



UNIVERSIDADE D
COIMBRA

Marta Isabel Perienes Franco

**OTIMIZAÇÃO E SIMULAÇÃO EM
SISTEMAS DE PRODUÇÃO E LOGÍSTICA**

Com aplicações em Matlab

**Dissertação no âmbito de Simulação e Logística
orientada pelo Professor Doutor Marco Seabra Reis e
co-orientada pelo Professor Doutor Samuel Moniz
e apresentada ao Departamento de Engenharia Química
para obtenção do grau de Mestre em Engenharia Química**

Outubro de 2020

Agradecimentos

Este trabalho foi um enorme desafio, numa fase delicada da minha vida. Foi com perseverança que o concluí, apesar das adversidades que, por vezes, nos levam a pensar desistir.

E o apoio de algumas pessoas foi essencial para chegar ao fim desta caminhada.

É tempo de agradecer!

Aos meus orientadores, Professor Doutor Marco Seabra e Professor Doutor Samuel Moniz, com quem aprendi muito com a sua sabedoria e rigor científico, a minha gratidão pela Vossa preciosa contribuição para as ideias apresentadas neste trabalho.

À Dulce, pela sua sempre pronta disponibilidade em ajudar e pelas suas palavras de constante incentivo e de amizade.

Aos amigos, que incansavelmente me repetiram “Força”, e por acreditarem sempre que eu seria capaz, mesmo quando eu mesma não o sabia.

Aos meus pais e à minha irmã, por todo o apoio incondicional que me deram ao longo de toda a minha vida.

Às minhas filhas, pela compreensão e paciência.

Sou grata!

Resumo

Na indústria, o planeamento da produção é uma área decisiva, permitindo reduzir custos operacionais e melhorar o desempenho. No entanto, a implementação de novos sistemas e/ou estratégias requer uma análise cuidada dos riscos inerentes. A utilização da simulação computacional tem um papel preponderante nesta análise permitindo estudar cenários e tomar decisões mais informadas.

Neste trabalho, explora-se a combinação de modelos de simulação e otimização para obter melhores soluções para problemas de planeamento e de projeto. Em particular, esta dissertação pretende reforçar o papel da simulação computacional aplicada a sistemas de produção, como complemento analítico de metodologias clássicas de otimização. O modelo proposto é de um sistema de produção, com aplicação de uma abordagem mista “Simulação – Otimização”, aplicando um modelo de otimização e de simulação de eventos discretos. Na procura da solução ótima, o processo termina quando os critérios de paragem ou convergência são atingidos. As ilustrações são efetuadas recorrendo à plataforma *Matlab/Simulink*®. Os resultados obtidos comprovam a utilidade desta metodologia na análise de problemas de planeamento e de projeto.

Adicionalmente, a técnica de Monte Carlo foi utilizada neste trabalho, com o intuito de adequar o sistema à realidade, uma vez que ao introduzir a incerteza no sistema (através de *inputs* aleatórios), o modelo determinístico é transformado num modelo estocástico e consequentemente, é possível avaliar qual o impacto que esta incerteza tem, no desempenho do sistema. O modelo em estudo foi o projeto de um reator contínuo perfeitamente agitado. Desta forma, comprova-se como a simulação estocástica, associada à otimização, consegue fornecer resultados melhorados, na presença de incertezas em alguns parâmetros do sistema.

Concluimos que a simulação computacional, quando utilizada de forma apropriada, com dados fiáveis e uma correta modelação dos sistemas, pode dar uma preciosa ajuda na redução de custos e aumento da qualidade de produtos ou de serviços, melhorando a eficiência da organização, e fomentando o aumento da sua competitividade.

Abstracts

In industry, the planning of production is a key area, as it entails the reduction in operative costs and improvement of achievements. Although implementing new systems and/or resources, requires a detailed analysis of the inherit risks. As though, resources like computerized simulation have a major place in the analysis, allowing to study scenarios and make more reasoned decisions.

In this work it is studied the combination of simulation and optimization software to achieve the better scheduling solutions. In particular, this dissertation intends to reinforce the role of computational simulation applied to production systems, as an analytical complement of the classic optimization methodologies. The proposed model is a production system, using a dual approach of simulation-optimization outsources, with a discret event optimization and simulation model. Searching for the best solution, the process ends when the convergence or the stopping criteria are fulfilled. The illustrations were done using the Matlab/Simulink® platform. Results have proved the effectiveness of this tool in the analysis of scheduling processes.

The Monte Carlo technique was used with the aim of adequate the system to reality, as the introduction of uncertainty turns the deterministic model into a stochastic model. As a result, it is possible to evaluate the impact of uncertainty in the system performance. The model used was the CSTR project. As though it is proved that stochastic simulation associated to optimization achieves improved results, even facing the uncertainty of some system parameters.

When used appropriately, with the correct data and the right modelation, computerized simulation can be used to reduce costs and improve the quality of products or services, leading to a more competitive company.

Índice

1. Introdução.....	8
1.1 Contextualização do problema.....	8
1.2 Âmbito e objetivos.....	9
1.3 Estrutura da dissertação.....	11
2. Revisão bibliográfica.....	12
2.1. Conceito de sistema.....	12
2.2. Características das instalações descontínuas.....	13
2.3. <i>Scheduling</i>	19
2.4. Formulação dos problemas de planeamento.....	20
2.5. Métodos de resolução de problemas de planeamento.....	22
2.5.1. Métodos de otimização.....	23
2.5.2. Métodos de simulação.....	25
2.5.3. Métodos de simulação e otimização.....	26
3. Metodologia.....	30
3.1. Ferramentas e tecnologias de simulação.....	30
3.2. Modelação conceptual.....	31
3.3. <i>Software Matlab</i>	32
3.3.1. Simulink.....	33
3.3.2. Descrição dos blocos no <i>Simulink</i>	36
3.3.3. “ <i>Simulink design optimization</i> ”.....	39
4. Casos de estudo.....	40
4.1. Otimização de recursos partilhados num sistema produção <i>batch</i>	41
4.1.1. Implementação e tratamento de dados.....	45
4.1.2. Análise de resultados.....	48
4.2. Projeto de um CSTR (Bequette, 1998; <i>Mathworks.com</i>).....	57
4.2.1. Implementação e tratamento de dados.....	59
4.2.1. Análise de resultados.....	63
5. Conclusão.....	65
6. Referências bibliográficas.....	66
7. Anexo A – Caso de estudo 1: otimização de recursos partilhados.....	70
8. Anexo B – Caso de estudo 2: Projeto de um CSTR.....	73

Índice de Figuras

Figura 1 – Roteiro para problemas scheduling de processos descontínuos (Mendez, 2006) ...	16
Figura 2 – Roteiro para modelos otimização para scheduling curto prazo de processos descontínuos (Mendez, 2006).....	17
Figura 3 - Classificação dos Métodos de otimização global (adaptado)	23
Figura 4 - Ambiente de trabalho do Simulink	33
Figura 5 - Representação esquemática do sistema	41
Figura 6 – Processo de produção baseado no modelo do Simulink	41
Figura 7 - Subsistema: Ordens de Produção	42
Figura 8 - Subsistema: Plano de Produção	43
Figura 9 - Subsistema: Análise de dados	44
Figura 10 - Output do Algoritmo de otimização	46
Figura 11 - Metodologia Simulação Otimização.....	47
Figura 12 - Resultados da Simulação Inicial	49
Figura 13 - Cenário 1: Simulação após otimização.....	50
Figura 14 - Comparação das ordens de Produção A e B, antes e após otimização	51
Figura 15 - Cenário 2: incerteza na produção de A.....	52
Figura 16 - Cenário 2: Comparação da solução otimizada com a incerteza na produção de A	53
Figura 17 - Cenário 2: Comparação da situação otimizada com incerteza na produção de B	54
Figura 18 - Cenário 2: Comparação da situação otimizada com a incerteza nas produções de A e B.....	55
Figura 19 - Análise da influência das variáveis de decisão no número total de produções	56
Figura 20 - Projeto de um CSTR em ambiente Simulink	58
Figura 21 - Metodologia para projeto de um CSTR.....	59
Figura 22 - Correlação entre os parâmetros do projeto	60
Figura 23 - Gráfico de dispersão	61
Figura 24 - Gráfico de Tornado	61
Figura 25 - Metodologia para o problema de otimização.....	63
Figura 26 - Resultados da otimização.....	63

Índice de Tabelas

Tabela 1 - Valores dos parâmetros como estimativa inicial para otimização 62

Tabela 2 - Valores dos parâmetros obtidos após otimização..... 64

1. Introdução

Hoje em dia, a eficiência dos processos de tomada de decisão é um fator determinante para obter níveis elevados de rendimento e desempenho operacional dos sistemas de produção. Neste sentido, as empresas industriais têm a necessidade de sistematizar os processos de planeamento de produção, recorrendo para isso a métodos analíticos, como é o caso de ferramentas de otimização e/ou simulação. A aplicação destes métodos nos sistemas de produção é tanto mais complexa, quanto mais complexo é o sistema de produção. Uma vez que nem sempre os métodos analíticos conseguem fornecer uma solução completa e robusta para o problema, surge a necessidade de os complementar com modelos de simulação que permitam avaliar aspetos não explicitamente contemplados durante a otimização (Figueira, 2019). Os avanços computacionais têm permitido o desenvolvimento dos métodos de simulação, apresentando-se hoje como uma ferramenta de análise de sistemas e processos complexos (Frazzon et al., 2016; Grossmann et al., 2014, 2016). As duas técnicas, otimização e simulação, podem assim ser exploradas conjuntamente de forma a complementarem-se na construção de soluções mais efetivas para o planeamento de processos produtivos (Ammeri, 2011; Trigueiro de Sousa Junior et al., 2019; Velez & Maravelias, 2013). No Capítulo 2 serão feitas referências a estes métodos, bem como a vários estudos realizados, que comprovam a efetividade da sua combinação na área do planeamento da produção.

1.1 Contextualização do problema

No contexto económico mundial, a globalização, os avanços tecnológicos e a maior consciência ambiental têm levado a grandes mudanças na indústria em geral, e na Indústria Química em particular (Aqlan & Lam, 2016; Frazzon et al., 2016). Estas alterações podem ser de foro económico (oscilações nos preços de matérias primas e equipamentos e o seu reflexo no preço do produto final), ambiental (a legislação é cada vez mais exigente e as preocupações cada vez maiores com as práticas ambientais) ou técnico (o desenvolvimento científico e tecnológico reflete-se nos critérios de qualidade dos produtos). A dualidade preço/qualidade é um dos principais fatores para a competitividade das indústrias. Para aumentar as margens de lucro, é imprescindível reduzir os custos operacionais e melhorar os serviços ligados à distribuição. Um sistema eficaz de produção, ou seja, a capacidade de fornecer ao cliente os produtos certos, na quantidade encomendada, no tempo acordado, é fundamental para manter a

competitividade das empresas. E são as empresas mais competitivas, aquelas que também mais investem em ferramentas analíticas, com vista a auxiliar a gestão de topo na identificação de oportunidades, sua avaliação, comparação e tomada de decisão.

Na engenharia, uma das metodologias mais importantes para conduzir estas questões é a otimização, nomeadamente na procura das condições ótimas de funcionamento de sistemas (Papageorgiou, 2009). Uma vez que a capacidade dos computadores tem crescido exponencialmente, também a complexidade dos problemas que podem ser resolvidos por técnicas de otimização tem aumentado. Atualmente é possível resolver num computador pessoal um problema real de otimização com uma ou mais funções objetivo, restrições, (milhares de) variáveis contínuas ou discretas, etc. Mas, para isso, é preciso formular adequadamente o problema e recorrer às ferramentas de otimização mais adequadas ao problema em estudo, para assim chegar aos melhores resultados. Para alcançar esse objetivo, importa conhecer e classificar os sistemas mais comuns na Engenharia Química, para que de seguida se possa formular matematicamente o problema. Este processo implica uma série de decisões que passam pela escolha do modelo, definição da função objetivo, estimativa de parâmetros desconhecidos, análise de sensibilidade.

As técnicas de modelação e simulação são ferramentas alternativas na análise e tomada de decisões estratégicas nas operações e sistemas logísticos da indústria. Ao longo deste trabalho serão analisadas as diferentes abordagens de otimização e simulação, as suas características, como funcionam onde e quando devem ser aplicadas e quais os recursos necessários para as implementar.

Para contextualizar melhor o problema, serão descritos alguns exemplos práticos da aplicação conjunta das abordagens de simulação e otimização, evidenciando o seu elevado potencial como ferramentas de apoio à tomada de decisão quando aplicadas em conjunto.

1.2 Âmbito e objetivos

Neste trabalho pretende-se explorar as sinergias entre as abordagens de simulação e otimização para obtenção de soluções robustas e sua análise. Abordam-se duas metodologias, uma sequencial outra integrada, as quais serão ilustradas em dois sistemas simulados, com recurso ao *software Matlab/Simulink*. Num primeiro caso, aplica-se a técnica de otimização determinística, seguida da técnica de simulação, na melhoria do desempenho e racionalização de recursos envolvidos num sistema de produção com recursos partilhados. Num segundo caso, o projeto de um CSTR (*Continuous Stirred Tank Reactor*), aplicam-se as técnicas de simulação

e otimização de uma forma integrada, recorrendo à simulação de Monte Carlo, técnica muito útil nos modelos de natureza estocástica, uma vez que permite estudar a forma como um modelo se comporta, em resposta a *inputs* gerados aleatoriamente.

Deste modo, procurar-se-á analisar problemáticas associadas aos sistemas de produção através das perspetivas de otimização e simulação. Em suma, a formulação matemática dos problemas permite identificar e avaliar sistematicamente as várias alternativas possíveis. Havendo decisões a tomar, estas são representadas por variáveis de decisão, em que os valores dos elementos têm de ser determinados.

Em geral, a evolução dos modelos tem prosseguido ao longo de duas direções principais: abordagens determinísticas e abordagens não determinísticas ou estocásticas. Numa abordagem determinística, assume-se a hipótese de que todos os parâmetros são conhecidos com 100% de certeza, situação que raramente acontece na realidade. Numa abordagem não determinística, admite-se que pelo menos um dos parâmetros é incerto. Reconhecendo-se a existência de incerteza em relação ao conhecimento dos parâmetros, esta pode ser ignorada sem causar grande prejuízo à qualidade das soluções caso a resposta do modelo não seja muito sensível aos parâmetros incertos ou o grau de precisão dos parâmetros seja elevado. Porém, e principalmente para modelos de longo prazo (modelos em que a solução é válida por longos períodos de tempo), o resultado de uma abordagem determinística pode não ser realista e vir a trazer prejuízos (Aqlan & Lam, 2016; Grossmann et al., 2016). A não consideração da incerteza no planeamento de infraestruturas pode resultar em soluções que não respondem às necessidades num futuro imediato, em soluções que se tornam obsoletas a curto/médio prazo ou em soluções que acabam por se revelar sobredimensionadas. Assim, a melhor forma para lidar com a incerteza é, em primeiro lugar, aceitar que ela existe, depois estruturá-la e compreendê-la e, por fim, integrá-la nos processos de decisão (Kouvelis e Yu, 1996).

Depois de formalizado o problema real num modelo (isto é, estabelecida a função objetivo e as restrições aplicáveis), há que resolver o problema de otimização (isto é, determinar os valores das variáveis de decisão que minimizam ou maximizam a função objetivo satisfazendo as restrições). A determinação das soluções faz-se recorrendo a metodologias de otimização, as quais podem ser analisadas recorrendo a estudos de simulação, no sentido de a caracterizar de uma forma mais completa e determinar a sua robustez.

1.3 Estrutura da dissertação

O presente trabalho está estruturado da seguinte forma:

Capítulo 2 – Revisão Bibliográfica – Neste capítulo é apresentada uma revisão bibliográfica, sendo inicialmente introduzidos os principais conceitos necessários para a compreensão do trabalho. Introduzem-se as abordagens de otimização, simulação e a interação entre ambas.

Capítulo 3 – Metodologia – Neste capítulo são descritos os modelos e abordagens utilizados no desenvolvimento do trabalho, bem como a maneira como elas interagem entre si. Apresenta-se a metodologia utilizada na elaboração deste trabalho.

Capítulo 4 – Casos de estudo – São apresentados dois casos de estudo, analisando em detalhe as suas características e variáveis em estudo, recorrendo ao *software Simulink*. Neste capítulo são apresentados os resultados obtidos.

Capítulo 5 – Conclusões – Finalmente são apresentadas as conclusões obtidas com este trabalho, bem como indicações e sugestões de trabalhos futuros que contribuam para a continuação de pesquisas na área.

2. Revisão bibliográfica

Esta secção introduz os conceitos básicos necessários subjacentes a este trabalho. Para isso define-se sistema e classificam-se os seus tipos na indústria. Introduce-se o conceito de *scheduling* (escalonamento ou sequenciamento da produção) e abordam-se as diferentes estratégias para a sua formulação.

Uma vez formulado o problema de *scheduling*, analisam-se as várias metodologias para a sua resolução. De seguida introduzem-se os métodos de otimização e simulação, descrevendo as características de cada um, bem como as suas vantagens e inconvenientes. Finalmente, e com maior importância, faz-se uma revisão detalhada das técnicas híbridas de simulação e otimização, e a sua evolução ao longo dos últimos anos, onde têm tido uma importância crescente.

As pesquisas foram direcionadas a artigos científicos, maioritariamente publicados em revistas científicas internacionais e alguns livros da especialidade. O período de pesquisa incidu entre 1995-2019.

2.1. Conceito de sistema

Um sistema pode ser definido como um conjunto de elementos estruturais e não estruturais organizados e ligados entre si que processam os dados de entrada e produzem respostas, que constituem os dados de saída. Mays e Tung (1992) caracterizam os sistemas por uma fronteira (determina se um dado elemento pertence ou não ao sistema); pela relação dos estímulos e das respostas entre si, e com o meio envolvente. A identificação de um sistema facilita a construção de modelos matemáticos, que podem ser definidos como representações ideais de uma realidade expressa por símbolos e expressões matemáticas. Na indústria existem processos contínuos, descontínuos e semi-descontínuos. Este trabalho incide maioritariamente sobre instalações e processos descontínuos. A operação descontínua é escolhida preferencialmente quando o objetivo da produção é a obtenção de produtos de elevado valor acrescentado em reduzidos volumes de produção, com tecnologia altamente especializada e com tempos de residência nas unidades mais elevados.

2.2. Características das instalações descontínuas

Muitos processos químicos atuais incluem etapas onde o processamento é descontínuo ao longo do tempo. Este modelo de funcionamento pode caracterizar o processo no seu todo, ou apenas algumas etapas, as restantes funcionando em regime contínuo. É também possível encontrar na indústria instalações contínuas interligadas com unidades processuais que funcionam em modo descontínuo, denominando-se neste caso como semi-contínuas (Castro, 1995).

Processos de natureza essencialmente descontínua são frequentemente usados no fabrico de produtos químicos de elevado valor acrescentado, produzidos em quantidades reduzidas (*specialities*), na indústria farmacêutica, nas áreas alimentar e biotecnológica, no fabrico de tintas e vernizes, e ainda na produção de alguns tipos de polímeros. Para além dos baixos volumes de produção, outras características favorecem também geralmente o funcionamento em regime descontínuo:

- Uma maior flexibilidade operacional, o que permite o fabrico de uma variedade de produtos com características semelhantes no mesmo equipamento. Esta flexibilidade possibilita também que a instalação se possa adequar a novos requisitos de mercado, nomeadamente à variação da procura (por exemplo, variações sazonais).
- A versatilidade permite que uma instalação possa processar novos produtos sem que isso implique grandes alterações e/ou investimentos. Desta forma é possível reduzir significativamente os custos de investimento necessários, não apenas devido ao carácter multifuncional atribuído aos equipamentos instalados, como também à facilidade de utilização de tempos de residência mais elevados, mantendo os volumes úteis dos equipamentos necessários dentro de limites razoáveis.
- Razões operacionais específicas como a melhor rastreabilidade de lotes particulares de um determinado produto, ou uma gestão mais eficaz dos recursos disponíveis.

No entanto, são também conhecidas algumas desvantagens da operação em descontínuo:

- Normalmente, este tipo de operação envolve necessidades adicionais de mão-de-obra, devido à maior variedade de operações distintas (por exemplo, preparação, carga, operação regular, preparação do final de operação, descarga, limpeza, etc.).
- A operação em descontínuo origina também habitualmente requisitos adicionais ao nível da automação de processos, devido à necessidade de supervisão de um maior leque de operações.
- A operação em descontínuo pode, por vezes estar associada a processos menos eficientes, devido à existência de tempos mortos do equipamento ou ao aparecimento de alterações imprevisíveis nas sequências de processamento pré-estabelecidas. Por outro lado, também a recuperação de matérias-primas e as tarefas de integração energética são normalmente mais dificultadas nestes casos, relativamente ao modo de operação contínuo.
- Como consequência da sua maior flexibilidade, estes processos estão também sujeitos a mais fontes de variabilidade, as quais podem impactar a consistência da qualidade do produto final, se não forem adequadamente controladas ou os seus efeitos mitigados.

A tendência da indústria para o fabrico preferencial de produtos de elevado valor acrescentado vem colocar os processos químicos descontínuos numa posição de destaque. Esta tendência é consequência por um lado das fortes pressões de mercado resultantes de uma economia cada vez mais global. Por outro lado, ela é acentuada pelo desenvolvimento e pela aplicação crescente dos princípios de Engenharia de Produto, sinónimo de diversidade de produtos e soluções, que embora sejam de elevado valor, estão associados a baixos volumes de produção. A Indústria Farmacêutica é um exemplo desta tendência, encontrando-se na literatura estudos diversos ((Marques et al., 2020), (Marques et al., 2018), (Marques et al., 2017)), onde é possível observar que as técnicas de otimização/simulação trazem enormes vantagens na tomada de decisões, e consequentemente melhoria dos processos.

