Sequenciamento

A.1.1 Gráfico de Desempenho

Comparação das diferentes Regras de Despacho
(3weeks)3 periods - 2400min

Rule	TotalTime (hours)	% Delays	IdleTime (hours)		Products				Production						
			Total	Machine	Period	ID	Start	Out							
FIFO Same as EDD	138,92	24,066	154,891	Server1 19,1352 Server2 61,6019 Server3 71,7352 Server4 66,4186	0	O1	150	150	874						
						O2	111	111							
						O3	124	55							
					1	O1	125	125							
						O2	127	127							
						O3	126	32							
					2	O1	160	160							
						O2	105	105							
						O3	122	9							
					CR+SPT	132,65	23,8923	129,826		Server1 12,869 Server2 55,3356 Server3 1,469 Server4 60,1423	0	O1	150	130	876
												O2	111	78	
												O3	124	124	
1	O1	125	105												
	O2	127	59												
	O3	126	126												
2	O1	160	101												
	O2	105	30												
	O3	123	123												
Parallel Machines FIFO	133,8	16,07	Server1 14,1029 Server2 56,5696 Server3 2,7029 Server4 61,3863	0					O1		150	150	966		
									O2		111	111			
									O3		124	86			
				1	O1	125	125								
					O2	127	127								
					O3	126	63								
				2	O1	160	160								
					O2	105	105								
					O3	123	39								
				Parallel Machines CR + SPT	131,4	22,15	Server1 11,7182 Server2 54,1849 Server3 0,3182 Server4 59,0015	0	O1	150	129	896			
									O2	111	89				
									O3	124	124				
1	O1	125	105												
	O2	127	69												
	O3	126	126												
2	O1	160	79												
	O2	105	52												
	O3	123	123												

Figura A.1: Desempenho das diferentes regras de sequenciamento para o primeiro ciclo de execução

Anexo B

B.1 Comparação dos Modelos Abordados

B.1.1 *Byrne e Bakir* [1]

Tabela B.1: Resultados obtidos pela abordagem de *Byrne e Bakir* [1]

Período	Produto	Iteração						
		1	2	3	4	5	6	7
1	1	150	150	150	150	150	150	150
	2	111	40	21	14	13	12	12
	3	124	76	64	59	58	57	57
2	1	125	125	125	125	125	125	125
	2	127	60	44	38	35	35	35
	3	126	82	71	67	65	65	65
3	1	160	160	160	160	160	160	160
	2	105	31	12	5	3	2	2
	3	123	74	61	57	55	54	54
<i>Output</i> Total		1151	798	708	675	664	660	660
Custo Total (\$000)		188	304	335	345	348	350	350

B.1.2 *Kim e Kim* [11]

Tabela B.2: Resultados obtidos pela abordagem de *Kim e Kim* [11]

Período	Produto	Iteração		
		1	2	3
1	1	150	136	131
	2	111	121	121
	3	124	118	118
2	1	125	129	129
	2	127	125	125
	3	126	124	124
3	1	160	144	137
	2	105	130	130
	3	123	111	111
<i>Output Total</i>		1151	1138	1126
Custo Total (\$000)		213,068	232,526	240,677

B.1.3 Modelo Base

Tabela B.3: Resultados obtidos pela abordagem Modelo Base

Período	Produto	Iteração			
		1	2	3	4
1	1	150	129	129	129
	2	111	89	89	89
	3	124	124	124	124
2	1	125	129	128	128
	2	127	99	89	89
	3	126	124	124	124
3	1	160	106	127	127
	2	105	71	86	86
	3	123	123	122	122
<i>Output Total</i>		1151	994	1018	1018
Custo Total (\$000)		188,79	255,07	277,2	302,8

B.1.4 Modelo Base

Tabela B.4: Resultados obtidos pela abordagem Modelo Base Modificado

Período	Produto	Iteração					
		1	2	3	4	5	6
1	1	150	129	129	129	129	129
	2	111	89	94	89	89	89
	3	124	124	124	124	124	124
2	1	125	118	125	123	124	124
	2	127	79	87	86	88	88
	3	126	126	126	126	126	126
3	1	160	93	121	126	134	134
	2	105	62	84	85	87	87
	3	123	123	123	123	123	123
<i>Output Total</i>		1151	943	1013	1011	1024	1024
Custo Total (\$000)		188,79	232,7	263,9	281,175	293,83	301,82