Muitas vezes a operação de um processo descontínuo decorre de acordo com receitas (i.e., sequências de procedimentos) baseadas em heurísticas e na experiência acumulada. Por outro lado, a necessidade de partilha de recursos na produção de múltiplos produtos requer a programação (*scheduling*) cuidada das várias operações intervenientes no processo de produção. Na supervisão da execução de cada um dos diferentes passos que fazem parte da

receita, recorre-se a técnicas de controlo de operação. O uso alargado destas técnicas é possível desde que elas possuam simultaneamente algumas características, tais como:

- **Flexibilidade:** permitindo tratar a variedade de objetivos operacionais habitual nestes problemas, com natureza eventualmente hierárquica. De forma idêntica, é importante considerar explicitamente os diversos tipos de restrições operacionais possíveis e os diversos tipos de modelos dos processos usados. Uma característica de flexibilidade igualmente desejável é também a fácil integração com outros sistemas de gestão de produção e de apoio à decisão eventualmente existentes, e com os operadores fabris (utilizadores).
- **Capacidade de ordenação:** para uma maior facilidade de utilização e de manutenção, deve ser escolhido um conjunto limitado de metodologias de supervisão para tratar as diversas fases de processamento existentes. A adaptação destas estratégias para novos produtos e formas de processamento diferentes deve ainda ser simples, e poder ser efetuada rapidamente.
- **Eficácia:** as soluções encontradas deverão permitir uma utilização eficiente dos recursos produtivos existentes, possibilitando a implementação de soluções “ótimas” segundo critérios operacionais claramente definidos, tendo em consideração toda a informação processual disponível.
- **Robustez:** devem ser consideradas as limitações existentes no tipo e na qualidade da informação processual disponível, limitando a sensibilidade das soluções produzidas a eventuais fontes de incerteza nos dados operatórios e nos modelos matemáticos considerados.

As instalações descontínuas são amplamente estudadas na literatura, sendo possível encontrar diversas classificações (Amaran et al., 2016; Ammeri, 2011; Figueira & Almada-Lobo, 2014a; M. C. Fu, 1994; Trigueiro de Sousa Junior et al., 2019). Da análise destas referências, encontrou-se uma referência compreensiva para a classificação de problemas de *scheduling* descontínuo (Méndez, 2006) sob várias perspetivas, que se apresenta na Figura 1. Esta figura é ilustrativa dos vários aspetos a considerar quando se está a desenvolver o modelo de *scheduling* para os processos *batch*. Este roteiro considera 13 características principais, cada uma das quais está ligada ao problema central.

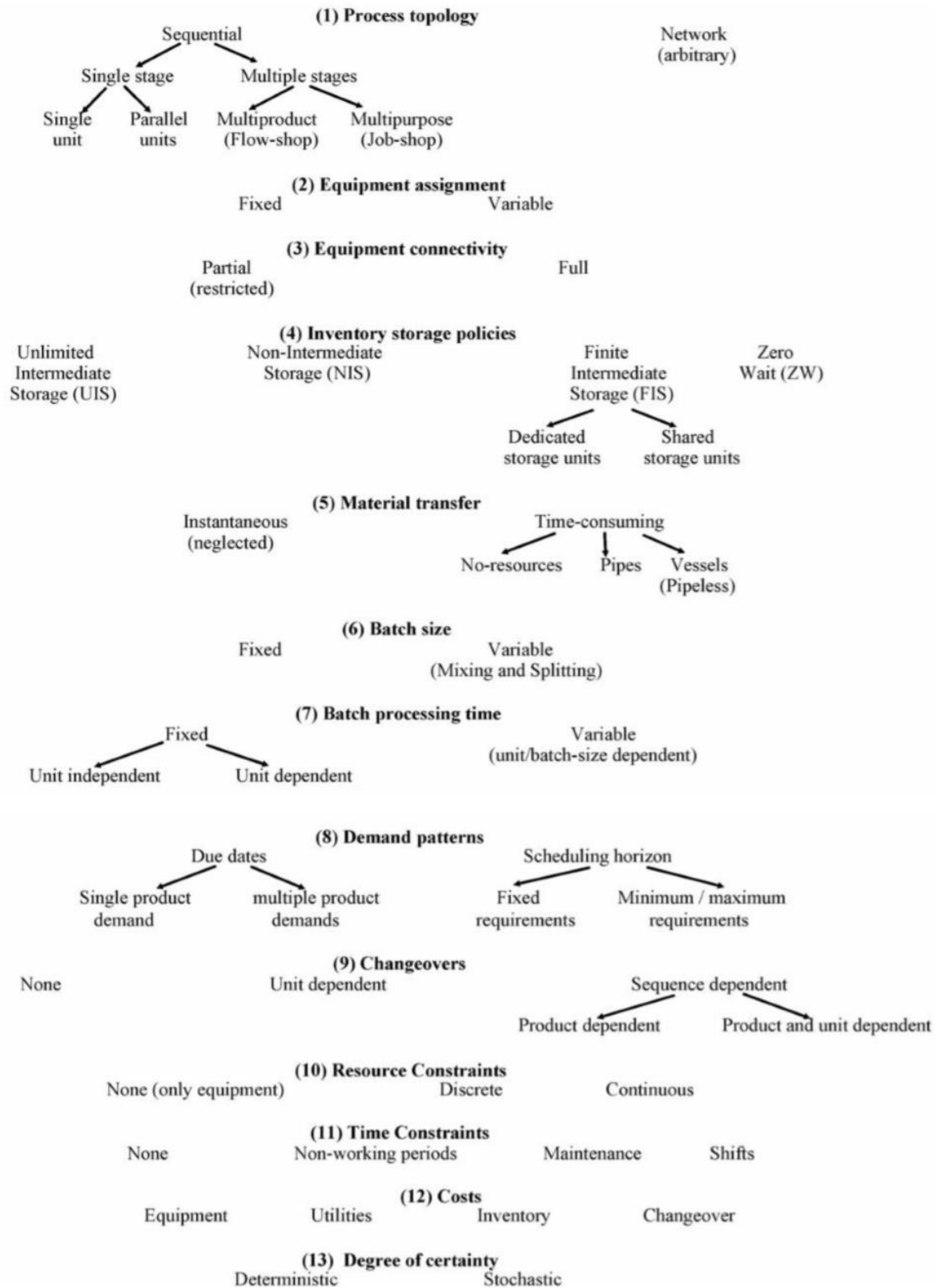


Figura 1 – Roteiro para problemas scheduling de processos descontínuos (Mendez, 2006)

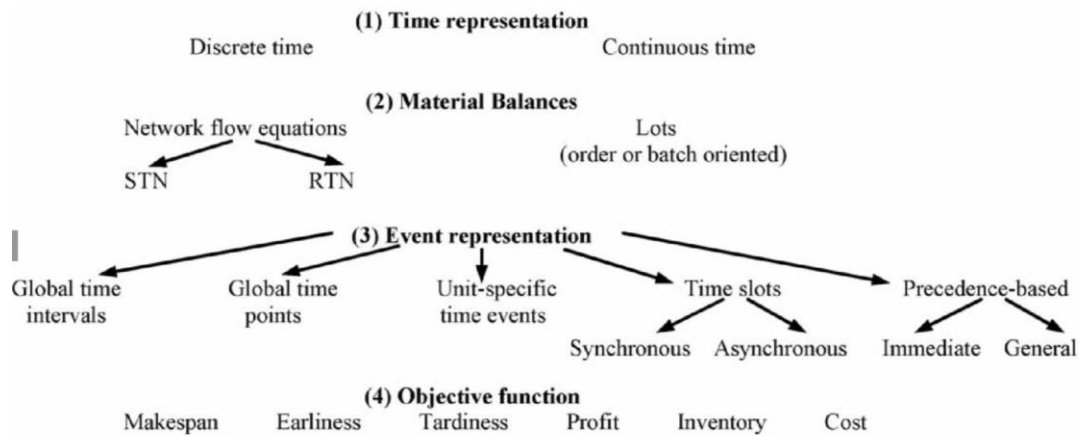


Figura 2 – Roteiro de modelos otimização para scheduling a curto prazo de processos descontínuos (Mendez, 2006)

Neste roteiro, analisando as figuras 1 e 2, podemos observar que são consideradas questões tão diversas, que necessariamente vão ter um enorme impacto na complexidade do problema. Por outro lado, esta diversidade de fatores dificulta a tarefa de encontrar modelos generalistas. A análise destas categorias é fundamental não só para a adequada modelação do problema, mas também para a sua compreensão e consequente resolução, pelo que se apresentam uma descrição detalhada dos fatores em análise.

(1) Topologia do processo.

Os processos descontínuos podem ser sequenciais, em estádio ou múltiplos estádios, onde uma ou mais unidades podem trabalhar em paralelo em cada estádio. Cada lote precisa de ser processado seguindo uma sequência de fases, definidas pela receita do produto.

(2) Políticas de armazenagem

É frequente as instalações terem uma política de armazenagem finita, ainda que nas instalações descontínuas seja muito comum encontrar processos de fabrico com tempos de espera nulos. A classificação das diferentes políticas de armazenagem descreve-se de seguida:

- NIS (Armazenagem Intermédia inexistente): Caso em que uma instalação não possui qualquer tanque para a armazenagem temporária dos seus produtos em qualquer das suas fases de processamento, ficando portanto, as unidades sujeitas a longos tempos de ocupação improdutivo.
- UIS (Armazenagem Intermédia Ilimitada): A instalação dispõe de capacidade de armazenagem ilimitada, isto é, para qualquer produto, em qualquer fase de processamento, existe sempre um tanque disponível para o receber.

- FIS (Armazenagem Intermédia Finita): A capacidade de armazenagem da instalação é finita, o que significa que nem sempre é possível dispor de um tanque de armazenagem para ser ocupado por um produto intermédio. Neste caso, o produto fica em “*hold-time*” a ocupar “improdutivamente” a unidade em que se encontra. Segundo esta política, as unidades de armazenagem podem estar localizadas por forma a servirem toda a instalação (FIS partilhada) ou colocadas entre duas unidades para serviço exclusivo na transição entre elas (FIS local)
- ZW (Inexistência de tempos de espera): Sempre que numa instalação se processe um produto instável, é recomendável que o seu ciclo produtivo não seja interrompido. Nestas situações, o produto só é introduzido na instalação quando existem garantias de que pode passar imediatamente à unidade seguinte sem esperar pela sua libertação.

(3) Gestão do stock

Dependendo do tipo de empresa e da política seguida internamente, a produção poderá ser direcionada para a satisfação dos seus clientes, “*Open-shop*” ou para manter determinadas quantidades em stock, “*Closed Shop*”. Na política “*Open-shop*” minimizam-se os prazos de entrega, evitando-se assim a aplicação de penalizações. Quando seguimos uma política “*Closed-shop*”, pretende-se minimizar o tempo total para cumprir um ciclo de produção, ou o tempo médio de permanência dos produtos na instalação.

(4) Duração do ciclo produtivo

O conjunto de operações que permite transformar a matéria-prima num lote de produto final – ciclo produtivo-, está intimamente ligado ao tempo que decorre entre dois ciclos produtivos do mesmo produto. Assim, podemos distinguir entre regimes sobrepostos (*overlapping*), onde o tempo limite de ciclo é igual ao tempo de processamento no estádio limitante (mais lento), e não sobrepostos (*non-overlapping*), onde o tempo limite de ciclo iguala o tempo de residência na instalação.

(5) Padrão da procura

A questão das exigências da procura leva à distinção das instalações entre Instalações Multiproduto e Multitarefa. Nas Instalações Multiproduto todos os produtos são submetidos, pela mesma ordem, às mesmas etapas de transformação, embora com diferentes tempos de processamento, enquanto que nas Instalações Multitarefa estão envolvidas várias famílias de produtos exigindo, por isso, a utilização de diferentes linhas de produção e partilha de recursos.

2.3. *Scheduling*

É possível encontrar na literatura várias definições de *scheduling*, sendo que Méndez (2006) refere-se ao *scheduling* como uma questão crítica nas operações dos processos e crucial para melhorar o desempenho da produção. Para processos descontínuos, o planeamento da produção a curto prazo lida com a alocação de um conjunto de recursos limitados no tempo, para a produção de um ou mais produtos que seguem uma receita descontínua.

Scheduling é uma metodologia de gestão de recursos com implicações quer ao nível do projeto das instalações (*scheduling* do projeto) quer ao nível da gestão da sua operação (*scheduling* de produção), segundo Castro (1995):

- O ***scheduling* do projeto** tem como objetivo definir a estrutura ótima da instalação que melhor se adequa ao fabrico de um dado conjunto de produtos, minimizando o investimento de capital. Identifica o número e a dimensão das unidades que devem ser colocadas em cada etapa do processo tendo em vista a minimização dos custos e a satisfação de um conjunto de requisitos de produção num horizonte de tempo limitado.
- O ***scheduling* da produção** tem como objetivo identificar o escalonamento ótimo (conjunto de produtos, ordem de processamento e percurso através da instalação) de um dado conjunto de produtos que otimiza um critério económico (custos de produção) ou um índice de desempenho industrial (tempo global de produção, prazos de entrega, etc.)

O *scheduling* é necessário sempre que existe competição entre atividades a decorrer com recursos limitados que estão disponíveis durante um período finito de tempo, e compreende três fases distintas, a saber:

1-**Atribuição de recursos** (Análise dos recursos necessários).

2-**Sequenciamento de atividades** (atribuição de uma sequência ordenada das atividades existentes).

3- **Determinação dos tempos de utilização dos recursos pelas atividades.**

2.4. Formulação dos problemas de planeamento

Após a análise das características gerais dos problemas de planeamento mais comuns, importa analisar as questões a ter em consideração na construção do modelo. A definição deste modelo e suas inerentes características vão ser decisivas, no esforço computacional exigido na resolução do problema, limitações e mais-valias geradas.

A classificação dos modelos de otimização para planeamento de produção descontínuo, segundo Méndez (2006) foca-se em 4 aspetos principais:

- **Representação temporal**

De acordo com o plano de produção, assim os eventos podem decorrer em momentos temporais predefinidos – modelos discretos, ou em qualquer momento do horizonte de tempo definido – modelos contínuos.

- **Balancos materiais**

Se o problema em análise assume que as tarefas de processamento produzem e consomem materiais, estamos perante um modelo STN (*State-Task Network*). Por outro lado, se todos os recursos (equipamentos e materiais) são tratados de forma uniforme, o problema é representado por um modelo RTN (*Resource-Task Network*).

- **Representação eventos**

O plano de produção inclui vários eventos ao longo do tempo, com o propósito de garantir que a capacidade máxima dos recursos partilhados não é excedida. Assim, é possível distinguir entre formulações em tempo discreto, quando se utiliza uma grelha de tempo fixa, e em tempo contínuo.

- **Função objetivo**

A função objetivo (ou várias no caso de modelos multiobjectivo) traduz matematicamente os objetivos definidos pelos decisores para as suas intervenções sobre os sistemas e que se pretende minimizar ou maximizar (ou seja, genericamente, otimizar).

A definição da função objetivo tem um papel decisivo na resolução do problema, uma vez que a sua complexidade vai influenciar o desempenho computacional.

O problema em estudo é transformado num modelo matemático, constituído pela função objetivo, pelas variáveis, pelas restrições e pelos parâmetros, sendo o objetivo obter as variáveis de decisão que otimizam a referida função objetivo.

Para a formulação do problema de planeamento de produção, é então necessário definir as **variáveis**, tais como o número de unidades, número de produtos, número de lotes de cada produto e política de armazenagem. As **restrições** são o conjunto de expressões matemáticas que descrevem o comportamento dos sistemas e representam as condicionantes a que as decisões e a função objetivo vão ser sujeitas. Dentro dos vários grupos de restrições, apresentam-se dois exemplos:

- Restrições Materiais, é o caso, por exemplo dos serviços auxiliares de frio (fluido de refrigeração), de aquecimento (vapor de água), eletricidade, etc. A utilização simultânea destas utilidades em diferentes setores da instalação/fábrica pode inviabilizar um dado plano de produção previamente considerado como ótimo.
- Restrições humanas estão ligadas à disponibilidade de mão-de-obra qualificada necessária para o funcionamento e supervisão das unidades. Estes recursos são limitados e deverão, portanto, ser atribuídos no tempo, tendo em vista a minimização dos custos envolvidos.

Os **parâmetros** são os valores de elementos que são ditados do exterior.

A **função objetivo**, que se pretende que seja otimizada, pode consistir em:

- minimizar o tempo final de processamento - “*makespan*” (tempo ao fim do qual o último produto da sequência sai da instalação).
- minimizar o valor máximo ou médio do tempo de residência - “*flowtime*”, ou seja, o tempo total para completar um ciclo produtivo.
- minimizar o valor máximo ou médio do atraso nas entregas dos diferentes produtos (“*tardiness*”).
- minimizar os custos de mudança de regime (“*Change-over*”).

Na literatura analisada (Amaran et al., 2014; Figueira & Almada-Lobo, 2014), um problema genérico de otimização é descrito por:

$$\begin{aligned} \min \quad & f(\theta) \\ \text{s. a.} \quad & g(\theta) \geq 0 \\ & h(x,y) < 0 \end{aligned}$$

onde: $f(x,y,w)$ e $g(x,y,w)$ são, respetivamente, a função objetivo e as restrições determinadas por um modelo de simulação. θ representa o vetor das variáveis de entrada, x variáveis entrada

contínuas, y variáveis entrada discretas, variáveis aleatórias e restrições que não envolvem variáveis aleatórias. Esta formulação é muito genérica e pode ser aplicada numa vasta gama de domínios, que vão desde a engenharia à medicina, passando pela biologia, transportes e logística.

2.5. Métodos de resolução de problemas de planeamento

Depois de estabelecida a função objetivo e as restrições, é necessário resolver o problema matemático, ou seja, determinar os valores das variáveis de decisão que minimizam ou maximizam a função objetivo, satisfazendo as restrições. Na literatura (Ammeri, 2011; Carson & Maria, 1997; Figueira, 2019; Frazzon et al., 2016) existem duas vias principais: métodos de simulação e métodos otimização.

Os **métodos baseados em simulação** são procedimentos iterativos para obtenção das melhores soluções por tentativa e erro. Por intuição, experimentam-se diferentes soluções fixando os valores das variáveis de decisão, verifica-se a não violação das restrições e avalia-se a qualidade das soluções, nomeadamente através do valor da função objetivo. Baseiam-se em regras empíricas, não garantindo, portanto, uma solução ótima global. Estes métodos estão muito dependentes do fator humano, o que pode levar a soluções ineficientes para sistemas complexos. São pouco exigentes em tempo de cálculo e proporcionam, regra geral, boas estimativas iniciais para outros métodos.

Os **métodos de otimização** baseiam-se em técnicas de programação matemática, com as quais se procura determinar os valores das variáveis de decisão que permitem otimizar uma função objetivo definida, tendo em consideração as restrições estabelecidas, e os limites das variáveis de decisão. Recorrem a procedimentos de cálculo que, em geral, garantem uma solução ótima global. São bastante exigentes em tempo de cálculo, mas não dependem do tipo de função objetivo, da política de armazenagem e do número de unidades.

Na literatura encontramos variadas metodologias, onde há uma interligação entre estes dois métodos, surgindo o conceito de Simulação Otimização (SO). As possibilidades de combinar Simulação e Otimização – Métodos Simulação-Otimização – são vastas, e a escolha da abordagem mais adequada depende do sistema em estudo, variáveis em causa, grau de incerteza, etc.

2.5.1. Métodos de otimização

Na resolução de um problema de otimização, o objetivo é determinar o ótimo global. Mas por vezes encontrar o ótimo global é muito mais difícil do que encontrar ótimos locais, tendo-se por isso desenvolvido várias abordagens ao problema. A figura 3 mostra a classificação dos métodos de otimização global, encontrada na literatura (Edgar & Himmelblau, 2001). Os métodos exatos, baseados numa abordagem determinística, garantem a obtenção de um ótimo global e conseguem assegurá-lo, sendo a metodologia *Branch-and-bound* um exemplo bem conhecido e largamente utilizado. Os métodos heurísticos, baseados numa abordagem não determinística, não garantem a obtenção de soluções ótimas globais, sendo incapazes de provar que encontraram esta solução ótima, mesmo quando a encontraram. No entanto, são largamente utilizados.

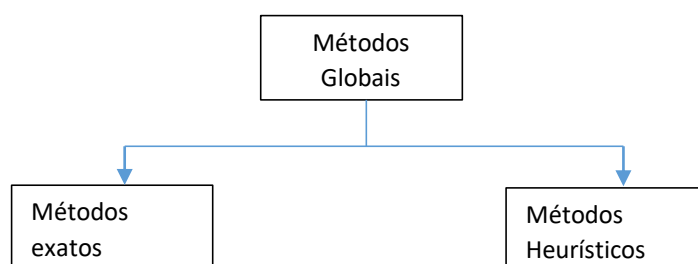


Figura 3 - Classificação dos Métodos de otimização global (adaptado)

(Edgar & Himmelblau, 2001)

Métodos Exatos

No contexto de uma abordagem determinística e de uma representação conceptual da realidade na qual se faz a distinção entre decisões estruturais e decisões operacionais, um modelo de otimização depende essencialmente do tipo de funções matemáticas e da natureza das variáveis de decisão.

A existência de técnicas de programação matemática que permitem resolver de forma eficiente os modelos lineares (e.g., método *Simplex*) faz com que, sempre que possível, se representem os sistemas com modelos deste tipo. Dependendo da sua dimensão, os modelos de otimização inteira podem ser ainda resolvidos por técnicas de resolução que garantem a determinação da solução ótima global (i.e., solução ótima para todo o espaço de solução) no caso de esta existir (como exemplo o método de *Branch-and-Bound*). A complexidade na resolução de um modelo de otimização aumenta consideravelmente na presença de modelos não lineares. Sem condicionamentos à forma das funções, os métodos de resolução de modelos não lineares garantem apenas a determinação de ótimos locais (i.e., solução ótima apenas na

sua vizinhança). No entanto, existem métodos para modelos convexos que permitem a obtenção da solução ótima global (exemplo, o método do Gradiente Reduzido Generalizado).

Métodos Heurísticos

As abordagens não determinísticas são baseadas maioritariamente em metodologias heurísticas. Os métodos de pesquisa heurística iniciam com uma solução corrente, exploram todas as soluções na vizinhança desse ponto, à procura de uma melhoria, e repetem o processo, sempre que encontram uma melhoria na solução. Os métodos heurísticos são conduzidos por algoritmos meta heurísticos, tais como: pesquisa tabu, pesquisa dispersão e algoritmos genéticos. Estes algoritmos utilizam um procedimento heurístico para o problema, o que por si só pode não permitir encontrar o ótimo global, podendo ser necessário reorientar a pesquisa para que esta não fique “presa” num ótimo local.

Os algoritmos genéticos (Edgar & Himmelblau, 2001) usam métodos heurísticos que imitam alguns processos biológicos. Estes métodos são baseados na população e combinam um conjunto de soluções, num esforço para encontrar soluções melhoradas e de seguida atualizar a população com essa solução. Devido à sua natureza aleatória, o algoritmo genético aumenta as possibilidades de encontrar uma solução global.

Os algoritmos genéticos diferem dos algoritmos tradicionais de otimização em basicamente quatro aspetos:

- Baseiam-se em uma codificação do conjunto das soluções possíveis, e não nos parâmetros da otimização em si;
- Os resultados são apresentados como uma população de soluções e não como uma solução única;
- Não necessitam de nenhum conhecimento derivado do problema, apenas de uma forma de avaliação do resultado;
- Usam transições probabilísticas e não regras determinísticas.

Nas abordagens não determinísticas, o efeito da variação dos valores dos parâmetros incertos pode ser avaliado à posteriori, ou em alternativa, incorpora-se explicitamente algum conhecimento acerca da incerteza na resolução dos problemas para se obterem soluções que sejam menos sensíveis a variações dos valores dos parâmetros.

Existindo incerteza em relação ao conhecimento de um determinado parâmetro, efetua-se a respetiva análise de sensibilidade através de resoluções sucessivas do modelo considerando valores diferentes do parâmetro incerto. Se a análise de sensibilidade revelar que a solução não é sensível a variações do valor do parâmetro incerto, então é provável que a solução obtida seja

apropriada. Pelo contrário se a solução for sensível a variações do valor do parâmetro incerto, pode não ser fácil identificar uma solução fiável. Naturalmente, a análise de sensibilidade torna-se mais complicada se os problemas forem de elevada dimensão e se se estiver na presença de diferentes parâmetros de incerteza. Perante vários parâmetros não totalmente conhecidos, além da incerteza em relação ao valor numérico, é necessário ter em consideração as correlações que se estabelecem entre os diferentes parâmetros (temporalmente e/ou espacialmente). As análises de sensibilidade podem facilmente consumir bastante tempo sem que existam garantias de que alguma das soluções encontradas seja a solução ótima.

Desenvolvimentos relativamente recentes no domínio da investigação operacional, juntamente com a evolução tecnológica dos meios informáticos, conduziram a uma alteração de paradigma. As heurísticas modernas correspondem a uma nova classe de métodos que têm vindo a permitir resolver problemas complexos, em que apenas uma enumeração completa das soluções garantiria a determinação da solução ótima global. No entanto, para a maior parte dos problemas reais, a pesquisa de todo o espaço de solução é simplesmente impraticável. As heurísticas modernas têm-se revelado como métodos eficientes na determinação de soluções aproximadas à solução ótima global.

2.5.2. Métodos de simulação

O comportamento de um sistema ao longo do tempo pode ser estudado recorrendo a um modelo de simulação, onde se definem matematicamente os pressupostos assumidos do sistema. Um modelo pode ser desenvolvido matematicamente ou através de um estudo estatístico. Uma vez que a maior parte dos sistemas reais são demasiados complexos para serem formulados e testados matematicamente, é frequentemente necessário recorrer à simulação computacional. Graças ao avanço tecnológico e computacional, a linguagem de modelação tornou-se uma ferramenta muito útil, permitindo uma diminuição dos custos de operação, a par de um aumento nas metodologias em operação.

Na simulação utilizam-se determinadas técnicas matemáticas (com recurso aos computadores), as quais permitem prever o funcionamento dos sistemas. Assim, com a simulação, conseguimos descrever o comportamento do sistema, construir teorias e hipóteses considerando as observações efetuadas, e ainda usar o modelo para prever o comportamento futuro, isto é, os efeitos produzidos por alterações no sistema ou nos métodos empregues na sua operação.

Uma vez que os modelos de simulação podem ser quase tão detalhados quanto desejável, novas políticas e procedimentos operacionais, regras de decisão, fluxos de informação, podem ser avaliados sem que o sistema real seja perturbado. Assim, pode-se compreender melhor quais as variáveis que são mais importantes e como as mesmas interagem entre si e com os outros elementos do sistema.

No entanto a modelação e a experimentação associadas a modelos de simulação consomem muitos recursos, principalmente tempo. A construção de modelos muito detalhados implicam um grande consumo de tempo de desenvolvimento, sujeito a uma lenta execução computacional, com maior incerteza e dificuldade de verificação. De facto, o maior nível de detalhes não implica, necessariamente maior precisão – tal só acontecerá se forem conhecidos boas estimativas para os parâmetros decorrentes do detalhe acrescido. O nível de detalhes deve ser apenas suficiente para satisfazer os objetivos traçados.

Segundo Banks (1998), o diagnóstico de problemas é também um benefício proveniente da utilização de um *software* de simulação. O diagnóstico de problemas, dificuldades e a realização de uma análise sobre a importância dessas variáveis aumenta a compreensão dos seus efeitos sobre o desempenho do sistema global. A simulação permite uma compreensão clara em relação às interações entre as variáveis constituintes do sistema.

Ao proceder-se à simulação, constrói-se uma opinião objetiva, uma vez que os resultados deste processo foram previamente modelados, testados, validados e representados visualmente (Banks, 1998).

No contexto da simulação é necessário existir uma predisposição para a mudança e para a conceção de novos sistemas e reformulação de sistemas já existentes. É importante a interação com todos os envolvidos no projeto de simulação, durante a fase da formulação do problema, para uma melhor análise dos cenários em causa e, assim, construir um modelo que responda às necessidades relativas desses cenários (Banks, 1998). Em termos de custos, um estudo de simulação é substancialmente menos dispendioso em comparação com os custos inerentes à instalação de um sistema ou à sua modificação, sem um estudo prévio (Banks, 1998).

2.5.3. Métodos de simulação e otimização

Segundo Figueira & Almada-Lobo (Figueira & Almada-Lobo, 2014b), tradicionalmente, olha-se para a simulação e para a otimização como dois métodos distintos para resolver problemas. No entanto o desenvolvimento dos computadores e da tecnologia tem permitido que surjam abordagens que combinam ambos os métodos. O objetivo desta junção é

explorar simultaneamente o detalhe fornecido pela simulação e a excelente capacidade que as técnicas de otimização têm, em encontrar boas soluções, ou mesmo ótimas num dado sentido.

Um dos maiores desafios a que a combinação Otimização/Simulação tenta dar resposta é a presença de incerteza. Nos últimos anos, esta tem sido uma área a que os investigadores se têm dedicado, seguindo uma série de diferentes abordagens. Sahinidis (2004) analisa as diferentes abordagens à otimização com incertezas, comparando a programação estocástica, *fuzzy* e dinâmica estocástica. Sahinidis (2004) faz uma análise detalhada destes métodos, concluindo que foram feitos grandes progressos, sendo que a dinâmica estocástica oferece um potencial para solucionar problemas complexos onde pode faltar por exemplo um modelo preciso. Por outro lado, a utilização destas abordagens gera resultados com uma precisão bastante inferior à das abordagens de simulação.

Nos últimos anos, vários investigadores (Ammeri, 2011) têm-se dedicado a combinar os procedimentos de simulação e otimização, com vista a obter uma solução mais robusta ou completa. Associar a simulação com a otimização requer maior esforço computacional. Por outro lado, há inúmeras formas de associar simulação com otimização. Assim, é essencial ter uma visão geral das abordagens híbridas, vantagens e inconvenientes, para que se possa escolher uma estratégia de resolução adequada.

Os critérios aplicados na classificação dos métodos de Otimização/Simulação podem-se basear em:

- **Técnicas aplicadas** (procedimentos estatísticos, abordagem gradientes, heurística, etc)
- **Propriedades de convergência** (associadas aos critérios de convergência)
- **Otimidade** (pesquisa do ótimo local e/ou ótimo global)
- **Formulação** (espaço de solução e função objetivo)

A questão torna-se ainda mais complexa, quando pensamos em combinar diferentes critérios. Figueira & Almada-Lobo (2014b) faz uma análise detalhada dos critérios, e propõe uma classificação dos métodos Simulação Otimização com base em 4 dimensões:

- **Objetivo simulação** (relacionada com a interação entre Simulação e otimização)
- **Estrutura Hierárquica** (relacionada com a interação entre Simulação e otimização)
- **Método de pesquisa** (relacionada com o design do algoritmo de pesquisa)
- **Esquema de pesquisa** (relacionada com o design do algoritmo de pesquisa).

As duas primeiras dimensões estão relacionadas com a interação entre simulação e otimização. Quando se refere o propósito da simulação, incluem-se as principais correntes de pesquisa na Simulação-Otimização, que se baseiam na avaliação da função e se distinguem pela existência ou não de um modelo de substituição (*surrogate*), para orientar a pesquisa. Em termos de estrutura Hierárquica, as diferenças estão relacionadas com o modo como as iterações decorrem e se o procedimento se inicia pela otimização ou pela simulação.

As duas últimas dimensões relacionam-se com o *design* do algoritmo de pesquisa. Quanto aos métodos de pesquisa, distinguem-se os métodos exatos (encontram e provam a existência de um ótimo global) e os métodos heurísticos (são métodos que não garantem o ótimo global, mas permitem fazer progressos na pesquisa). O esquema de pesquisa está relacionado com a sequência de soluções e iterações consideradas computacionalmente.

No trabalho de pesquisa desenvolvido por Figueira & Almada-Lobo (2014b) foram combinadas estas 4 dimensões, originando uma taxonomia que permite classificar os métodos Otimização/Simulação existentes na literatura, bem como as suas variantes. Desta análise, é possível constatar que alguns métodos propostos na literatura têm um procedimento claramente especificado (por exemplo a aproximação estocástica), enquanto que outros métodos têm abordagens mais gerais (por exemplo a abordagem simulação otimização recursiva, ROSA)

As estratégias a utilizar dependem das características do problema e das suas relações. Mas é evidente que se vai desvanecendo a fronteira existente entre abordagens exclusivamente baseadas na otimização e as abordagens exclusivamente baseadas na simulação.

Segundo Cheng, Feng e Hsu (2006), duas formas podem ser usadas a fim de aplicar a Simulação como ferramenta de avaliação e melhoria do desempenho de um processo. Uma delas é selecionar todas as configurações de recursos disponíveis e então executar a simulação para cada uma delas, analisando os resultados encontrados. Porém, este processo pode consumir muito tempo. Outra forma de avaliação consiste em testar algumas combinações de recursos, com base em algum critério de seleção, com a finalidade de otimizar o desempenho do processo avaliado. Desta forma, utiliza-se a otimização combinada à simulação, como um processo que testa várias combinações com diferentes valores para as variáveis controláveis, na tentativa de buscar uma solução ótima. Segundo Fu (2002), até pouco tempo atrás, na prática a simulação e a otimização eram mantidas de forma separada. Hoje em dia essa integração tem-se difundido bastante, devido ao fato de alguns pacotes de simulação incluírem rotinas de otimização. A combinação da otimização e simulação deu origem ao termo *Simulation Optimization*, que neste trabalho é chamado de Simulação com Otimização.

Apesar dos processos de produção industriais serem cada vez mais complexos, por outro lado, cada vez há mais informação em tempo real, o que torna as abordagens Simulação otimização muito úteis na resolução de problemas reais. E são vários os problemas reais que podem aplicar estas abordagens, sendo que este trabalho se debruça sobre o planeamento de produção.

Na determinação dos melhores perfis de operação, a definição adequada da função objetivo a utilizar no problema de otimização pode constituir por vezes uma tarefa complexa; existem simultaneamente, em muitos casos, vários objetivos de operação que competem entre si, tornando-se necessário estabelecer compromissos entre estes, ou usar técnicas de decisão multi-critério. No domínio dos processos descontínuos, a minimização do tempo final de operação constitui uma das funções objetivo mais frequentemente adotadas para a determinação de trajetórias ótimas de operação.

3. Metodologia

O foco desta dissertação é a análise de soluções de otimização discretas via simulação, ou seja, o enfoque será em encontrar configurações ótimas para variáveis que apenas podem ter valores discretos.

Existem no mercado vários pacotes que incorporam metodologias (incluindo rotinas de otimização), que permitem resolver um problema de simulação de eventos discretos. (Amaran et al., 2016) faz um estudo detalhado dos *softwares* existentes, comparando os algoritmos que servem de base ao software, e classificando-os de acordo com o objetivo em vista, o tipo de variáveis discretas/contínuas, e se o foco é a otimização global ou local.

Neste capítulo introduzem-se metodologias de simulação, apresentando-se detalhadamente o *software* utilizado neste trabalho, a ferramenta *Simulink*, que faz parte do software *Matlab*.

3.1. Ferramentas e tecnologias de simulação

A versatilidade, a flexibilidade e a dinâmica dos métodos de simulação caracterizam a sua aplicabilidade em processos de tomada de decisão, tendo ênfase no setor industrial. No setor industrial é crucial que os gestores e os engenheiros possuam uma visão ampla do efeito que determinadas alterações desencadeiam no sistema.

Como foi abordado no capítulo anterior, o desenvolvimento computacional tem tido um papel de extrema importância na Simulação Otimização, permitindo a criação de *software* que auxilia a análise destes sistemas. O método com maior incidência e com maior utilização na compreensão e gestão de processos é a simulação de eventos discretos, em sistemas de produção, sendo que os *softwares* ProModel, ARENA e Flexsim estão entre os mais utilizados.

O *software* ProModel é um dos métodos mais adequados em termos de fluxos de trabalho e na engenharia de processos. A funcionalidade deste *software* é a modelação de sistemas de produção, desde sistemas flexíveis de fabrico até sistemas de cadeia de abastecimento. O ProModel possibilita a oportunidade de testar novas ideias ou por outro lado, a melhoria contínua do processo. Foca-se na utilização de recursos, na capacidade do sistema, na melhoria dos processos, no rendimento, na análise de restrições e na logística e cadeia de abastecimento.

Em alternativa, o ARENA é outro *software* de simulação que possibilita às empresas aumentar a sua produção, ao analisar o fluxo do processo produtivo (os sistemas de embalagem, o controlo do stock) e ao avaliar as potenciais mudanças neste processo.

O *software* Flexsim é uma ferramenta de análise, desenvolvimento e suporte à simulação, orientada para o desenvolvimento, a modelação, a simulação, a visualização e a monitorização das atividades do processo. Este *software* permite a construção de um modelo tridimensional e computacional, que traduz um sistema real. Através de animações, gráficos e relatórios de desempenho, possibilita a identificação de problemas e a avaliação de soluções alternativas, num reduzido horizonte temporal.

Paralelamente, a plataforma de cálculo numérico Matlab permite também o estudo de um sistema real, dado ser possível construir aplicações para a indústria. A construção de uma aplicação com uma interface gráfica no *Matlab* permite automatizar uma determinada função, que se utiliza diversas vezes e, para além disso, permite partilhar a terceiros uma demonstração interativa. Uma das vantagens da sua utilização é o facto de conferir comunicação entre os diferentes componentes, sem ter que utilizar comandos de programação. Estes componentes têm sido desenvolvidos e conferem ao programa, um conjunto de ferramentas intuitivo. Estas ferramentas, por sua vez, conferem um aumento da produtividade por parte do utilizador.

No desenvolvimento desta dissertação recorreu-se ao *Matlab-Guide*, para a elaboração e a construção de uma aplicação, com capacidade de simular computacionalmente o sistema em causa.

3.2. Modelação conceptual

Um modelo, ainda que seja uma simplificação do sistema, requer muito treino e conhecimento para ser construído. No entanto, é possível identificar alguns passos comuns na análise de simulação computacional de um sistema

- **Formulação do problema** – O problema em estudo deve ser bem conhecido.
- **Definição dos objetivos** – Os objetivos de vem ser claros, próximos da realidade.
- **Criação do modelo** – No início deve ser simples, aumentando a sua complexidade gradualmente.
- **Recolha de dados** – Deve ser completa e pormenorizada. Recorre-se a ferramentas computacionais, o que permite trabalhar com um elevado número de dados.
- **Validação** – Para que o modelo seja o mais próximo possível da realidade, deve fazer-se um número elevado de simulações.
- **Análise** – Com os dados obtidos no ponto anterior, faz-se uma análise aos mesmos.

- **Implementação** – A implementação depende do sucesso dos pontos anteriores sendo o mais importante a validação, se todos os passos forem executados então a simulação deve ser implementada.

Há várias vantagens na simulação computacional, quando comparada com simulação real. A primeira prende-se com os custos envolvidos: é muito menos dispendioso efetuar experiências com o modelo de simulação do que com o sistema real, especialmente quando são utilizados equipamentos com um custo elevado. Um outro aspeto, tem a ver com os riscos presentes, quer materiais quer humanos, na realização de testes reais, pois os resultados são inesperados enquanto na simulação computacional isso não ocorre, os testes são realizados em ambientes virtuais.

3.3. Software Matlab

MATLAB® (MATrix LABoratory) foi criado no final de 1970 para cursos de teoria matricial, álgebra linear e análise numérica e, hoje em dia, tornou-se um ambiente de computação técnica da maior importância. O *Matlab* é um *software* interativo, que integra a capacidade de fazer cálculos, visualização gráfica e programação em um ambiente fácil de usar, em que problemas e soluções são expressos em uma linguagem matemática familiar e muito versátil. Os usos típicos para o *Matlab* incluem:

- Cálculos matemáticos;
- Desenvolvimento de algoritmos;
- Modelagem, simulação e confecção de protótipos;
- Análise, exploração e visualização de dados;
- Gráficos científicos e da engenharia;
- Desenvolvimento de aplicações, incluindo a elaboração de interfaces gráficas com o usuário.

O *Matlab* possui diversos módulos, chamados de pacotes, que agrupam funções para diferentes aplicações. Uma das ferramentas mais conhecidas e que foi amplamente utilizada neste trabalho é o *Simulink*.

3.3.1. Simulink

O *Simulink* é uma ferramenta para modelar, simular, e analisar sistemas dinâmicos, baseando-se em modelos.

A figura seguinte apresenta um pouco do ambiente de trabalho do *Simulink*. É neste ambiente que foram implementados os casos de estudo deste trabalho. A versão do *Matlab* utilizada neste trabalho foi a 2019b.

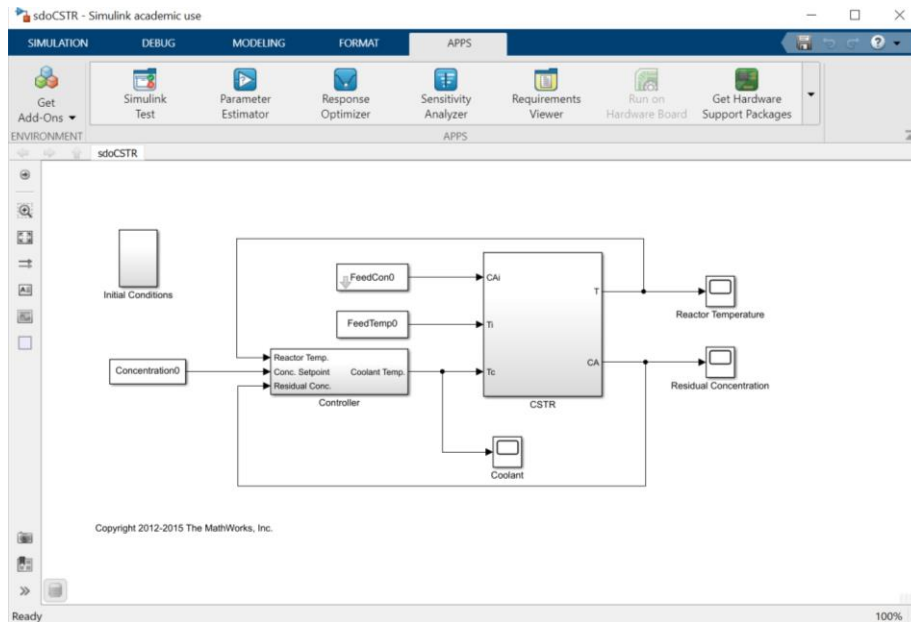


Figura 4 - Ambiente de trabalho do Simulink

O *Simulink* é um *software* que suporta grandes variedades de sistemas, tais como, lineares e não-lineares, modelados em tempo contínuos, tempo discreto ou a mistura dos dois. Para modelar, o *Simulink* possui uma interface gráfica com utilizador (*GUI - Graphical User Interface*) e para construir modelos utiliza diagramas de blocos, usando as operações de clicar-e-arrastar do rato. Com esta interface, o programador pode construir os modelos de forma simples, como se efetua no papel. Esta abordagem constitui um avanço relativamente às soluções tradicionais, que se baseiam na programação e utilização direta de métodos numéricos. O *Simulink* inclui um conjunto de bibliotecas de blocos pré-definidos, que ajudam na redução de tempo, para a criação do modelo de simulação, uma vez que os blocos já se encontram nas bibliotecas, não sendo necessário programá-los desperdiçando-se tempo, alguns blocos permitem também que se vejam os resultados enquanto a simulação estiver em execução. Além disso, o utilizador consegue em tempo real alterar os parâmetros e verificar os resultados

obtidos. Os resultados da simulação, podem ser enviados para o *workspace* do *Matlab* para visualização e utilização posterior.

Outras ferramentas do *Matlab* podem ser usadas no *Simulink*, razão pela qual se considera que o *Matlab* e o *Simulink* estão integrados. É possível simular, analisar e rever os modelos em qualquer ambiente em qualquer ponto. O *Simulink* é uma extensão gráfica do *Matlab*, para a simulação dinâmica de sistemas.

Existem duas fases distintas de uso: a definição do modelo e análise do modelo. Para facilitar a definição do modelo, o *Simulink* fornece uma ferramenta gráfica de edição em blocos, onde o modelo é criado e editado tipicamente com o rato.

Depois de definir o modelo, este pode ser analisado diretamente, no menu na janela atual do *Simulink*, ou entrando nos comandos devidos, pela janela de comando do *Matlab*. Dentro das ferramentas de análise de modelos, temos a simulação propriamente dita, linearização de modelos, etc. Muito sucintamente podemos considerar alguns pontos de relevância e que se podem encarar como vantagens em relação ao ambiente *Matlab*:

- Permite observar o comportamento dos sistemas de uma forma prática e fácil;
- Permite representar os sistemas de controlo, como se de um diagrama de blocos se tratasse;
- Aplicável para sistemas de controlo discretos;
- Cooperação entre o ambiente *Matlab* e o ambiente *Simulink* (pode-se exportar e importar informação entre ambos os ambientes);
- Existência de um conjunto de blocos pré-definidos que incluem, controladores, filtros, elementos somadores, funções de transferência, etc.;
- Capacidade de utilização de bibliotecas de tempo real para aplicações;
- Um procedimento consideravelmente simplificado;
- Possibilidade de um uso múltiplo de modelos matemáticos básicos;
- A possibilidade de comparar os resultados teóricos e experimentais;

O *Simulink* permite que se represente um sistema dinâmico por meio de diagramas em blocos. Para definir um sistema, os blocos são copiados das bibliotecas de blocos padrões do *Simulink*. A biblioteca padrão é organizada em vários subsistemas, que agrupam blocos, de acordo com a função pretendida. Para correr o *Simulink*, primeiro é necessário inicializar o *Matlab* clicando duas vezes sobre o ícone do *Windows* e então entrar no comando *Simulink*. Este comando mostra uma nova janela contendo ícones dos blocos de subsistemas que constituem a biblioteca padrão.