Anexo C

C.1 Resultados das várias janelas temporais

C.1.1 Modelo Base – 3 Períodos

		Normal -A cada período é definido o que produzir sem erro.						DispatchingRule -> Parallel Machine CR + SPT											
Period	Product	Iteration Number																	
		1		2		3		4		1		2		3		4			
		Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out		
1	1	1	150	129	129	129	129	129	129	129	129	129	129	129	129	129	129	129	
		2	111	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
		3	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124
	2	1	125	105	144	129	129	128	128	128	128	128	128	128	128	128	128	128	128
		2	127	69	116	99	99	89	89	89	89	89	89	89	89	89	89	89	89
		3	126	126	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124
	3	1	160	79	162	106	177	127	127	127	127	127	127	127	127	127	127	127	127
		2	105	52	104	71	94	86	86	86	86	86	86	86	86	86	86	86	86
		3	123	123	123	123	122	122	122	122	122	122	122	122	122	122	122	122	122
Total Output		1151	896	1115	994	1087	1018	1018	1018	1018	1018	1018	1018	1018	1018	1018	1018	1018	
Total Cost		187925		255075		277200		302800											
%Delay		23,89		10,85		6,35		0											

Figura C.1: Resultados obtidos pela abordagem Modelo Base – 3 Períodos

C.1.2 Modelo Base Modificado – 3 Períodos

Period	Product	Iteration Number						DispatchingRule -> Parallel Machine CR + SPT						
		1		2		3			4		5		6	
		Start	Out	Start	Out	Start	Out		Start	Out	Start	Out	Start	Out
1	1	150	129	140	129	135	129	132	129	130	129	129	129	
	2	111	89	99	89	95	94	94	89	92	89	89	89	
	3	124	124	124	124	124	124	124	124	124	124	124	124	
2	1	125	105	130	118	125	125	125	123	124	124	124	124	
	2	127	69	124	79	100	87	95	86	92	88	88	88	
	3	126	126	126	126	126	126	126	126	126	126	126	126	
3	1	160	79	165	93	175	121	150	126	135	134	134	134	
	2	105	52	102	62	94	84	89	85	87	87	87	87	
	3	123	126	123	123	123	123	123	123	123	123	123	123	
Total Output		1151	899	1133	943	1097	1013	1058	1011	1033	1024	1024	1024	
Total Cost		187925		232725		263925		281175		293825		301825		
%Delay		23,89		16,77		7,66		4,44		0,87		0		

Figura C.2: Resultados obtidos pela abordagem Modelo Base Modificado – 3 Períodos

C.1.4 Modelo Base Modificado – 6 Períodos

Period	Product	DispatchingRule -> Parallel Machine CR+SPT																	
		Iteration Number						1						2					
		Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out		
1	1	80	65	72	65	68	65	67	64	65	64	64	64	64	64	64	64	64	
	2	52	42	47	42	45	42	43	43	43	43	43	43	43	43	43	43	43	
	3	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	
2	1	70	52	80	62	72	64	68	65	67	64	65	64	64	64	64	64	64	
	2	58	34	52	39	45	42	44	42	43	43	43	43	43	43	43	43	43	
	3	63	63	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	
3	1	62	39	58	49	72	62	68	65	66	64	65	64	64	64	64	64	64	
	2	64	24	67	30	58	40	45	42	44	43	44	43	43	43	43	43	43	
	3	63	63	63	63	62	62	62	62	62	62	62	62	62	62	62	62	62	
4	1	62	26	65	35	62	49	72	64	68	65	66	64	65	64	64	64	64	
	2	64	14	62	19	64	30	58	42	51	42	42	43	44	43	43	43	43	
	3	63	63	63	63	63	63	62	62	62	62	62	62	62	62	62	62	62	
5	1	80	14	80	21	80	35	80	51	86	62	74	65	70	67	68	67	67	
	2	52	4	52	11	52	21	52	33	49	39	44	42	43	43	43	43	43	
	3	62	62	62	62	62	62	62	62	61	61	61	61	61	61	61	61	61	
6	1	81	2	80	9	81	23	80	41	83	48	101	70	85	71	76	73	73	
	2	51	0	52	3	51	13	52	24	51	33	46	46	46	46	46	46	46	
	3	62	59	62	62	62	62	62	62	61	61	60	60	60	60	60	60	60	
Total Output		1151	688	1141	759	1123	859	1101	948	1086	980	1071	1020	1043	1024	1030	1026	1026	
Total Cost		200555	209900	209900	228525	228525	247950	247950	259225	259225	270250	270250	277550	277550	279775	279775	280375	280375	
%Delay		40,22589053	33,48	33,48	23,51	23,51	13,90	13,90	9,76	9,76	4,76	4,76	1,82	1,82	0,39	0,39	0	0	