Existem diferentes termos nas linguagens de simulação que utilizam definições diferentes para conceitos semelhantes ou até iguais. Por isso, para uniformizar e explicitar as ferramentas utilizadas, é importante definir claramente os conceitos base: entidade, evento, atividade, etc.

A entidade é um termo usado para designar uma unidade de produção. Um evento é um acontecimento que muda o estado de um modelo. Num modelo de um sistema de uma linha de enchimento, por exemplo, a chegada de uma encomenda pode ser simulada por uma entidade trazida para o modelo de simulação. Existem dois tipos possíveis de entidades, referidas como entidades externas e entidades internas. Entidades externas são aquelas cuja criação e movimento é explicitamente providenciado pelo modelador. Em contraste, entidades internas são criadas e manipuladas implicitamente pelo *software* de simulação em si. Por exemplo, as entidades internas podem ser usadas em algumas línguas para simular falhas da máquina, enquanto entidades externas podem ser usadas para simular o uso de máquinas. No caso de estudo neste trabalho, a entidade é a ordem de produção dos produtos A e B. O termo recurso designa um elemento do sistema que fornece o serviço (por exemplo, reatores disponíveis, fornecimento de água ou eletricidade). Os utilizadores de recursos são normalmente entidades. (Por exemplo, um transporte de matéria prima para um "armazém de entrada"). Os recursos são geralmente de capacidade limitada.

Uma atividade representa normalmente um tempo de serviço, um tempo ou outro tempo de processo cuja duração foi caracterizada ou definida pelo modelador, podendo ser especificada de várias maneiras.

A lista de eventos futura está sempre orientada pelo tempo que o evento demorou. O termo elemento de controlo designa uma construção que suporta outros tipos de alternativas de atraso ou lógica com base no estado de um sistema. Os elementos de controlo podem assumir a forma de interruptores, balcões, valores de dados de utilizadores, e valores de dados do sistema construído na ferramenta de modelação. As condições de controlo complexos podem ser baseados em expressões que usam aritmética e/ou combinações de elementos de comando booleanas. Uma operação é um passo levado a cabo por ou sobre uma entidade, enquanto se move através de um sistema, por exemplo.

Em resumo:

- **Sistema.** Um conjunto de entidades (pessoas ou máquinas) que interagem entre si de modo a atingir os objetivos.

- **Modelo.** Representação abstração de um sistema, normalmente contém uma relação estrutural matemática ou lógica que descreve o sistema em termos de estado, entidades, processos, eventos, atributos e atrasos.
- **Estado de um sistema.** Um conjunto de variáveis que contém toda a informação de modo a descrever o sistema a qualquer momento.
- **Atributos.** Propriedades de um sistema.
- **Lista.** Um conjunto de entidades associadas, ordenadas de uma forma lógica.
- **Facto.** Uma ocorrência que acontece num determinado período no futuro, mas com a informação para tal.
- **Lista de Eventos.** Uma lista de eventos que ocorrem infinitamente, ordenado pelo termo de operação, também conhecida por Future Lost Event (FEL).
- **Tempo (T).** Representa uma espécie de tempo de simulação em cada modelo. O tempo é um valor inteiro e neste padrão não tem unidades específicas para a medir.
- **Evento (E).** Um evento especifica um sistema instantâneo de transição entre dois estados diferentes ao mesmo tempo T.
- **Entidade (e).** Uma entidade é algo que é processado, por exemplo, uma matéria-prima ou um produto. Vai através de algumas filas e atividades que representam a produção de sistema e é definido pelos seus atributos.
- **Fila (Q).** É uma fila de entidades administradas com algumas especificidades da filosofia de fila (LIFO, FIFO, etc.).
- **Atividade (A).** É um processo ou ação demorado, ou o passo com uma duração conhecida.
- **Recurso (R).** Um recurso é algo que é necessário para uma atividade começar. Em ambientes de produção, normalmente as máquinas e operadores.
- **Estrutura de dados (D).** É semelhante a um recurso, mas existem diferenças de modelação em termos de semântica.

3.3.2. Descrição dos blocos no *Simulink*

As linguagens de simulação e os simuladores já têm um conjunto enorme de ferramentas pré definidas para auxílio do analista e modelador. O *Simulink* para além de algumas bibliotecas com blocos comuns, tem uma biblioteca específica de simulação de eventos discretos. Descrevemos de seguida os principais blocos que usamos da biblioteca do *Simulink* de eventos discretos.

➤ **Bloco "Input Switch" (Filtro de entradas)**

Este bloco aceita entidades através de um número definido de portas de entrada e só tem uma única porta de saída. Pode especificar-se o número de portas de entrada do bloco. Se várias entidades chegarem ao bloco "Input Switch" simultaneamente, enquanto a porta de saída entidade não é desbloqueada, a sequência das entidades de partida dependem da sequência de eventos de partida de blocos que precedem o bloco "Input Switch". Embora o tempo de partida seja o mesmo para todas as entidades, a sequência pode afetar o comportamento do sistema. Por exemplo, se as entidades avançarem para uma fila, a sequência de partida determina as suas posições na fila.

➤ **Bloco "Single Server" (Atividade)**

Este bloco representa uma atividade sobre uma entidade por um período de tempo e, em seguida, se a porta de saída está bloqueada, então a entidade permanece neste bloco até que a porta fique desbloqueada. Se uma entidade neste bloco está prevista para um tempo limite, então pode sair prematuramente através uma saída opcional. Se foi especificado o tempo de serviço, que é a duração do serviço, através de um parâmetro, atributo, ou de sinal, dependendo do tempo de serviço definido pelo parâmetro. O bloco determina o tempo de serviço para uma entidade após a sua chegada. Tempos de serviço são assumidos como especificados em unidades definidas de início do modelo de simulação.

➤ **Bloco "Output Switch" (Filtro de saídas)**

Este bloco recebe uma entidade de cada vez e define a saída através de uma das portas de saída fazendo que a entidade possa seguir um caminho diferente na sua simulação. A porta selecionada pode mudar durante a simulação.

➤ **Bloco "Set Attribute" (Definição de atributo)**

Este bloco aceita uma entidade, atribui-lhe valores e parâmetros. Os dados atribuídos são armazenados em campos específicos de acordo com tipo de atributo, nome e valor.

➤ **Bloco "Random Source" (Fonte aleatória)**

O bloco Random Source gera valores aleatórios de 0 a 1 seguindo uma distribuição uniforme. É possível definir outras distribuições para servir de padrão para gerar números aleatórios.

➤ **Bloco "FIFO Queue" (Fila de espera FIFO)**

Este bloco armazena N entidades ao mesmo tempo, onde N é o valor do parâmetro de capacidade. O bloco retém a entidade se a porta de saída estiver bloqueada. Se o bloco está a armazenar várias entidades porque a porta de saída está bloqueada, o bloco armazena as entidades partir do conceito "first-in, first-out" modo (FIFO), isto é, sai primeiro a entidade que chegou primeiro. Se uma entidade tiver um tempo limite definido de espera, pode sair prematuramente através de uma saída opcional do bloco. O período de tempo que uma entidade fica neste bloco não pode ser antecipadamente determinado.

➤ **Bloco "Entity Combiner" (Combinador de atividades)**

Este bloco gera uma nova entidade para cada conjunto de entidades que chegam simultaneamente das várias portas de entrada. As entidades que chegam são chamados entidades componentes. Podem representar diferentes partes dentro de uma entidade maior. As escolhas de parâmetro deste bloco determina se outros blocos podem terminar os atributos ou temporizadores das entidades e se a operação de combinação é reversível. Algumas escolhas de parâmetros exigem exclusividade de nomes de atributos ou temporizadores nas entidades componentes. O bloco Combiner tem múltiplas entradas e uma porta de saída. A operação de combinação ocorre quando todas as entidades de componentes necessários estão prontos e a entidade resultante tem possibilidade de sair. Mais explicitamente, quando todos os blocos que se ligam às portas de entrada do bloco Combiner tem uma entidade pendente ao mesmo tempo e a porta de ligação para a porta de saída estiver disponível.

➤ **Bloco "Entity Sink" (Armazenamento de entidades")**

Este bloco fornece uma maneira de encerrar o percurso de uma entidade. Ao selecionar a porta de entrada disponível para chegadas de entidades, o bloco aceita sempre chegadas de entidades.

➤ **Bloco "Math Function" (Função matemática)**

O bloco Função Matemática realiza inúmeras funções matemáticas, que servem para alterar parâmetros de atributos de entidades, definir tempos, comparar valores, etc.

➤ **Bloco "Get Attribute" (Recolha de atributos)**

Este bloco gera valores usando os dados de atributos de entidades. Para cada entidade de chegada, o bloco atualiza o sinal nas portas de saída usando os valores dos atributos definidos. O bloco deixa a entidade inalterada.

➤ **Bloco "Enable Gate" (Portão de saída)**

Este bloco representa uma porta que está aberta quando o sinal de controlo, na porta de entrada é positivo, e fechada sempre que o sinal é zero ou negativo. Por definição, um portão aberto permite que as entidades que chegam possam avançar imediatamente para o bloco seguinte, ao passo que um portão fechado proíbe a passagem das entidades.

3.3.3. “*Simulink design optimization*”

O *Simulink* permite aceder a um módulo bastante completo e versátil na área da otimização, denominado “*Simulink Design Optimization*”. Este *software* resolve problemas diversos na área da otimização, recorrendo a algoritmos de otimização, uma vez que converte os requisitos do projeto num problema de otimização com restrições. Este problema simula iterativamente o modelo no *Simulink*, compara os resultados das simulações com os objetivos das restrições, e recorre a métodos de otimização para fazer ajustes e alcançar os objetivos. Os métodos mais utilizados são por exemplo: o método dos gradientes descendentes, método pesquisa *Simplex* e método pesquisa padrão.

Existe ainda uma aplicação específica para análise de sensibilidade, que permite identificar os parâmetros que têm maior impacto no comportamento do modelo. O segundo caso de estudo, apresentado no capítulo 4.2, é um exemplo de como se pode aplicar esta ferramenta num estudo de análise de sensibilidade.

4. Casos de estudo

A abordagem analisada nesta dissertação é categorizada como híbrida, uma vez que consiste no uso iterativo entre os diferentes métodos de otimização e simulação, onde se procura atenuar as limitações do primeiro tirando partido da análise efetuada com base no segundo. Cada um dos métodos de otimização e simulação apresenta diferentes vantagens e desvantagens quanto à sua natureza, conteúdo e aplicação, como foi visto no capítulo 2.5. Os métodos de otimização garantem soluções ótimas globais e baseiam-se em formulações matemáticas que se adaptam ao problema em estudo, no entanto tais soluções podem não apresentar a robustez necessária para a sua implementação industrial. A análise da robustez pode ser feita simulando condições próximas da solução ótima e verificando o quanto se degrada o desempenho do sistema. A busca por uma solução ótima e robusta depende assim da interação entre os dois métodos.

Esta abordagem mista é aplicada neste trabalho, nos dois casos de estudo analisados, sendo que a formulação dos problemas segue as diretivas referidas nos capítulos 2.3 e 2.4 para problemas de planeamento da produção. Neste sentido, define-se o problema, constituído pela função objetivo, pelas variáveis, pelas restrições e pelos parâmetros, e cujo objetivo é encontrar as variáveis de decisão que otimizam a função objetivo.

Nesta dissertação são estudados dois exemplos, com o intuito de mostrar a versatilidade do *software Matlab/Simulink* em casos de planeamento da produção. No subcapítulo 4.1 é descrito o caso de estudo: Otimização de recursos partilhados num sistema de produção *batch* (*Mathworks.com*). Neste caso, o sistema é trabalhado em ambiente *Simulink* do *Matlab*. No subcapítulo 4.2, no caso de estudo relativo ao projeto de um reator *batch* ((Bequette, 1998), recorre-se à simulação de Monte Carlo, utilizando para isso o *Simulink*, em especial a ferramenta *Simulink Design Optimization*.

4.1. Otimização de recursos compartilhados num sistema produção *batch*

Conforme já abordado no capítulo 2.2, as instalações descontínuas com processos *batch* são muito comuns na indústria, sendo por isso objeto de estudo na investigação. Por esta razão, e uma vez que o problema deve ser representativo da complexidade existente na indústria, optou-se por analisar um sistema descontínuo com processos *batch*, baseado no modelo-exemplo do *Matlab* (“*Optimization of shared resources in a batch production process*”, *Mathworks.com*).

Este trabalho debruça-se sobre um processo de produção descontínuo com recursos partilhados, que processa dois tipos de ordens de produção, A e B. As ordens de produção são processadas de acordo com a disponibilidade dos reatores *batch*. O modelo é híbrido, baseado no tempo e em eventos discretos. A figura 5 mostra o modelo dividido em 3 componentes principais: o subsistema “Ordens de produção”, o subsistema “Plano de produção” e o subsistema “Análise de Dados”, enquanto que a figura 6 mostra detalhadamente o processo de produção que irá ser estudado.



Figura 5 - Representação esquemática do sistema

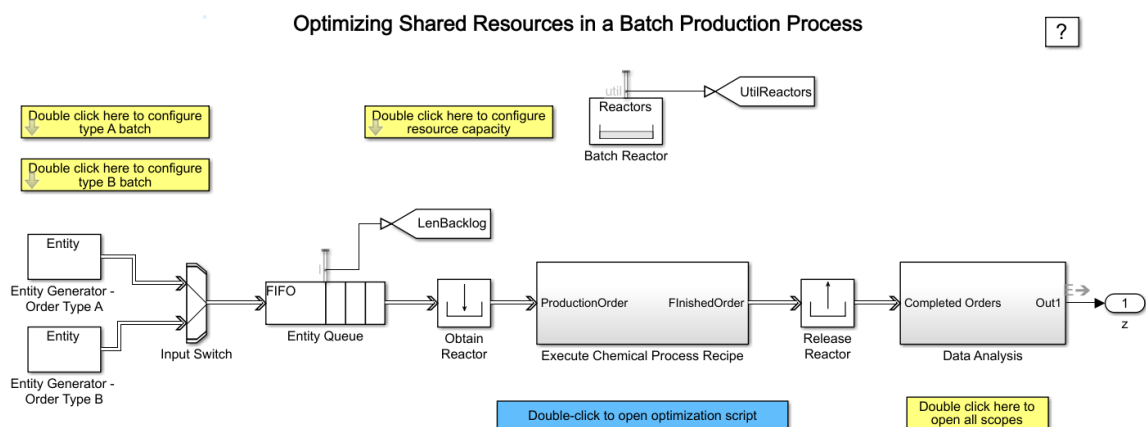


Figura 6 – Processo de produção baseado no modelo do Simulink “*Optimization of shared resources in a batch production process*”

O subsistema “Ordens de produção”, cuja representação esquemática se apresenta na figura 7, vai gerar ordens de produção com diferentes necessidades de recursos. Definiram-se as ordens de produção dos lotes A e B, que após serem geradas, aguardam numa fila de espera, “Queue”, até que o reator esteja disponível. Ambas as ordens de produção requerem diferentes tempos de disponibilidade dos recursos partilhados.

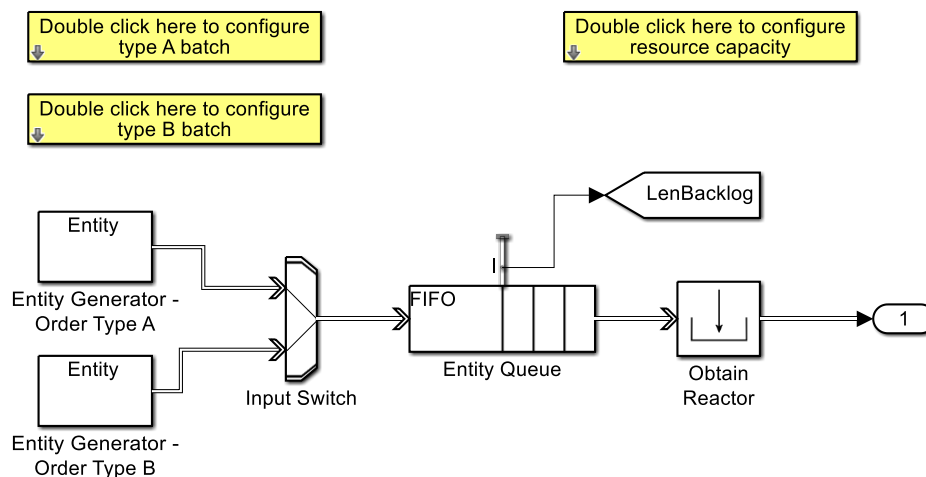


Figura 7 - Subsistema: Ordens de Produção, baseado no modelo “Optimization of shared resources in a batch production process”

O subsistema “Plano de Produção”, cuja representação esquemática se apresenta na figura 8, é um ciclo que atribui reatores *batch* a uma determinada ordem de produção e processa essa mesma ordem. Os reatores *batch* são um dos principais recursos partilhados deste sistema, sendo que um reator é utilizado para processar uma ordem de produção ou lote. No modelo, os reatores aguardam no bloco “FIFO Queue”, até serem necessários. Após a utilização, os reatores *batch* são libertados e novamente introduzidos no processo, para serem utilizados noutra ordem de produção.

A produção processa-se da seguinte forma. Assim que uma ordem de produção é recebida, o bloco “Obtain Reator” solicita um reator ao subsistema “Batch Reactor”. Dá-se então início ao processo químico, constituído por uma sequência de fases, que incluem adição de água, aquecimento da mistura, adição de cor e partículas, agitação e termina com a saída de efluentes. O reator *batch* tem a capacidade de executar as fases de adição de cor e partículas e a agitação, pelo que os recursos necessários para estes passos não precisam de ser modelados em separado. Por outro lado, os passos de adicionar água, aquecimento e efluentes exigem recursos extra. Estes recursos são partilhados por todos os reatores *batch* e são limitados pela capacidade de sistema de produção. Por exemplo, quando a utilização de água atinge a

capacidade máxima, a pressão de água é muito baixa para que entre outro reator *batch*. Nesse caso, a produção no interior do reator pára, até que o fornecimento de água esteja novamente disponível. Em linguagem de *Simulink*, este processo de partilha de recursos é efetuado com um “*Queue Block*” denominado “*Wait for water supply*” e um bloco “*Entity Server*”, denominado “*Add water*”, no subsistema “*Add water*”. O parâmetro “*Capacity*” do “*Entity Server*” modela a capacidade de fornecimento de água. Durante a simulação, o número de entidades no “*Queue Block*” indica o número de reatores *batch* à espera de água. O número de entidades no “*Server Block*” representa o número de reatores *batch* a utilizar água. Assim que o processo químico termina, o reator *batch* é libertado para ser usado noutra ordem de produção.

O processo de produção *batch* em causa tem a capacidade de produzir dois tipos de lotes: o tipo A e o tipo B. Ainda que os passos principais para produzir cada um deles sejam os mesmos, a receita do processo químico é diferente. Por exemplo, a receita para produzir o tipo B utiliza mais água, pelo que o passo “*Add water*” demora mais tempo a ser executado.

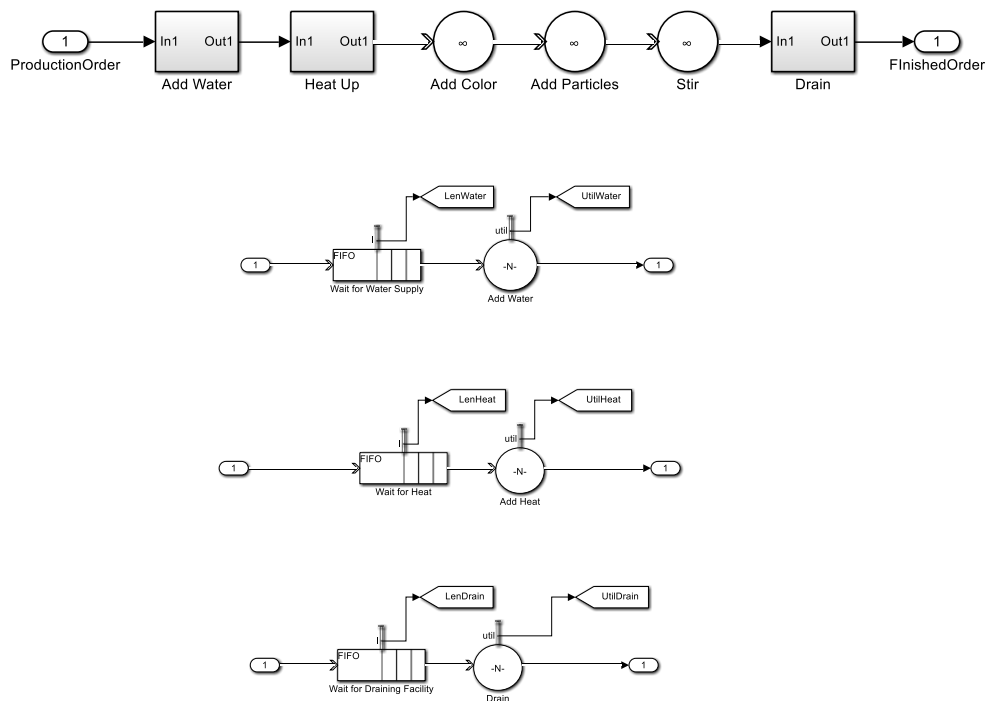


Figura 8 - Subsistema: Plano de Produção baseado no modelo “*Optimization of shared resources in a batch production process*”

Finalmente, o sistema “Análise do processo”, cuja representação esquemática se apresenta na figura 9, analisa a informação relacionada com o processo de produção e mostra como é que os recursos do processo de produção foram usados ao longo do tempo, mostrando nos resultados a utilização de cada um dos recursos, a média de ordens pendentes “Average order backlog” e a produção dos lotes “Throughput A” e “Throughput B”. O resultado mais ilustrativo é o “Average number of orders in backlog”, que representa o tempo de espera das ordens de produção, à medida que o sistema tenta manter o fluxo de entrada.

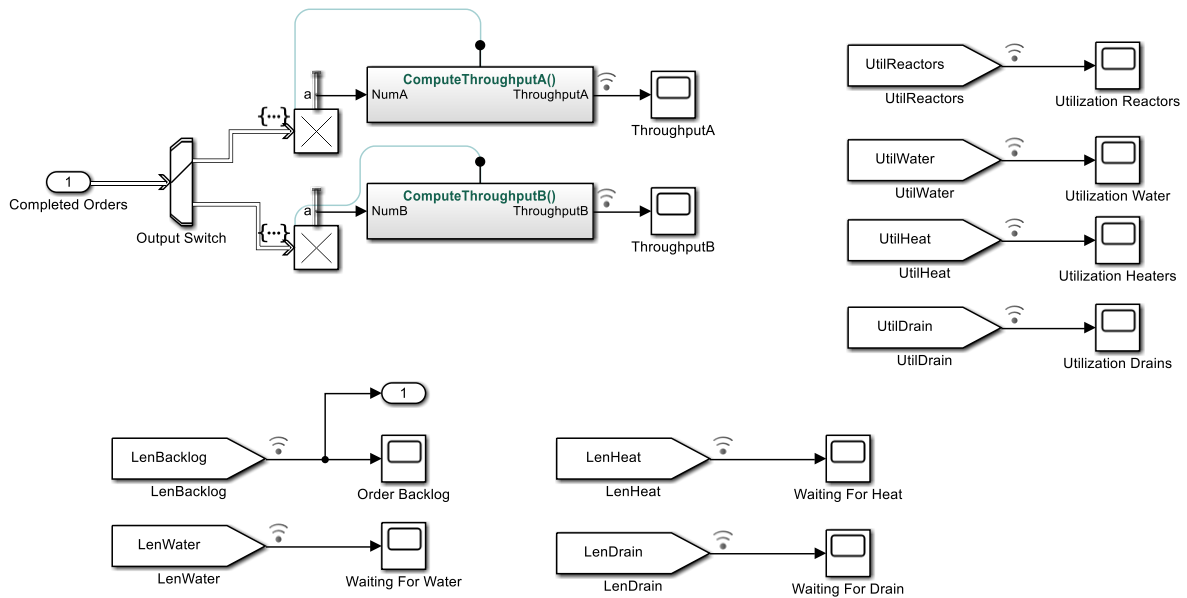


Figura 9 - Subsistema: Análise de dados baseado no modelo “Optimization of shared resources in a batch production process”

4.1.1. Implementação e tratamento de dados

Como já foi referido no capítulo 3.3, utilizou-se o *software Matlab* na implementação do modelo descrito na secção anterior. A ferramenta do *Matlab* utilizada é o *SIMEvents*, que se foca na simulação de eventos discretos. Esta ferramenta, pelas suas enormes potencialidades, tem um papel importante no tratamento de dados. Mas para isso, foi necessário proceder inicialmente a um processo de análise e observação, com o intuito de enquadrar, executar e automatizar cálculos e informações, bem como agilizar de um modo simples os *inputs* e *outputs*.

Nesta fase do projeto, após observações dos dados e limitações do sistema, foi possível adaptar o Sistema “*SeExampleBatchProduction*”, tornando-o flexível no tratamento e interpretação de dados. Como descrito na secção anterior, este modelo mostra como modelar e otimizar recursos partilhados num sistema, identificar deficiências nos recursos e melhorar a capacidade do sistema.

No sistema em estudo, as “*entities*” representam ambas as ordens de produção do processo, e os reatores *batch* necessários para as processar. Quando uma nova entidade é gerada, o bloco “*Obtain Reactor*” requisita um reator *batch* para processar a ordem. Depois do subsistema “*Execute Chemical Process Recipe*” completar a ordem de produção, o bloco “*Release Reactor*” liberta o Reator *batch* de volta para o depósito de recursos, ficando então o reator disponível para processar nova ordem.

Otimização – Algoritmo genético

O *Matlab* dispõe de uma ferramenta “*Matlab Global Optimization Toolbox*” para a otimização de modelos descontínuos baseados em eventos discretos, que permite uma grande flexibilidade na construção do modelo e na otimização da função objetivo, ao mesmo que facilita a ligação entre otimização e simulação discreta, sem necessidade de recorrer a múltiplos *softwares*.

A aplicação do algoritmo genético (que suporta otimização mista de inteiros-MIP) neste modelo vai permitir encontrar uma combinação adequada dos equipamentos a adquirir (reatores *batch*, tanques água, aquecedores e esgotos), minimizando as ordens de produção que ficam pendentes e o custo de adicionar recursos ao processo.

As variáveis de decisão nesta otimização são:

- número de reatores *batch* (R),
- fontes de água (W),
- aquecedores (H),
- efluentes (D).

No Anexo A apresenta-se o algoritmo utilizado neste trabalho (*seOptimizationForBatchProductionProcess.m*, sem data), cujo objetivo é minimizar a função objetivo programada em ambiente *Matlab*. A primeira parte da função objetivo define os valores para o número de reatores batch, fontes de água, aquecedores e efluentes. A função “*Set_param*” permite definir no *Simulink*, valores provenientes do *Matlab*. A função objetivo é expressa em função das ordens pendentes e do custo de adicionar mais recursos. O algoritmo genético vai correr a simulação várias vezes, com o objetivo de minimizar quer as ordens pendentes, quer o custo de adicionar mais recursos. Na figura 10, podemos observar o *output* fornecido pelo *software*.

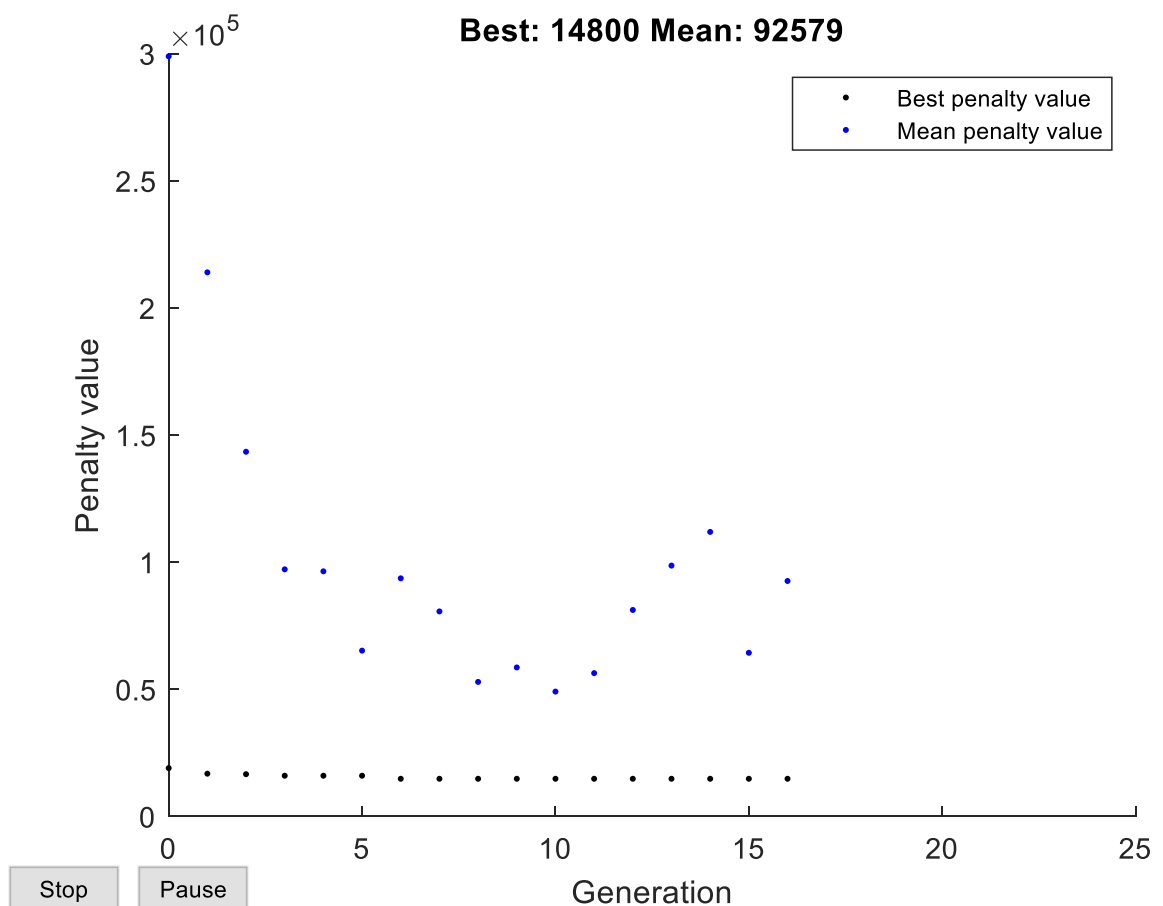


Figura 10 - Output do Algoritmo de otimização

A metodologia utilizada para encontrar a solução otimizada é apresentada na figura 11, e consiste na aplicação do algoritmo genético ao sistema, para encontrar uma solução otimizada para os recursos do sistema – variáveis de decisão: número de reatores *batch*, fontes de água, aquecedores e efluentes. O algoritmo genético define as variáveis de decisão à medida que corre múltiplas simulações do modelo através destas variáveis. A função objetivo é avaliada e a otimização termina quando a variação média no valor de ajuste for menor que a tolerância, ou quando a variação média no valor de ajuste for menor que a restrição, definidas no algoritmo, ou seja, quando o critério de paragem for cumprido.

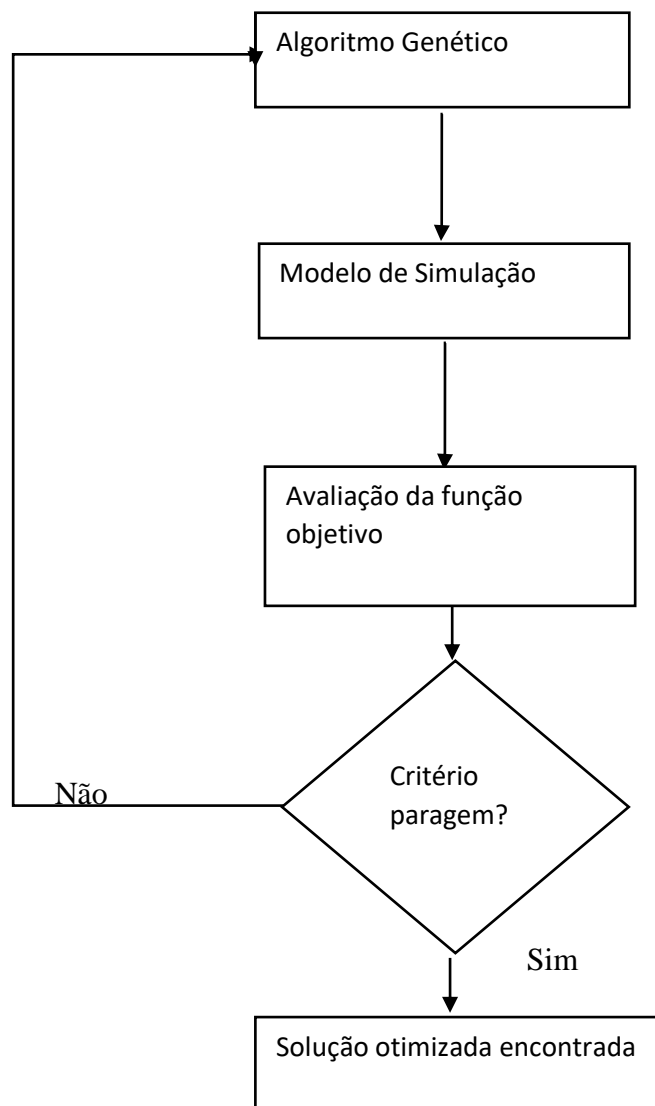


Figura 11 - Metodologia Simulação Otimização

4.1.2. Análise 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 ao introduzir a incerteza no sistema. Esta incerteza foi introduzida, aplicando uma distribuição exponencial na geração das ordens de produção.

De tal forma, irão ser apresentados resultados de diversas simulações, de acordo os Cenários apresentados. Estas simulações têm por base todo o processo descrito previamente no Capítulo 3, recorrendo aos blocos do *SimEvents*.

De forma a simular o mesmo comportamento, todas as simulações terão como base 1000 minutos de simulação, uma vez que, como é possível observar na figura 12, é o tempo ao fim do qual o sistema já estabilizou.

- **Cenário 1: Situação Inicial vs otimização**

Como ponto de partida, simulamos o processo inicial, atribuindo às variáveis de decisão o valor 2.

- número de reatores *batch* ($R=2$),
- fontes de água ($W=2$)
- aquecedores ($H=2$),
- efluentes ($D=2$).

As ordens de produção A e B são geradas com períodos fixos de 10 e 15 minutos, respetivamente.

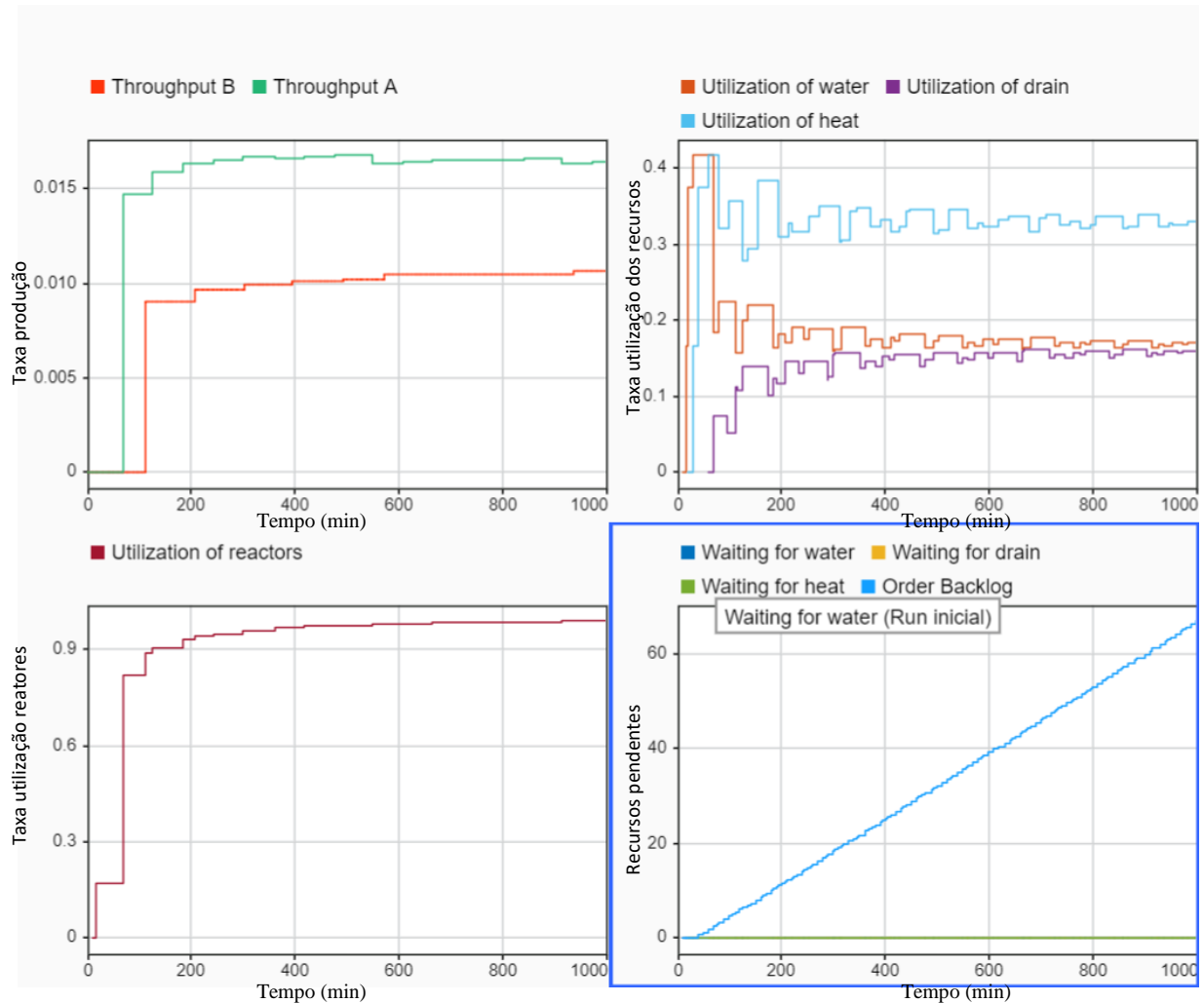


Figura 12 - Resultados da Simulação Inicial

Nesta simulação inicial, apesar da taxa de utilização dos reatores ser elevada, verifica-se que as ordens de produção pendentes são em número bastante elevado, aproximadamente 67 ordens de produção pendentes, o que significa que a taxa de chegada de ordens de produção ao sistema é superior à taxa de saída, logo o sistema não tem capacidade suficiente.

De seguida, corre-se o algoritmo de otimização-simulação, que tem por objetivo otimizar a função objetivo. A função objetivo “ $obj = backlog * 10000 + cost$ ” é a soma dos custos (custos do equipamento e custos adicionais de recursos) com as ordens de produção pendentes, que têm uma fator de penalização de 1000 (Anexo A). Pretende-se assim encontrar os melhores valores para as variáveis de decisão, encontrando o melhor equilíbrio entre os custos e as ordens de produção pendentes.

Após otimização, obtivemos então como resultados para as variáveis de decisão:

- número de reatores *batch*: 13
- fontes de água: 2
- aquecedores: 4
- efluentes: 2

Aplicando agora os resultados da otimização no modelo e voltando a simular verifica-se uma redução significativa nas ordens pendentes, conforme se pode visualizar na figura 13. O tempo de fila de espera é nulo, o que mostra que o sistema passa a ter capacidade suficiente e o seu funcionamento é mais regular.

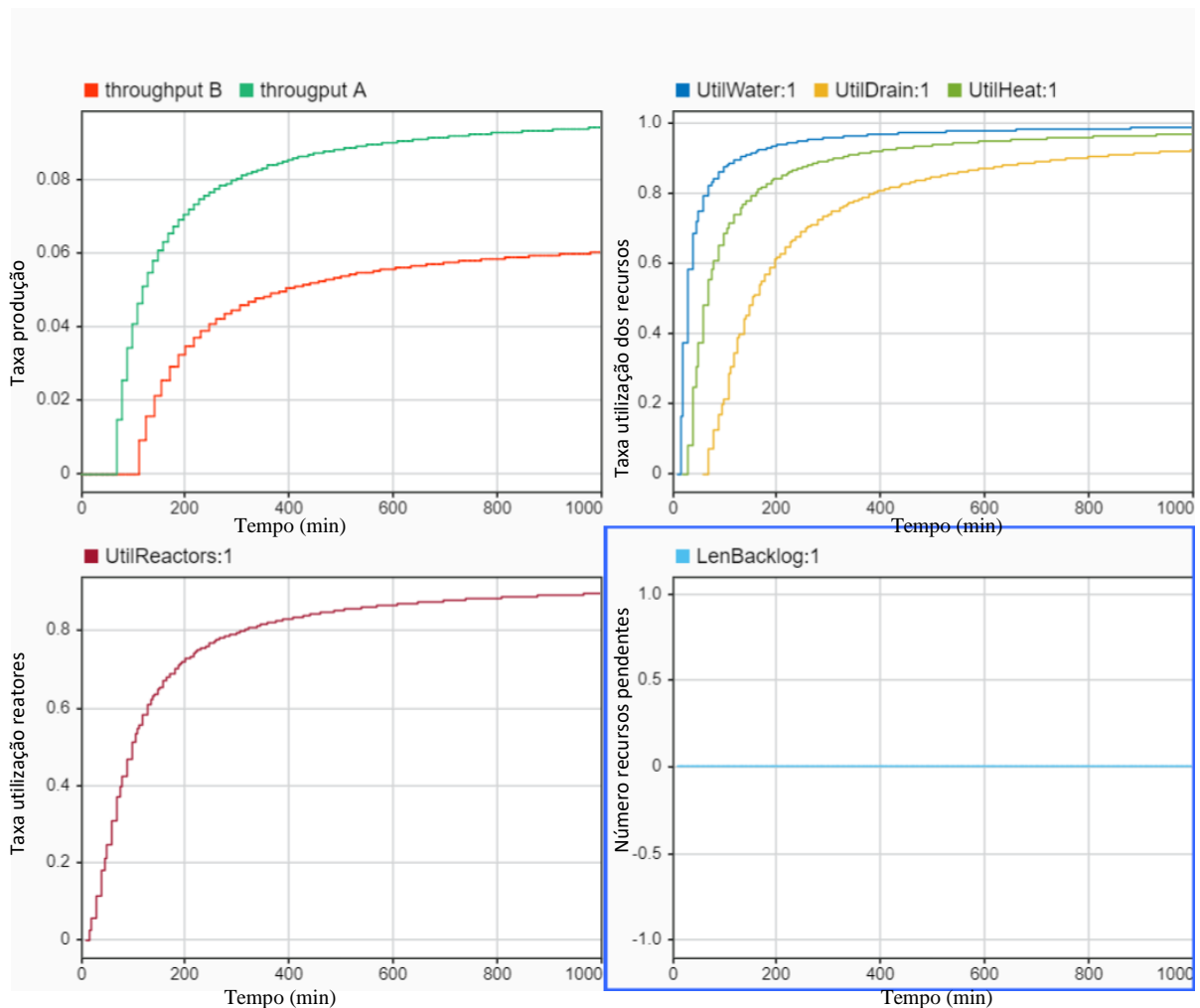


Figura 13 - Cenário 1: Simulação após otimização

Ainda pela observação da figura 13 é possível verificar que após a otimização, a taxa de utilização dos reatores é aproximadamente 85% e o número de ordens de produção pendentes é nulo.