Figura C.4: Resultados obtidos pela abordagem Modelo Base Modificado–6 Períodos

C.1.6 Modelo Base Modificado– 15 Períodos

Some Errors - A cada período é calculado o erro productivo com erro

Period	Product		1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		16		17		18			
	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out	Start	Out				
1	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
2	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
3	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
4	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
5	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
6	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
7	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
8	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
9	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
10	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
11	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
12	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
13	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
14	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15	30	15
Total Output	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764	1528	764
Total Cost	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5	7637	3818.5

Figura C.6: Resultados obtidos pela abordagem Modelo Base Modificado– 6 Períodos

Referências

- [1] MD Byrne e M Akif Bakir. Production planning using a hybrid simulation–analytical approach. *International Journal of Production Economics*, 59(1):305–311, 1999.
- [2] Gonçalo Figueira e Bernardo Almada-Lobo. Hybrid simulation–optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46:118–134, 2014.
- [3] Ludo F Gelders e Luk N Van Wassenhove. Production planning: a review. *European Journal of Operational Research*, 7(2):101–110, 1981.
- [4] Angel A Juan, Javier Faulin, Scott E Grasman, Markus Rabe, e Gonçalo Figueira. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2:62–72, 2015.
- [5] Farhad Azadivar. Simulation optimization methodologies. Em *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1*, páginas 93–100. ACM, 1999.
- [6] Hubert Missbauer e Reha Uzsoy. Optimization models of production planning problems. Em *Planning Production and Inventories in the Extended Enterprise*, páginas 437–507. Springer, 2011.
- [7] Yves Pochet e Laurence A Wolsey. *Production planning by mixed integer programming*. Springer Science & Business Media, 2006.
- [8] Julia Pahl, Stefan Voß, e David L Woodruff. Production planning with load dependent lead times: an update of research. *Annals of Operations Research*, 153(1):297–345, 2007.
- [9] Yi-Feng Hung e Robert C Leachman. A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *IEEE Transactions on Semiconductor manufacturing*, 9(2):257–269, 1996.
- [10] Yi-Feng Hung e Meng-Chi Hou. A production planning approach based on iterations of linear programming optimization and flow time prediction. *Journal of the Chinese Institute of Industrial Engineers*, 18(3):55–67, 2001.
- [11] Bokang Kim e Sooyoung Kim. Extended model for a hybrid production planning approach. *International Journal of Production Economics*, 73(2):165–173, 2001.
- [12] D Fatih Irdem, Necip Baris Kacar, e Reha Uzsoy. An exploratory analysis of two iterative linear programming—simulation approaches for production planning. *IEEE Transactions on Semiconductor Manufacturing*, 23(3):442–455, 2010.

- [13] MD Byrne e MM Hossain. Production planning: An improved hybrid approach. *International Journal of Production Economics*, 93:225–229, 2005.
- [14] Catarina M Marques, Samuel Moniz, Jorge Pinho de Sousa, e Ana Paula Barbosa-Póvoa. A simulation-optimization approach to integrate process design and planning decisions under technical and market uncertainties: A case from the chemical-pharmaceutical industry. *Computers & Chemical Engineering*, 2017.
- [15] June-Young Bang e Yeong-Dae Kim. Hierarchical production planning for semiconductor wafer fabrication based on linear programming and discrete-event simulation. *IEEE Transactions on Automation Science and Engineering*, 7(2):326–336, 2010.
- [16] Iraj Mahdavi, Babak Shirazi, e Maghsud Solimanpur. Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems. *Simulation Modelling Practice and Theory*, 18(6):768–786, 2010.
- [17] Jeffrey B. Sidney. Sequencing and scheduling-an introduction to the mathematics of the job-shop, by simon french, wiley, 1982, 245. *Networks*, (2):310–311.
- [18] Joseph Adams, Egon Balas, e Daniel Zawack. The shifting bottleneck procedure for job shop scheduling. *Management science*, 34(3):391–401, 1988.
- [19] Eugeniusz Nowicki e Czesław Smutnicki. An advanced tabu search algorithm for the job shop problem. *Journal of Scheduling*, 8(2):145–159, 2005.
- [20] Tung-Kuan Liu, Jinn-Tsong Tsai, e Jyh-Horng Chou. Improved genetic algorithm for the job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 27(9-10):1021–1029, 2006.
- [21] Liji Shen e Udo Buscher. Solving the serial batching problem in job shop manufacturing systems. *European Journal of Operational Research*, 221(1):14–26, 2012.
- [22] Hamid Davoudpour e Maryam Ashrafi. Solving multi-objective sdst flexible flow shop using grasp algorithm. *The International Journal of Advanced Manufacturing Technology*, 44(7):737–747, 2009.
- [23] Suk Jae Jeong, Seok Jin Lim, e Kyung Sup Kim. Hybrid approach to production scheduling using genetic algorithm and simulation. *The International Journal of Advanced Manufacturing Technology*, 28(1):129–136, 2006.
- [24] Hong Zhou, Waiman Cheung, e Lawrence C Leung. Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm. *European Journal of Operational Research*, 194(3):637–649, 2009.
- [25] Jingang Liu, Chihui Li, Feng Yang, Hong Wan, e Reha Uzsoy. Production planning for semiconductor manufacturing via simulation optimization. Em *Simulation Conference (WSC), Proceedings of the 2011 Winter*, páginas 3612–3622. IEEE, 2011.
- [26] Veronique Sels, Kjeld Craeymeersch, e Mario Vanhoucke. A hybrid single and dual population search procedure for the job shop scheduling problem. *European Journal of Operational Research*, 215(3):512–523, 2011.
- [27] H Missbauer. Aggregate order release planning for time-varying demand. *International Journal of Production Research*, 40(3):699–718, 2002.