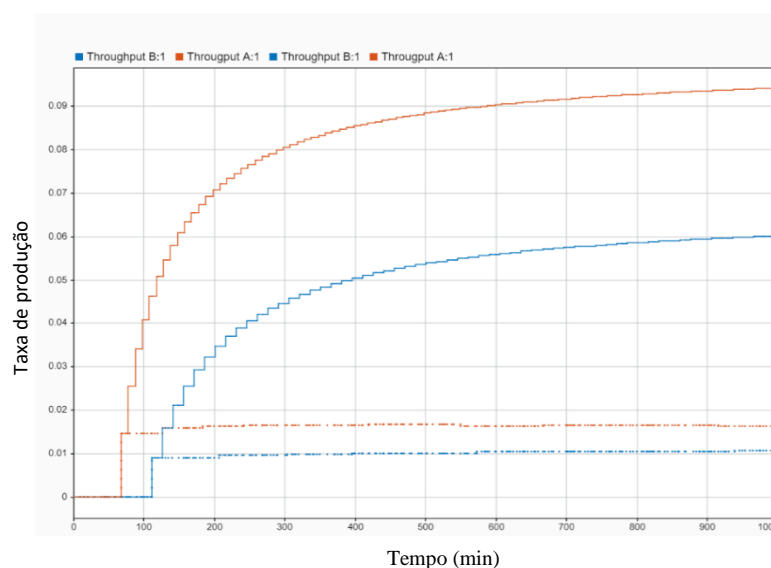


Figura 14 - Comparação das ordens de Produção A e B, antes e após otimização

A solução obtida após otimização mostra um aumento significativo da taxa de produção de A e B, como é possível constatar pela análise da figura 14.

- **Cenário 2 – Aplicação de incerteza às ordens de produção A e B**

Neste caso, vamos explorar como a incerteza aplicada nas variáveis do plano de produção (adição de água, calor, esgoto) vai interferir nos resultados. Partiu-se da situação otimizada, em que as variáveis de decisão tomam os seguintes valores:

- número de reatores *batch*: (R=13)
- fontes de água: (W=2)
- aquecedores: (H=4)
- efluentes: (D=2)

Inicialmente a incerteza foi aplicada ao tempo de intergeração da ordem de produção A. Foi utilizada uma distribuição exponencial, uma vez que é a distribuição que é frequentemente utilizada como modelo para distribuição dos tempos entre a ocorrência de sucessivos eventos, como é o caso dos tempos de geração das ordens de produção A e B. Os resultados apresentam-se na figura 15.

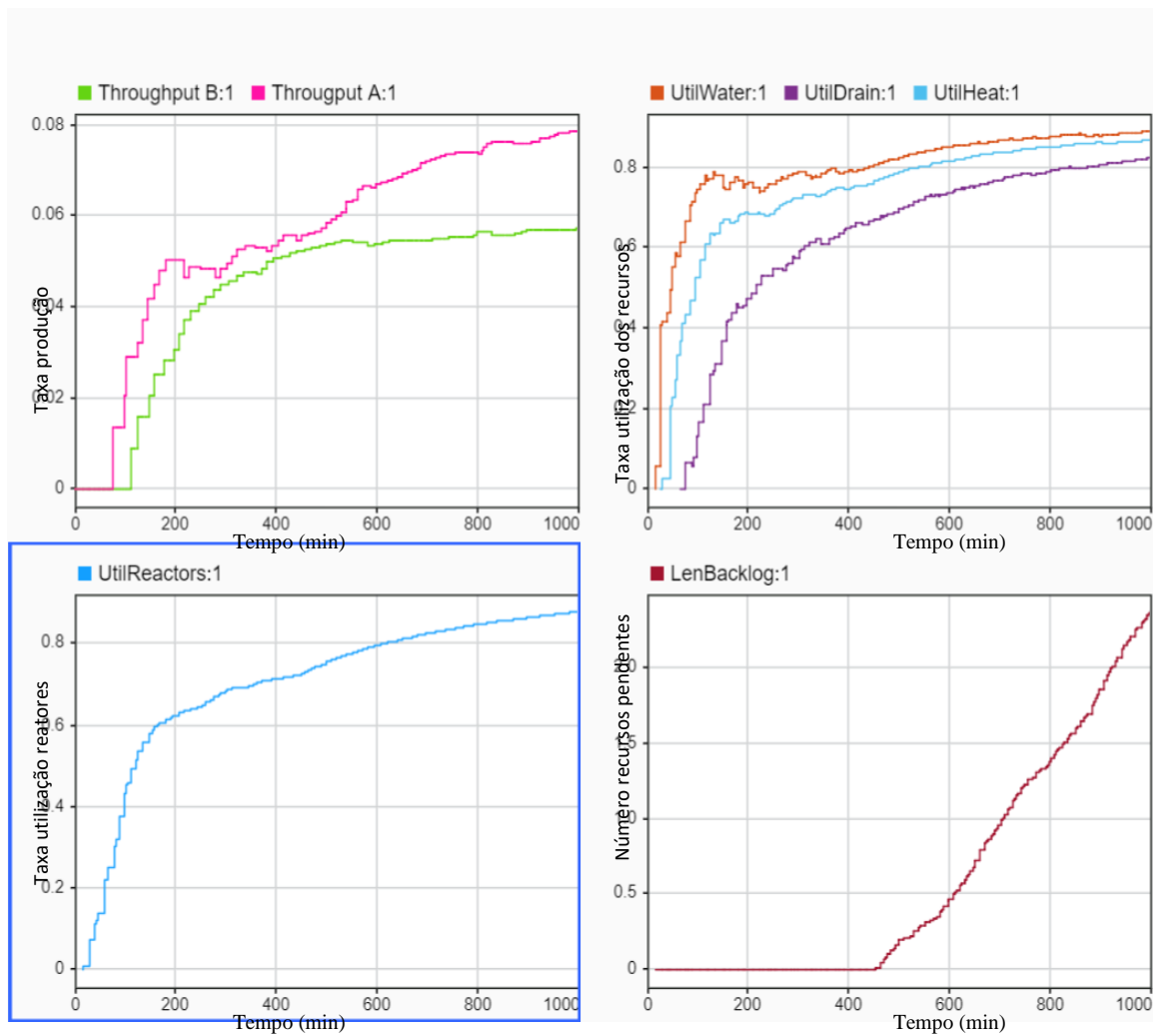


Figura 15 - Cenário 2: incerteza na produção de A

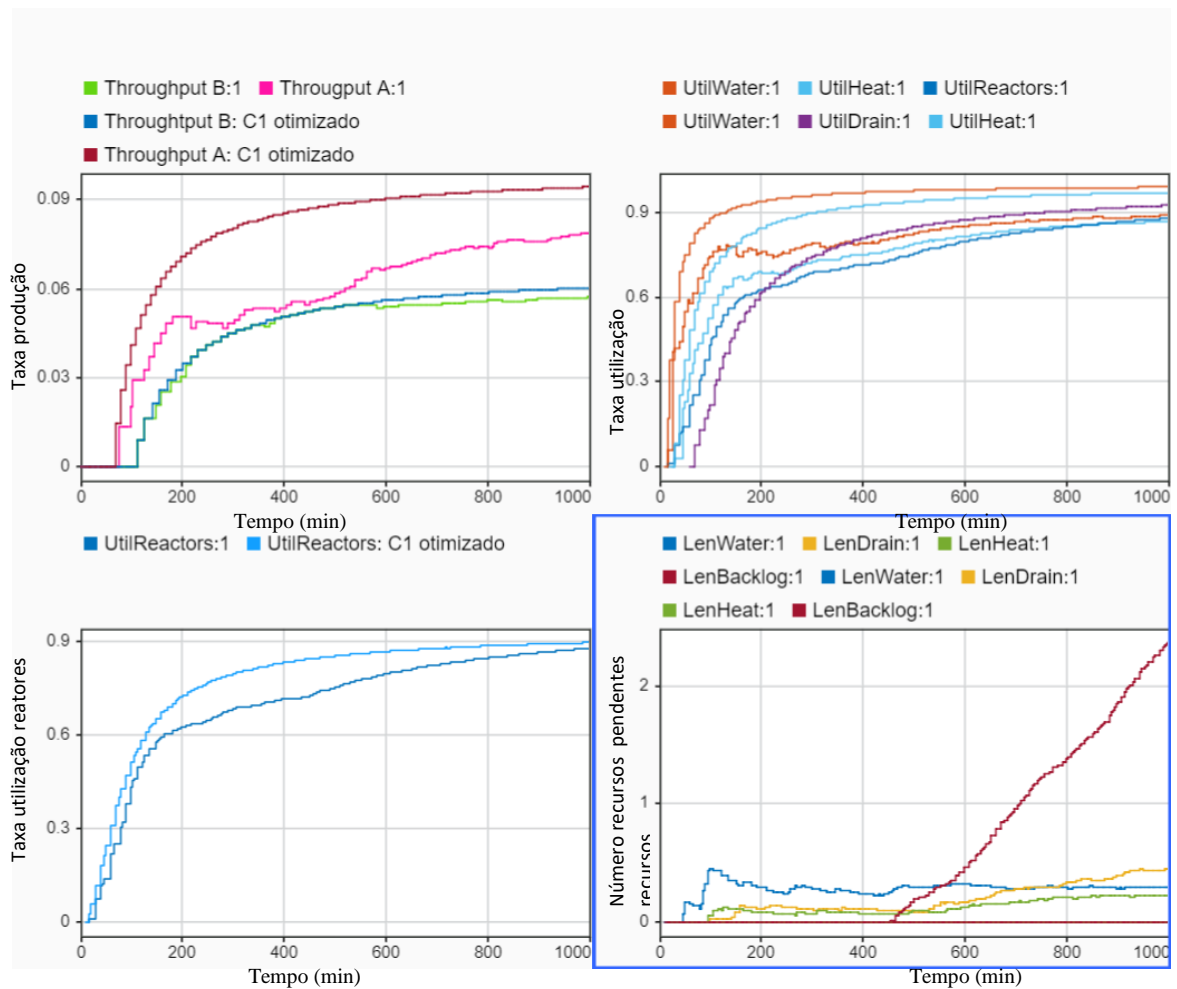


Figura 16 - Cenário 2: Comparação da solução otimizada com a incerteza na produção de A

Observa-se, pela análise da figura 16 que a produção de A e B é mais baixa, em especial a produção de A, passando de um valor para a taxa de produção superior a 0,09 para um valor de 0,08. A taxa de utilização dos recursos também é mais baixa e há um aumento das ordens pendentes, para 2 unidades, aos 1000 min, o que revela alguns problemas de robustez da solução otimizada, pois quando se está perante uma solução robusta, a introdução de incerteza nas variáveis não deveria conduzir a um pior cenário para a variável ordens pendentes.

Na fase seguinte foi aplicada a incerteza ao tempo de intergeração da ordem de produção B, retirando a incerteza da produção A, para visualizar o impacto da incerteza em cada ordem de produção, individualmente. Foi utilizada novamente uma distribuição exponencial.

Os resultados obtidos ao introduzir incerteza apenas na ordem de produção B não diferem muito daqueles obtidos na situação anterior, onde apenas introduzimos incerteza na produção A. Observa-se que a produção de A e B é mais baixa, em especial para B, que passa de um valor para a taxa de produção de 0,06 para 0,045. A taxa de utilização dos recursos também é mais baixa e há 2 ordens de produção pendentes, aos 1000 min.

Comparando a produção de A e B neste caso, com a situação otimizada, obtêm-se os resultados apresentados na figura 17.

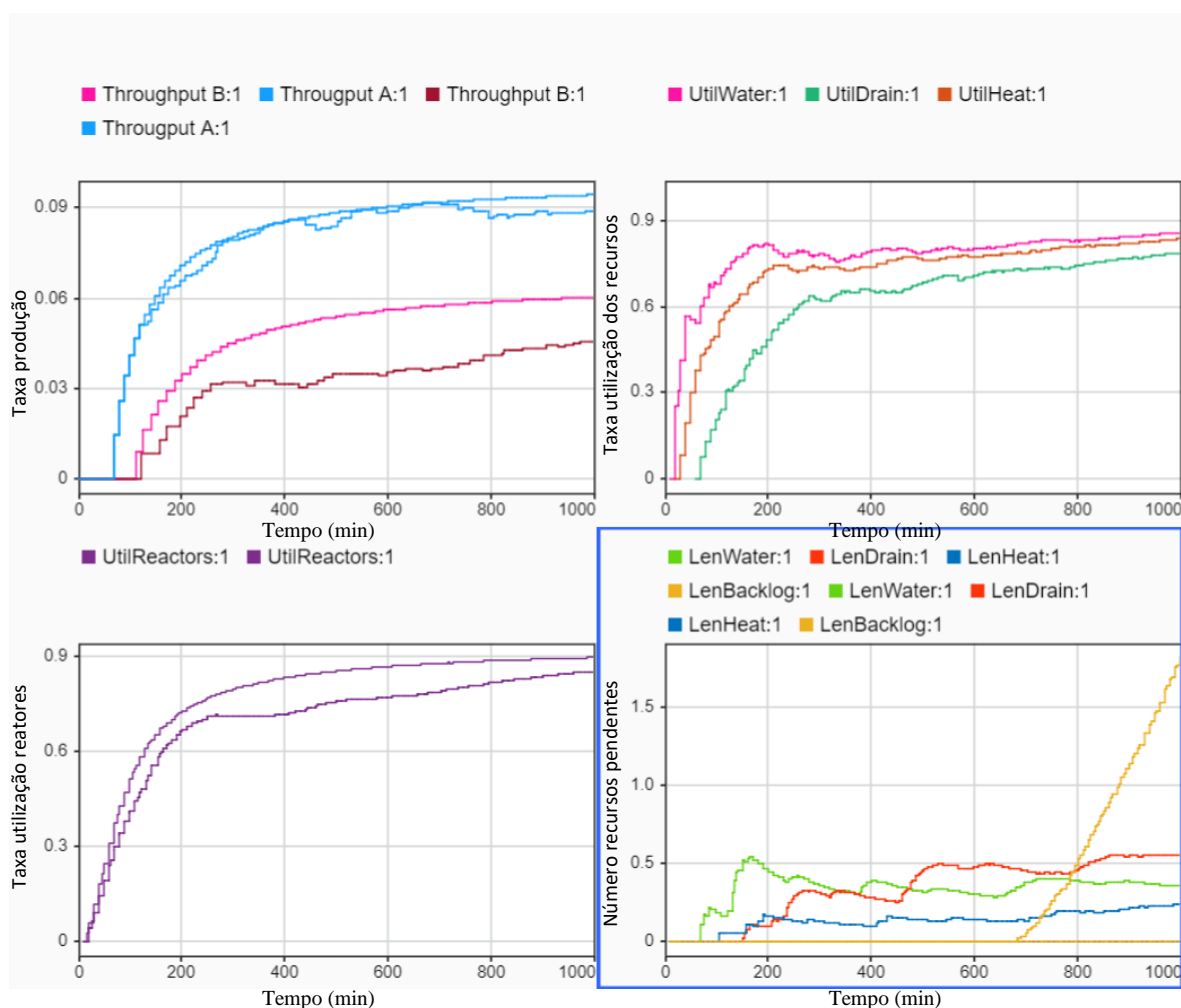


Figura 17 - Cenário 2: Comparação da situação otimizada com incerteza na produção de B

Finalmente, vamos introduzir simultaneamente incerteza em ambas os ordens de produção A e B, partindo da situação otimizada e comparando de seguida com o cenário com incerteza em A e B.

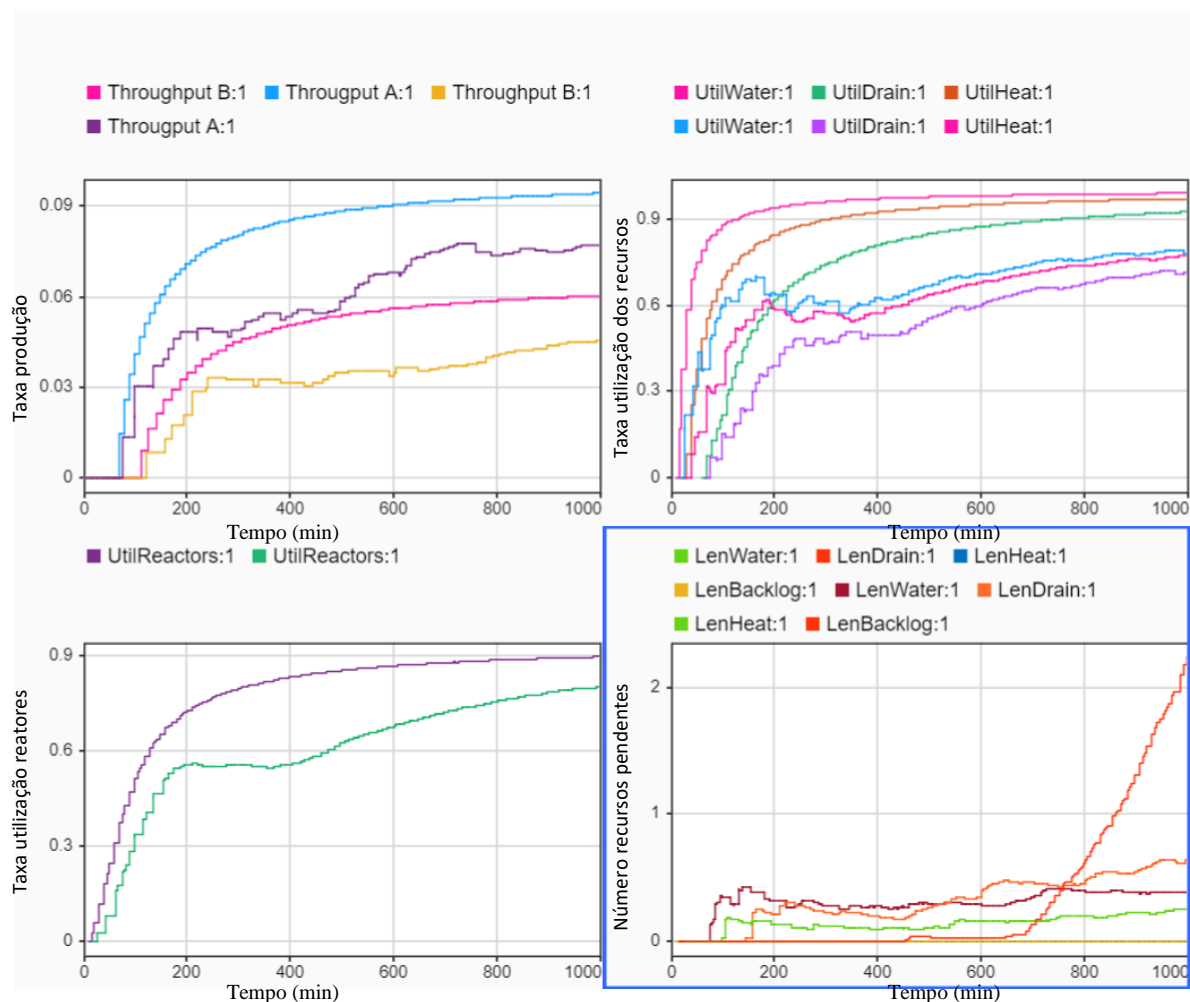


Figura 18 - Cenário 2: Comparação da situação otimizada com a incerteza nas produções de A e B

Nesta situação, por observação da figura 18, e comparativamente com as figuras 16 e 17, obtiveram-se resultados idênticos, pela que as conclusões se mantêm válidas.

Com vista a analisar agora o impacto que cada um dos parâmetros poderá ter sistema, definiu-se o cenário 3.

- **Cenário 3 – alteração dos valores das variáveis de decisão**

Neste caso, analisa-se o que acontece nas produções de A e B quando falha um dos recursos (adição de água, calor, esgoto), o que pode acontecer por motivos de avaria, por exemplo. Trata-se agora de uma avaliação da robustez não ao nível dos pressupostos considerados na construção do modelo, mas do impacto de desvios à solução otimizada.

As variáveis de decisão base são as provenientes da otimização:

- número de reatores *batch*: (R=13)
- fontes de água: (W=2)
- aquecedores: (H=4)
- efluentes: (D=2)

Em resumo apresentam-se os resultados na figura 19. Na primeira coluna apresentam-se os valores para as variáveis de decisão (Número reatores *batch*, fontes de água, aquecedores, efluentes) .

Valores Variáveis Decisão	Ordens produção pendentes	Produção lote A	Produção lote B	Total de produção
2,2,2,2	70	16	16	32
13,2,4,2	0	48	29	77
12,2,4,2	2	48	29	77
13,1,4,2	34	47	30	77
13,2,3,2	16	48	29	77
13,2,4,1	32	47	30	77

Figura 19 - Análise da influência das variáveis de decisão no número total de produções

Da análise destes resultados pode-se tirar conclusões relativamente à influência das variáveis de decisão, na produção do sistema e nas ordens pendentes. Para isso simulou-se o sistema retirando a cada recurso uma unidade, de cada vez, (a partir da solução otimizada com variáveis de decisão R=13,W=2, H=4, D=2, utilizou-se para o 1º ensaio R=12,W=2, H=4, D=2) observando o impacto no sistema. Verifica-se que o número total de produções se mantém inalterado, e por outro lado a variável de decisão que tem mais impacto no sistema são as fontes de água e os efluentes, uma vez que, quando se retira um destes elementos, o número de ordens de produção pendentes aumenta consideravelmente. Podemos também verificar, por observação das segunda e terceira linhas, que ao reduzir em 1 unidade o número de reatores, não há alteração do número de produções.

4.2. Projeto de um CSTR (Bequette, 1998; *Mathworks.com*)

A simulação de Monte Carlo é um método bastante utilizado na análise de sensibilidade, pois permite estudar a incerteza do sistema, através da geração de *inputs* aleatórios, com base em distribuições de probabilidade. Com os dados gerados, simula-se o modelo iterativamente, obtendo um conjunto de *outputs* que pode ser representado por distribuições de probabilidade e tirar conclusões importantes relativamente ao impacto que os parâmetros têm no modelo. O *software Simulink* tem uma ferramenta adicional “*Simulink Design Optimization*”, que se revelou um poderoso auxiliar na análise da simulação de Monte Carlo. Esta ferramenta inclui módulos de otimização e análise de sensibilidade, utilizados neste exemplo. No módulo de otimização, o método aplicado é o método dos gradientes descendentes, sendo o algoritmo de programação quadrática sequencial.

O modelo em estudo para a análise de sensibilidade é o projeto de um CSTR (reator cilíndrico, contínuo e perfeitamente agitado) (Bequette, 1998; *Mathworks.com*), minimizando quer a variação na concentração do produto final (através da concentração residual do reagente A), quer o custo de produção (minimizando a temperatura média do refrigerador). Neste reator dá-se uma reação exotérmica de 1ª ordem, irreversível, onde o reagente A é convertido no reagente B. Na base do reator existe um refrigerador, sendo que este modelo é não adiabático. Seguindo os princípios básicos de conservação de massa e energia, os balanços mássico e energético podem ser escritos por:

$$\text{Balanço mássico} \quad \frac{dC_A}{dt} = \frac{F}{A \cdot h} (C_{feed} - C_A) - r * C_A$$

$$\text{Balanço energético} \quad \frac{dT}{dt} = \frac{F}{A \cdot h} (T_{feed} - T) - \frac{H}{c_p \rho} r - \frac{U}{c_p \rho * h} (T - T_{cool}), \text{ sendo } r = k_0 * e^{\frac{-E}{R \cdot T}}$$

sendo:

C_A e C_{feed} - concentrações de A no reator e na alimentação (kg.mol/m³)

T , T_{feed} e T_{cool} - Temperatura no CSTR, na alimentação e no refrigerador (K)

F e ρ - Caudal volumétrico (m³/h) e densidade do produto no reator (1/m³)

h e A - altura (m) e área de seção reta do CSTR (m²)

k_0 - fator não térmico pré-exponencial $A \rightarrow B$ (1/h)

E e H - Energia de ativação e calor da reação para $A \rightarrow B$ (kcal/kgmol)

R - Constante Boltzman (kcal/(kg.mol*K))

C_p e U - Capacidade calorífica (kcal/K) e coeficiente transferência de calor (kcal/(m²*K*h))

Neste projeto, as variáveis a definir são a área de seção reta e a altura do reator. Os parâmetros do projeto são a concentração inicial da alimentação e a temperatura da alimentação. A função objetivo “ $\text{design.F} = \text{costConc} + \text{costCoolant}$ ”, que se pretende minimizar, é o custo total do projeto do reator e inclui o custo relativo à concentração do produto final, e o custo relativo ao refrigerador (Anexo B).

Apresenta-se de seguida a nomenclatura utilizada no exemplo do Projeto de um CSTR:

A – área de seção reta

$ConcVar$ – Variação da concentração residual

$CoolMean$ – Temperatura média do refrigerador

$FeedCon0$ – Concentração da alimentação

$FeedTemp0$ – Temperatura da alimentação

h – altura do reator

A figura 20 mostra a representação esquemática do exemplo do Projeto de um CSTR, em ambiente *Simulink*.

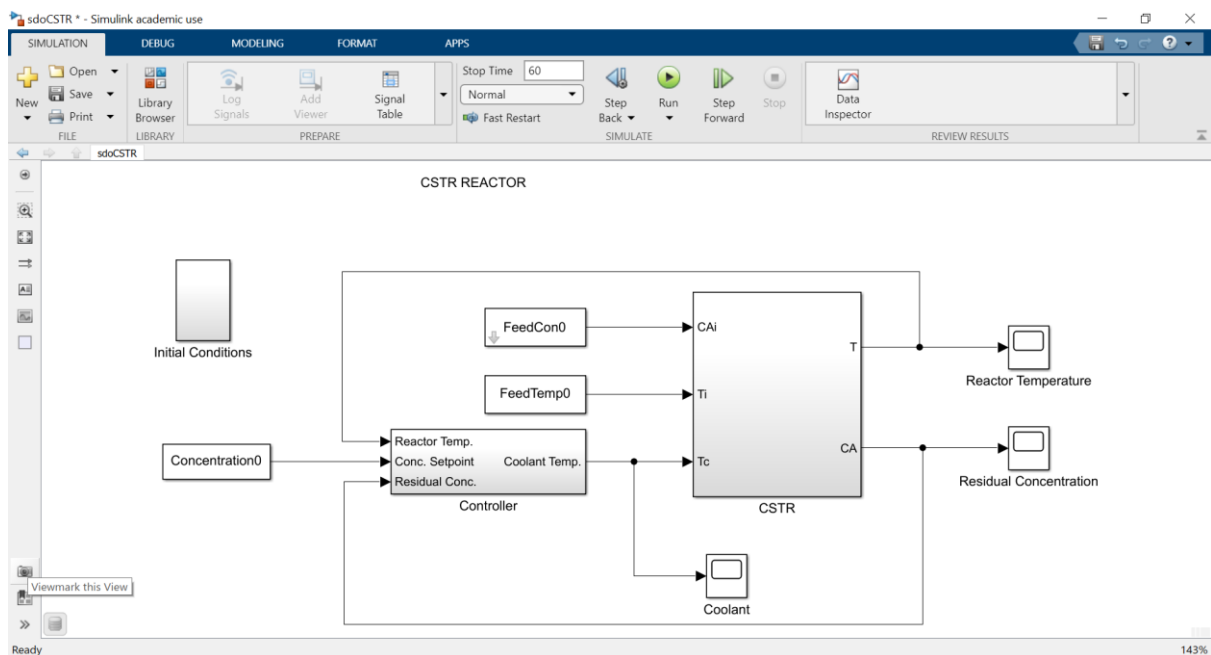


Figura 20 - Projeto de um CSTR em ambiente Simulink (Bequette, 1998; Mathworks.com-)

A metodologia utilizada no Projeto de um CSTR

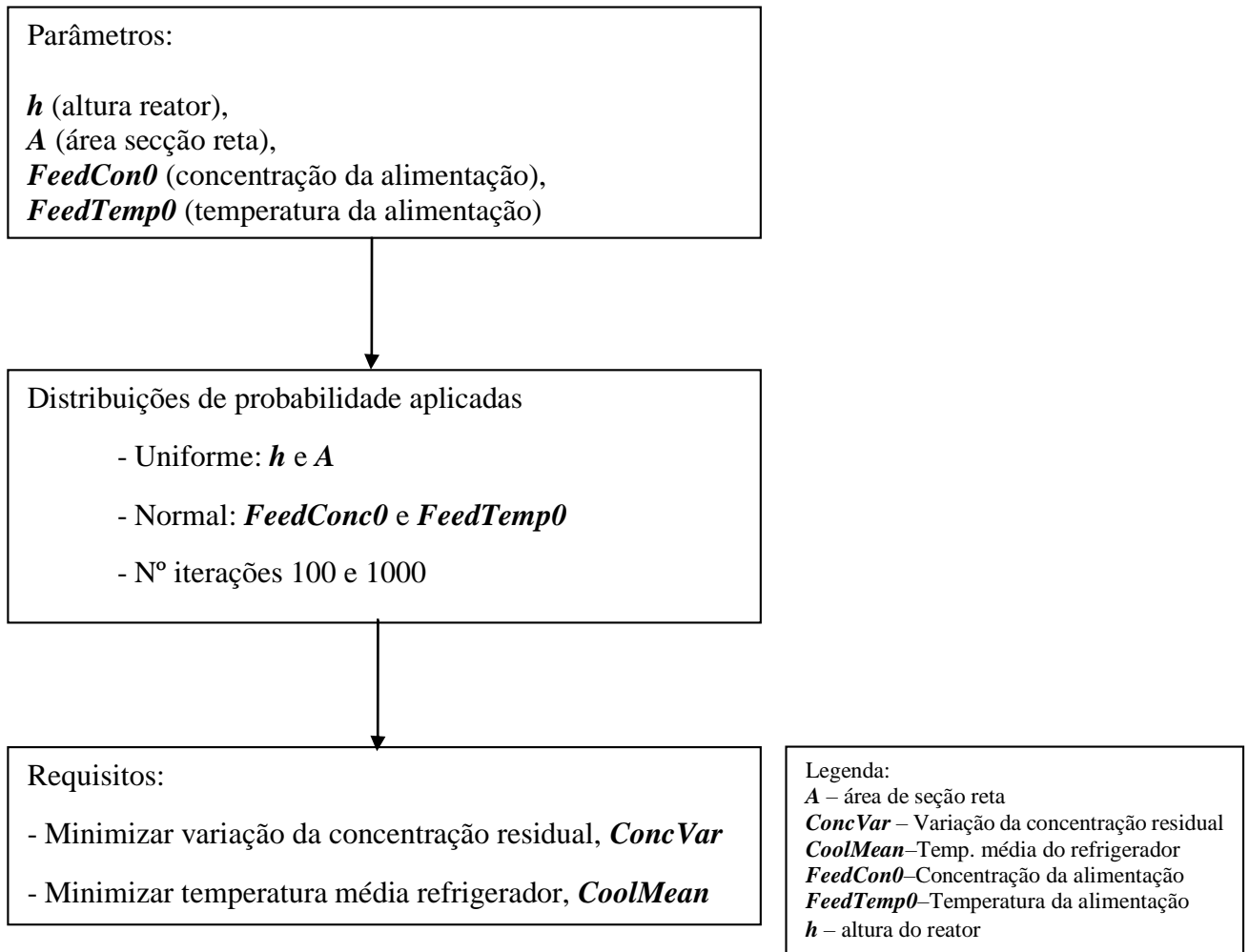


Figura 21 - Metodologia para projeto de um CSTR

4.2.1. Implementação e tratamento de dados

Seguindo a metodologia apresentada na figura 21, aplicam-se distribuições de probabilidade para caracterizar os parâmetros do modelo. Para a altura do reator, h , e para a área, A , aplicou-se uma distribuição uniforme, uma vez que assumimos que nos intervalos definidos, a probabilidade de ocorrência de um determinado resultado é sempre a mesma. Nos casos de $FeedCon0$ (concentração da alimentação) e $FeedTemp0$ (temperatura da alimentação) aplicou-se uma distribuição normal, uma vez que é a mais usualmente utilizada, caracterizada por uma média e desvio padrão, adequando-se a situações em que a função probabilidade é uma curva simétrica em torno do seu ponto médio.

Com a aplicação destas distribuições geram-se amostras aleatórias que vão ser usadas para implementar a simulação de Monte Carlo.

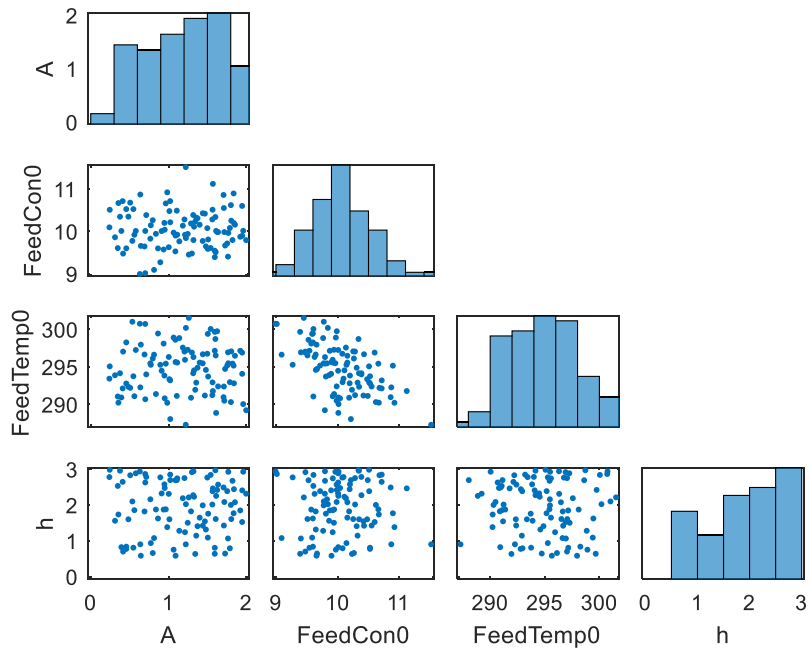


Figura 22 - Correlação entre os parâmetros do projeto

Cada um dos pontos dos gráficos da figura 22 representa uma linha da tabela do conjunto de dados, sendo cada linha simultaneamente mostrada em todos os gráficos.

A figura 22 mostra como estão relacionados os parâmetros do projeto, sendo evidenciada a distribuição de probabilidades escolhida para cada um deles, tal como apresentado no esquema da figura 21.

Para cada um dos conjuntos de valores dos parâmetros (A , h , $FeedCon0$, $FeedTemp0$), obtém-se, através do modelo, um valor para os requisitos $ConcVar$ e $CoolMean$. No final das iterações concluídas, obtém-se um conjunto de dados que poderá ser analisado de seguida.

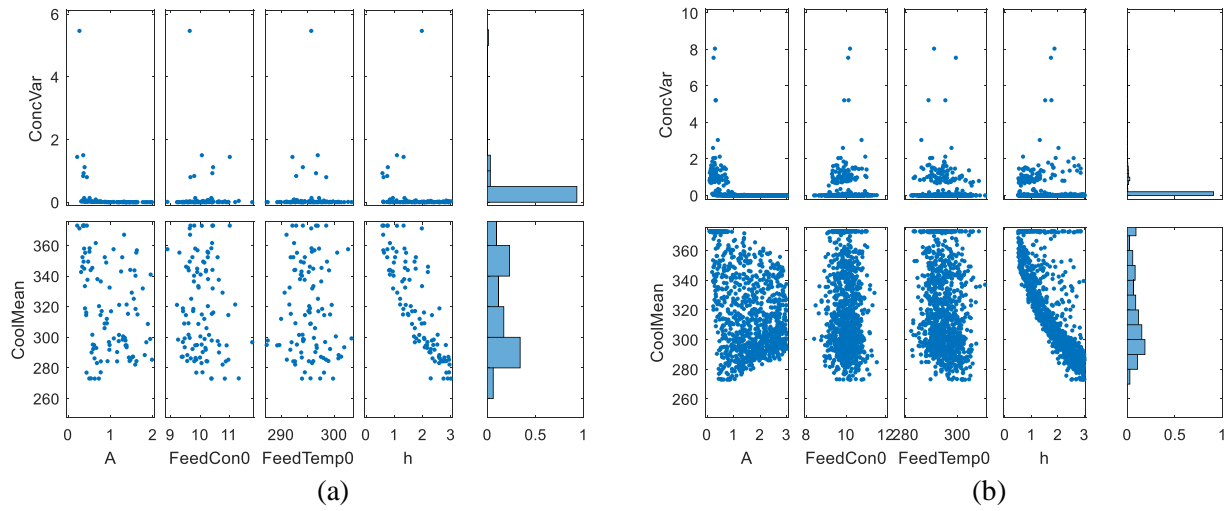


Figura 23 - Gráfico de dispersão ((a) 100 iterações (b) 1000 iterações).

A figura 23 pretende mostrar, através de gráficos de dispersão, a forma como cada valor dos parâmetros (h , A , $FeedCon0$, $FeedTemp0$), está relacionado com o valor dos requisitos. Assim, é possível concluir, pela análise da figura 23, que a altura do reator, h , aumenta à medida que a temperatura média do refrigerador, $CoolMean$ diminui. Por outro lado, podemos verificar que a relação de A com $ConcVar$ e $CoolMean$ não é explícita.

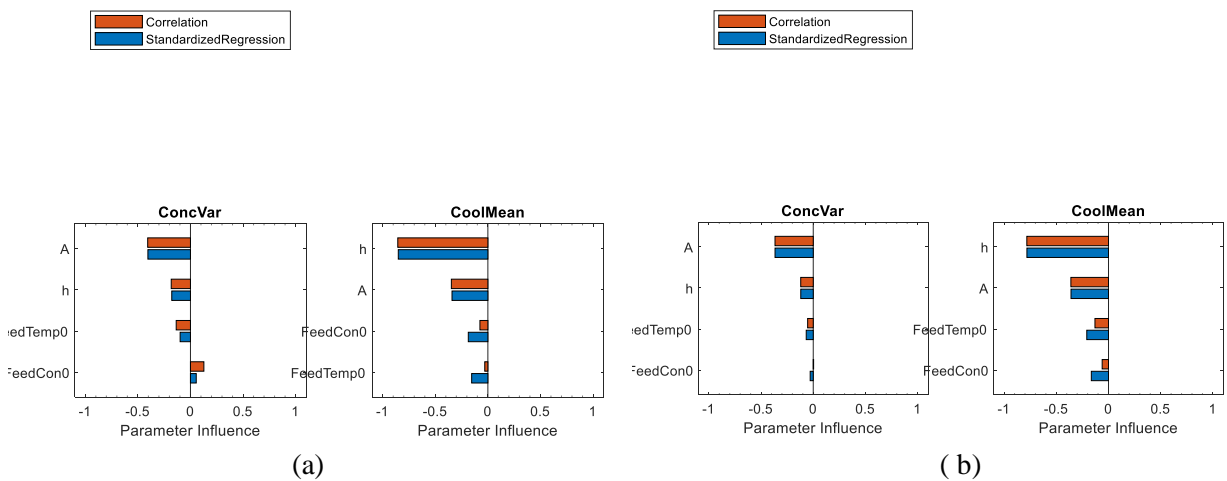


Figura 24 - Gráfico de Tornado ((a) 100 iterações (b) 1000 iterações).

A análise dos resultados através do Gráfico de Tornado é ilustrada na figura 24, sendo que o gráfico de Tornado mostra a influência de cada parâmetro (A , h , $FeedCon0$, $FeedTemp0$) nos requisitos em estudo ($ConcVar$ e $CoolMean$). Da análise destes gráficos é possível concluir que a altura do reator, h é o parâmetro que tem maior influência em $CoolMean$, enquanto que a área se seção reta do reator, A é o parâmetro que tem maior influência em $ConcVar$. É possível ainda afirmar que $FeedCon0$ e $FeedTemp0$ têm pouca influência em $CoolMean$ e $ConcVar$.

Com esta análise a principal conclusão a que é possível chegar para o projeto do reator, é de que a escolha ideal seria um valor elevado para a altura do reator, h e para a área da seção reta, A pois é a solução que minimiza os requisitos $CoolMean$ e $ConcVar$.

Uma vez feita a análise de sensibilidade, é possível obter a estimativa inicial para o problema de otimização, cujo objetivo é encontrar os valores ótimos para as variáveis A e h .

Para obter esta estimativa inicial, ordenam-se os resultados das iterações por ordem decrescente (relativamente à variável h) e encontra-se a linha que tem os menores valores para $ConcVar$ e $CoolMean$. Os valores dos parâmetros usados como estimativa inicial para a otimização apresentam-se na tabela 1.

Tabela 1 - Valores dos parâmetros como estimativa inicial para otimização

Parâmetro	Valor
A	1,58 m ²
FeedCon0	10,76 kg.mol/m ³
FeedTemp0	292,66 K
h	2,95 m

4.2.2 Análise de resultados

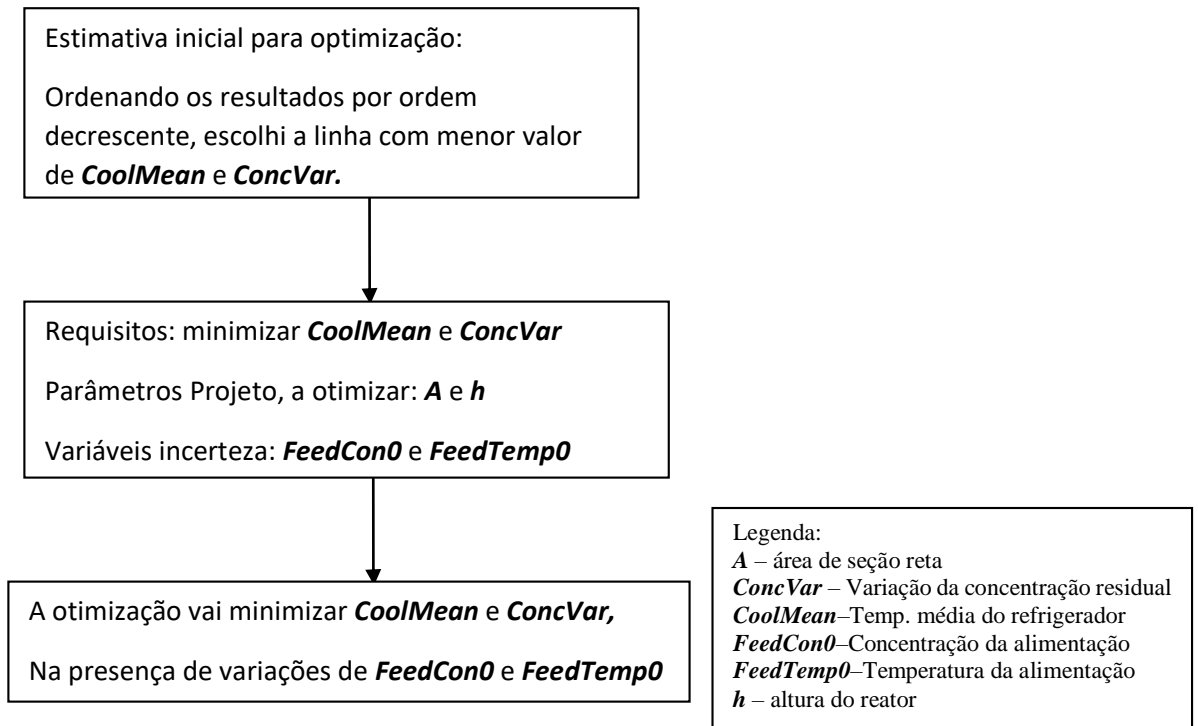


Figura 25 - Metodologia para o problema de otimização

Seguindo o procedimento descrito na figura 25, é possível executar a ferramenta “*Simulink Design Optimization*”, para a componente da otimização. O método de otimização é o dos gradientes descendentes, e o algoritmo utilizado é de programação quadrática sequencial. A tolerância definida para os parâmetros e para as restrições é de 0,001.

O resultado gráfico, após otimização, é apresentado na figura abaixo.

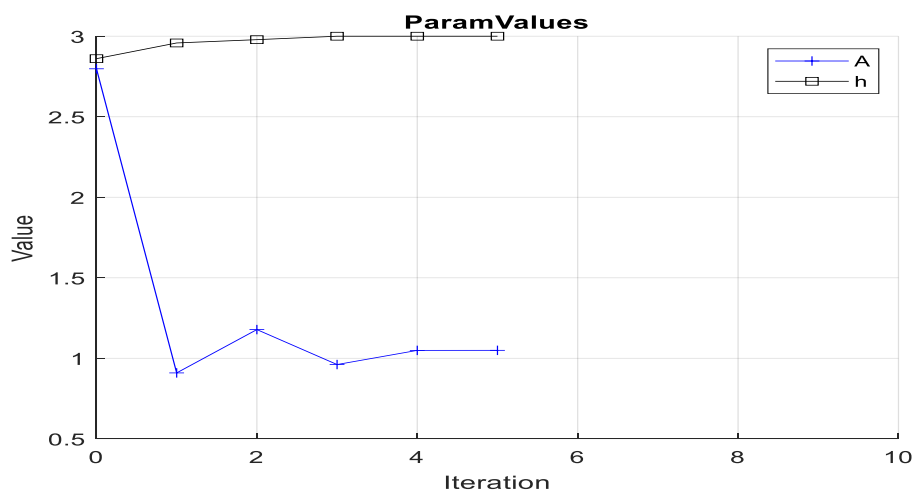


Figura 26 - Resultados da otimização

Finalmente, obtêm-se os resultados ótimos, sendo o *output* do sistema apresentado na tabela 2.