- [28] US Karmarkar. Capacity loading and release planning with work-in-progress (wip) and leadtimes. *Journal of Manufacturing and Operations Management*, 2(105-123), 1989.
- [29] Stephen C Graves. A tactical planning model for a job shop. *Operations Research*, 34(4):522–533, 1986.
- [30] Necip Baris Kacar, Lars Monch, e Reha Uzsoy. Planning wafer starts using nonlinear clearing functions: A large-scale experiment. *IEEE Transactions on Semiconductor Manufacturing*, 26(4):602–612, 2013.
- [31] Peter Pürgstaller e Hubert Missbauer. Rule-based vs. optimisation-based order release in workload control: A simulation study of a mto manufacturer. *International Journal of Production Economics*, 140(2):670–680, 2012.
- [32] Necip Baris Kacar, Durmus Fatih Irdem, e Reha Uzsoy. An experimental comparison of production planning using clearing functions and iterative linear programming-simulation algorithms. *IEEE Transactions on Semiconductor Manufacturing*, 25(1):104–117, 2012.
- [33] Ashkan Negahban e Jeffrey S Smith. Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, 33(2):241–261, 2014.
- [34] Jerry Banks, JOHN S CARSON II, L Barry, et al. *Discrete-event system simulation fourth edition*. Pearson, 2005.
- [35] C Dennis Pegden. Simio: a new simulation system based on intelligent objects. Em *Simulation Conference, 2007 Winter*, páginas 2293–2300. IEEE, 2007.
- [36] C Dennis Pegden. Introduction to simio. Em *Simulation Conference, 2008. WSC 2008. Winter*, páginas 229–235. IEEE, 2008.
- [37] Anylogic. <http://anylogic.com/system-dynamics>. Cited June 2017.
- [38] Anylogic. <http://anylogic.com/discrete-event-simulation>. Cited June 2017.
- [39] Anylogic. <http://anylogic.com/agent-based-modeling>. Cited June 2017.
- [40] Anylogic. <http://www.anylogic.com/features>. Cited June 2017.
- [41] Software Rockwell. Arena basic user’s guide. Em *Simulation Conference, 2005 Winter*, 2005.
- [42] F Robert Jacobs, Richard B Chase, e N Aquilano. Operations management for competitive advantage. *Boston: Mc-Graw Hill*, 64:70, 2004.
- [43] Nigel Slack, Stuart Chambers, e Robert Johnston. *Operations management*. Pearson education, 2010.
- [44] Steven Nahmias e Ye Cheng. *Production and operations analysis*, volume 2. Irwin Homewood, IL, 1993.
- [45] GUROBI. <http://www.gurobi.com/products/gurobi-optimizer>. Cited June 2017.

- [46] EXCEL LIBRARY C. [https://msdn.microsoft.com/en-us/library/ms173186\(v=vs.80\).aspx](https://msdn.microsoft.com/en-us/library/ms173186(v=vs.80).aspx). Cited June 2017.