Tabela 2 - Valores dos parâmetros obtidos após otimização

Parâmetro	Valor
A	1,0494 m ²
FeedCon0	10 kg.mol/m ³
FeedTemp0	304,1371 K
h	3 m
Concentration0	2 kg.mol/m ³
deltaH	11,92 kcal/(kg.mol)
E_div_R	5,9636 e3 kcal/(kg.mol)
F	1 m ³ /h
Feednoise power	1x10 ⁻³
U	0,6 kcal/(m ² .K.h)
K0	34930800 1/h
ConcVar	0,0798
Coolmean	1,4154e3

A análise destes resultados, e por observação da figura 26, é evidenciada a mais-valia existente pela aplicação da análise de sensibilidade, implementada pela técnica de Monte Carlo. O uso desta técnica permitiu obter valores para a estimativa inicial, a partir dos quais se aplicou a técnica de otimização, que convergiu em apenas 5 iterações. Foi assim possível atingir rapidamente a solução ótima para este problema.

Podemos então afirmar, que o projeto ótimo do CSTR, é um reator com uma altura, *h* de 3 m e uma área de seção reta, *A* de 1,05 m².

5. Conclusão

O trabalho desenvolvido nesta dissertação foi realizado com base num *software* bastante utilizado a nível académico: *Matlab/Simulink*. Apesar dos casos de estudo utilizados se terem baseado em projetos já existentes, este trabalho permitiu comprovar as enormes potencialidades que uma ferramenta destas proporciona ao mundo académico e profissional, mesmo sendo necessária a integração de algumas ferramentas adicionais para o efeito. O fato da sua programação ser baseada em programação por blocos constitui uma grande vantagem. Trata-se de uma linguagem poderosa, flexível e bem documentada.

Esta dissertação conduziu à oportunidade de analisar a aplicação de otimização determinística e simulação computacional, em sistemas de produção descontínuos, com o intuito de avaliar e melhorar o funcionamento das atividades e tarefas. Através do estudo de um sistema de produção descontínuo, em ambiente *Simulink*, foi possível estabelecer o número ótimo para as variáveis do projeto, melhorando assim a capacidade do sistema na utilização de recursos partilhados. Num segundo caso de estudo, estudou-se o projeto de um reator CSTR, onde através de uma simulação de Monte Carlo, foi possível obter as estimativas iniciais, com as quais se implementou uma otimização, finda a qual foi possível chegar à solução ótima para as variáveis de projeto do sistema, a altura e a área de secção reta. A aplicação da metodologia integrada da técnica de Monte Carlo com a otimização, aplicada no projeto de um reator CSTR permitiu atingir os resultados ótimos com um reduzido número de iterações, o que é uma mais-valia, não só no caso em estudo, como seguramente na sua aplicação em situações industriais de maior complexidade.

Por outro lado, a utilização de uma metodologia integrada de otimização/simulação (aplicada no caso estudo 2), comparativamente com uma metodologia de otimização determinística (aplicada no caso de estudo 1), é vantajosa pois ao trabalhar na presença de variabilidade e incerteza, conseguimos aproximar-nos da realidade dos sistemas.

Estes dois casos de estudo, bem como todo o trabalho desenvolvido com as ferramentas auxiliares do *Simulink*, permitem perceber o alcance da utilidade deste *software*, na sua aplicação a modelos mais complexos, no âmbito dos sistemas de produção e no apoio à logística destes sistemas.

6. Referências bibliográficas

- Amaran, S., Sahinidis, N., Sharda, B., & Bury, S. (2016). Simulation optimization: A review of algorithms and applications. *Annals of Operations Research*, 240(1), 351-351–380. Business Source Complete. <https://doi.org/10.1007/s10479-015-2019-x>
- Amaran, S., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2014). Simulation optimization: A review of algorithms and applications. *4OR-A QUARTERLY JOURNAL OF OPERATIONS RESEARCH*, 12(4), 301-301–333. Science Citation Index. <https://doi.org/10.1007/s10288-014-0275-2>
- Ammeri, A. (2011). A COMPREHENSIVE LITTERATURE REVIEW OF MONO-OBJECTIVE SIMULATION OPTIMIZATION METHODS. *Advances in Production Engineering & Management*, 6(4), 291-291–302. Academic Search Complete.
- Aqlan, F., & Lam, S. S. (2016). Supply chain optimization under risk and uncertainty: A case study for high-end server manufacturing. *Computers & Industrial Engineering*, 93, 78–87. <https://doi.org/10.1016/j.cie.2015.12.025>
- Banks, J. (1998). *Handbook of Simulation—Principles, Methodology, Advances, Applications, and Practice*. Georgis Institute of Technology.
- Bequette, B. W. (1998). *Process Dynamics: Modeling, Analysis and Simulation* (1 st). Prentice Hall.
- Carson, Y., & Maria, A. (1997). Simulation optimization: Methods and applications. *Proceedings of the 29th Conference on Winter Simulation - WSC '97*, 118–126. <https://doi.org/10.1145/268437.268460>
- Castro, J. A. (1995). *Sistemas e Processos Químicos II*. Dep. Engenharia Química - UC.
- Edgar, T., & Himmelblau, D. (2001). *Optimization of Chemical Process*. MacGraw-Hill.

- Figueira, G. (2019). Simulação-Otimização: Porque e como combiná-las? *Pesquisa Operacional para o Desenvolvimento*, 11(1), 1–5.
<https://doi.org/10.4322/PODes.2019.001>
- Figueira, G., & Almada-Lobo, B. (2014a). Hybrid simulation–optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46, 118–134.
<https://doi.org/10.1016/j.simpat.2014.03.007>
- Frazzon, E. M., Albrecht, A., & Hurtado, P. A. (2016). Simulation-based optimization for the integrated scheduling of production and logistic systems. *IFAC-PapersOnLine*, 49(12), 1050–1055. <https://doi.org/10.1016/j.ifacol.2016.07.581>
- Fu, M. (2002). Optimization for simulation: Theory vs. Practice. *INFORMS JOURNAL ON COMPUTING*, 14(3), 192–192–215. Science Citation Index.
- Fu, M. C. (1994). Optimization via simulation: A review. *Annals of Operations Research*, 53(1–4), 199–199–247. Business Source Complete.
<https://doi.org/10.1007/BF02136830>
- Grossmann, I. E., Apap, R. M., Calfa, B. A., García-Herreros, P., & Zhang, Q. (2016). Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. *Computers & Chemical Engineering*, 91, 3–14.
<https://doi.org/10.1016/j.compchemeng.2016.03.002>
- Grossmann, I. E., Calfa, B. A., & Garcia-Herreros, P. (2014). Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. *Computers & Chemical Engineering*, 70, 22–34.
<https://doi.org/10.1016/j.compchemeng.2013.12.013>
- Marques, C. M., Moniz, S., & de Sousa, J. P. (2018). Strategic decision-making in the pharmaceutical industry: A unified decision-making framework. *Computers & Chemical Engineering*, 119, 171–189.
<https://doi.org/10.1016/j.compchemeng.2018.09.010>

- Marques, C. M., Moniz, S., de Sousa, J. P., & Barbosa-Póvoa, A. P. (2017). 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*, *106*, 796–813. <https://doi.org/10.1016/j.compchemeng.2017.04.008>
- Marques, C. M., Moniz, S., de Sousa, J. P., Barbosa-Povoá, A. P., & Reklaitis, G. (2020). Decision-support challenges in the chemical-pharmaceutical industry: Findings and future research directions. *Computers and Chemical Engineering*, *134*. ScienceDirect. <https://doi.org/10.1016/j.compchemeng.2019.106672>
- Mathworks.com*. (sem data). Mathworks.com. <https://www.mathworks.com/>
- Méndez, C. (2006). *State of art@j.compchemeng.2006.02.008.pdf*.
- Papageorgiou, L. G. (2009). Supply chain optimisation for the process industries: Advances and opportunities. *Computers & Chemical Engineering*, *33*(12), 1931–1938. <https://doi.org/10.1016/j.compchemeng.2009.06.014>
- Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-the-art and opportunities. *Computers and Chemical Engineering*, *28*(6), 971-971–983. ScienceDirect. <https://doi.org/10.1016/j.compchemeng.2003.09.017>
- SeOptimizationForBatchProductionProcess.m*. (sem data).
- Trigueiro de Sousa Junior, W., Barra Montevechi, J. A., de Carvalho Miranda, R., & Teberga Campos, A. (2019). Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review. *Computers & Industrial Engineering*, *128*, 526–540. <https://doi.org/10.1016/j.cie.2018.12.073>

Velez, S., & Maravelias, C. T. (2013). Multiple and nonuniform time grids in discrete-time MIP models for chemical production scheduling. *Computers & Chemical Engineering*, 53, 70–85. <https://doi.org/10.1016/j.compchemeng.2013.01.014>

7. Anexo A – Caso de estudo 1: otimização de recursos partilhados

Algoritmo Genético

```
function finalResult = seExampleRunOptimizationForBatchProductionProcess()
% Find optimal resource capacities of a chemical production system.
% Apply Genetic Algorithm solver of MATLAB Global Optimization Toolbox and
% SimEvents Discrete-Event Simulation to find optimal resource capacities
% of a chemical production

% The genetic algorithm solves optimization problems by repeatedly
% modifying a population of individual points. Due to its random nature,
% the genetic algorithm improves your chances of finding a global solution.
% It does not require the functions to be differentiable or continuous.

% Decision variables are the
% # of batch reactors
% # of water tanks
% # of heaters
% # of drains

% These are set by the genetic algorithm as it runs multiple
% simulations of the BatchProductionOptimization model via the variable
% ResourceCapacity.

% Open a parallel pool
pool = parpool;

% Setting plot and parallelization options
opts = gaoptimset(...
    'PlotFcns', @gaplotbestf, ...
    'Generations', 25, ...
    'StallGenLimit', 10, ...
```

```

    'UseParallel', 'always');

% Loading the model to run on all parallel workers
pctRunOnAll(load_system("seExampleBatchProduction"));

% Lower bound of decision variables
lb = [1 1 1 1];

% Upper bound of decision variables
ub = [20 10 10 10];

% Integer constraints: If the third was not an integer, it would be [1, 2,
% 4]
IntCon = [1 2 3 4];

% Track time spent for optimization
tic;

% Execute genetic algorithm solver
[finalResult, ~, ~] = ga(@productionCost, 4, [], [], [], [], ...
    lb, ub, [], IntCon, opts);
toc;

% Shut down the parallel pool
delete(pool);

end

% Cost function that assign different values to the decision variables in
% the model
function obj = productionCost(ResourceCapacity)
% Assigns costs to the values of ResourceCapacity, which correspond
% to [batch reactors, water tanks, heaters, drains]
cost = [1000 300 200 100] * ResourceCapacity';

```

```

% Assigns variables to the base workspace for simulation
assignin('base', 'ResourceCapacity', ResourceCapacity);

% Simulation of the model and assigns output to the variable z
if isempty(find_system('type', 'block_diagram', ...
    'Name', 'seExampleBatchProduction'))
    load_system('seExampleBatchProduction');
end

set_param('seExampleBatchProduction/ConfigResource', 'NumBatchReactor', ...
    num2str(ResourceCapacity(1)), 'NumWater', ...
    num2str(ResourceCapacity(2)), 'NumHeat', ...
    num2str(ResourceCapacity(3)), 'NumDrain', ...
    num2str(ResourceCapacity(4)));

[~, ~, z] = sim('seExampleBatchProduction');

% Takes the last value of the logged data as the final backlog
% value
backlog = z(end);

% Calculates the objective function, based on the backlog and costs
obj = backlog*10000 + cost;

end

```


8. Anexo B – Caso de estudo 2: Projeto de um CSTR

```
%% Design Optimization with Uncertain Variables (Code)
%
% This example shows how to optimize a design when there are uncertain
% variables. You optimize the dimensions of a Continuously Stirred Tank
% Reactor (CSTR) to minimize product concentration variation and production
% cost in case of varying, or uncertain, feed stock.
%% Continuously Stirred Tank Reactor (CSTR) Model
%
% In this example, you use the following two-state CSTR model, which uses
% basic accounting and energy conservation principles:
%
% 
$$\frac{d C_A}{dt} = \frac{F}{A * h}(C_{\text{feed}} - C_A) - r * C_A$$

%
% 
$$\frac{dT}{dt} = \frac{F}{A * h}(T_{\text{feed}} - T) - \frac{H}{c_p \rho} r - \frac{U}{c_p \rho h}(T - T_{\text{cool}})$$

%
% 
$$r = k_0 * e^{\frac{-E}{R * T}}$$

%
% *  $C_A$ , and  $C_{\text{feed}}$  - Concentrations of A in the CSTR and in the
% feed [kgmol/m3]
%
% *  $T$ ,  $T_{\text{feed}}$ , and  $T_{\text{cool}}$  - CSTR, feed, and coolant temperatures [K]
%
% *  $F$  and  $\rho$  - Volumetric flow rate [m3/h] and the density of
% the material in the CSTR [1/m3]
%
% *  $h$  and  $A$  - Height [m] and heated cross-sectional area [m2] of
% the CSTR.
%
% *  $k_0$  - Pre-exponential non-thermal factor for reaction
%  $A \rightarrow B$  [1/h]
%
% *  $E$  and  $H$  - Activation energy and heat of reaction for
%  $A \rightarrow B$  [kcal/kgmol]
%
% *  $R$  - Boltzmann's gas constant [kcal/(kgmol * K)]
%
% *  $c_p$  and  $U$  - Heat capacity [kcal/K] and heat transfer
% coefficients [kcal/(m2 * K * h)]
%
% Open the Simulink model.
open_system('sdoCSTR');
%%
% The model includes a cascaded PID controller in the |Controller|
% subsystem. The controller regulates the reactor temperature,  $T$ , and reactor residual
% concentration,  $C_A$ .
%
%% CSTR Design Problem
```

```

%
% Assume that the CSTR is cylindrical, with the coolant applied to the
% base of the cylinder. Tune the CSTR cross-sectional area, $A$, and CSTR height, $h$,
% to meet the following design goals:
%
% * Minimize the variation in residual concentration, $C_A$. Variations in the
% residual concentration negatively affect the quality of the CSTR product.
% Minimizing the variations also improves CSTR profit.
%
% * Minimize the mean coolant temperature $T_{cool}$. Heating or cooling the jacket
% coolant temperature is expensive. Minimizing the mean coolant temperature improves
% CSTR profit.
%
% The design must allow for variations in the quality of supply feed
% concentration, $C_{feed}$, and feed temperature, $T_{feed}$. The CSTR is
% fed with feed from different suppliers. The quality of the feed differs
% from supplier to supplier and also varies within each supply batch.
%
%% Specify Design Variables
%
% Select the following model parameters as design variables for optimization:
%
% * Cylinder cross-sectional area $A$
%
% * Cylinder height $h$
%
p = sdo.getParameterFromModel('sdoCSTR',{'A','h'});

%%
% Limit the cross-sectional area to a range of [1 2] m^2.
p(1).Minimum = 1;
p(1).Maximum = 2;
%%
% Limit the height to a range of [1 3] m.
p(2).Minimum = 1;
p(2).Maximum = 3;

%% Specify Uncertain Variables
%
% Select the feed concentration and feed temperature as uncertain
% variables. You evaluate the design using different values of feed
% temperature and concentration.
pUnc = sdo.getParameterFromModel('sdoCSTR',{'FeedCon0','FeedTemp0'});

%%
% Create a parameter space for the uncertain variables. Use normal
% distributions for both variables. Specify the mean as the current
% parameter value. Specify a variance of 5% of the mean for the feed
% concentration and 1% of the mean for the temperature.
%
uSpace = sdo.ParameterSpace(pUnc);

```

```

uSpace =
setDistribution(uSpace,'FeedCon0',makedist('normal',pUnc(1).Value,0.05*pUnc(1).Value));
uSpace =
setDistribution(uSpace,'FeedTemp0',makedist('normal',pUnc(2).Value,0.01*pUnc(2).Value));

%%
% The feed concentration is inversely correlated with the feed temperature.
% Add this information to the parameter space.

%uSpace.RankCorrelation = [1 -0.6; -0.6 1];

%%
% The rank correlation matrix has a row and column for each parameter with
% the (i,j) entry specifying the correlation between the i and j
% parameters.
%
% Sample the parameter space. The scatter plot shows the correlation
% between concentration and temperature.
rng('default'); %For reproducibility
uSmpl = sdo.sample(uSpace,60);
sdo.scatterPlot(uSmpl)

%%
% Ideally you want to evaluate the design for every combination of points
% in the design and uncertain spaces, which implies 30*60 = 1800
% simulations. Each simulation takes around 0.5 sec. You can use parallel
% computing to speed up the evaluation. For this example you
% instead only use the samples that have maximum & minimum concentration
% and temperature values, reducing the evaluation time to around 1 min.
%
[~,iminC] = min(uSmpl.FeedCon0);
[~,imaxC] = max(uSmpl.FeedCon0);
[~,iminT] = min(uSmpl.FeedTemp0);
[~,imaxT] = max(uSmpl.FeedTemp0);
uSmpl = uSmpl(unique([iminC,imaxC,iminT,imaxT]) ,:);

%% Specify Design Requirements
%
% The design requirements require logging model signals. During
% optimization, the model is simulated using the current value of the
% design variables. Logged signals are used to evaluate the design
% requirements.
%
% Log the following signals:
%
% * CSTR concentration, available at the second output port of the |sdoCSTR/CSTR|
% block
%
Conc = Simulink.SimulationData.SignalLoggingInfo;
Conc.BlockPath      = 'sdoCSTR/CSTR';
Conc.OutputPortIndex = 2;
Conc.LoggingInfo.NameMode = 1;

```

```

Conc.LoggingInfo.LoggingName = 'Concentration';

%%
% * Coolant temperature, available at the first output of the
% |sdoCSTR/Controller| block
%
Coolant = Simulink.SimulationData.SignalLoggingInfo;
Coolant.BlockPath      = 'sdoCSTR/Controller';
Coolant.OutputPortIndex = 1;
Coolant.LoggingInfo.NameMode = 1;
Coolant.LoggingInfo.LoggingName = 'Coolant';
%%
% Create and configure a simulation test object to log the required
% signals.
simulator = sdo.SimulationTest('sdoCSTR');
simulator.LoggingInfo.Signals = [Conc,Coolant];

%% Create Objective/Constraint Function
%
% Create a function to evaluate the CSTR design. This function is called at
% each optimization iteration.
%
% Use an anonymous function with one argument that calls the |sdoCSTR_design| function.
%
evalDesign = @(p) sdoCSTR_design(p,simulator,pUnc,uSmpl);

%%
% The |evalDesign| function:
%
% * Has one input argument that specifies the CSTR dimensions
%
% * Returns the optimization objective value
%
% The |sdoCSTR_design| function uses a |for| loop that iterates through the
% sample values specified for the feed concentration. Within the loop, the function:
%
% * Simulates the model using the current iterate, feed concentration, and
% feed temperature values
%
% * Calculates the residual concentration variation and coolant temperature
% costs
%
% To view the objective function, type |edit sdoCSTR_design|.
%

%% Evaluate Initial Design
%
% Call the |evalDesign| function with the initial CSTR dimensions.
%
dInit = evalDesign(p)

%%

```

```

% Plot the model response for the initial design. Simulate
% the model using the sample feed concentration values. The plot shows the
% variation in the residual concentration and coolant temperature.
sdoCSTR_plotModelResponse(p,simulator,pUnc,uSmpl);

%%
% The |sdoCSTR_plotModelResponse| function plots the model response. To
% view this function, type |edit sdoCSTR_plotModelResponse|.
%

%% Optimize Design
%
% Pass the objective function and initial CSTR dimensions to
% |sdo.optimize|.
%
pOpt = sdo.optimize(evalDesign,p)

%% Evaluate Optimized Design
%
% Call the |evalDesign| function with the optimized CSTR dimensions.
%
dFinal = evalDesign(pOpt)

%%
% Plot the model response for the optimized design. Simulate
% the model using the sample feed concentration values. The optimized design reduces
% the residual concentration variation and average coolant temperature for
% different feed stocks.
%
sdoCSTR_plotModelResponse(pOpt,simulator,pUnc,uSmpl);

bdclose('sdoCSTR')

type sdoCSTR_design
function design = sdoCSTR_design(p,simulator,pUnc,smplUnc)
%SDOCSTR_DESIGN
%
% The sdoCSTR_design function is used to evaluate a CSTR design.
%
% The |p| input argument is the vector of CSTR dimensions.
%
% The |simulator| input argument is a sdo.SimulinkTest object used to
% simulate the |sdoCSTR| model and log simulation signals.
%
% The |pUnc| input argument is a vector of parameters to specify the CSTR
% input feed concentration and feed temperature. The |smplUnc| argument is
% a table of different feed concentration and temperature values.
%
% The |design| return argument contains information about the design
% evaluation that can be used by the |sdo.optimize| function to optimize
% the design.

```

```

%
% see also sdo.optimize, sdoExampleCostFunction
%
%% Model Simulations and Evaluations
%
% For each value in |smpUnc|, configure and simulate the model. Use
% the logged concentration and coolant signals to compute the design cost.
%
costConc = 0;
costCoolant = 0;
for ct=1:size(smpUnc,1)
    %Set the feed concentration and temperature values
    pUnc(1).Value = smpUnc{ct,1};
    pUnc(2).Value = smpUnc{ct,2};

    %Simulate model
    simulator.Parameters = [p; pUnc];
    simulator = sim(simulator);
    logName = get_param('sdoCSTR','SignalLoggingName');
    simLog = get(simulator.LoggedData,logName);

    %Compute Concentration cost based on the standard deviation of the
    %concentration from a nominal value.
    Sig = find(simLog,'Concentration');
    costConc = costConc+10*std(Sig.Values-2);

    %Compute coolant cost based on the mean deviation from room
    %temperature.
    Sig = find(simLog,'Coolant');
    costCoolant = costCoolant+abs(mean(Sig.Values - 294))/30;
end

%% Return Total Cost
%
% Compute the total cost as a sum of the concentration and coolant costs.
%
design.F = costConc + costCoolant;

%%
% Add the individual cost terms to the return argument. These are not used
% by the optimizer, but included for convenience.
design.costConc = costConc;
design.costCoolant = costCoolant;
end

```