



# **Abordagens híbridas de *Machine Learning*/Simulação para sistemas logísticos dinâmicos**

**Vitor Hugo Sousa Carneiro**

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: José Soeiro Ferreira

Co-orientador: João Pedro Tavares Vieira Basto

24 de Junho de 2019



# Resumo

O desenvolvimento tecnológico e o aparecimento de novas soluções nas áreas da automação e robótica possibilitam a criação de sistemas de transporte cada vez mais inteligentes. Este tipo de evolução, quando bem explorado, permite às empresas obterem enormes vantagens face aos principais competidores.

Atualmente, existe uma necessidade de criação de sistemas logísticos mais flexíveis capazes de responderem à constante dinâmica dos sistemas de produção. Assim, surge a necessidade de se delinear um comportamento eficaz ao veículo responsável pelo fluxo de materiais ao longo do chão de fábrica.

Neste contexto, esta dissertação apresenta uma abordagem híbrida que combina técnicas evolucionárias, nomeadamente o *Neuroevolution*, com técnicas de simulação baseadas em eventos discretos para o desenvolvimento de um sistema logístico dinâmico. Da integração destas técnicas resulta uma definição do movimento a ser realizado por um *Automated Guided Vehicle* (AGV), em cada momento de decisão, com o objetivo de maximizar índices de produtividade.

A abordagem parte duma avaliação de *fitness* de uma população de redes neuronais responsáveis por modelarem o comportamento de um AGV. A avaliação resulta da verificação do desempenho de cada rede neuronal no ambiente de simulação. Na sequência desta avaliação, recorre-se a um algoritmo genético para seleção e evolução dos indivíduos com melhor função de *fitness*. Este processo resulta em indivíduos com comportamentos progressivamente melhores corroborados pelos maiores valores de produtividade adjacentes. Como demonstração da flexibilidade da abordagem os resultados são avaliados segundo diferentes parâmetros de velocidade do AGV para diferentes horizontes temporais e comparados com soluções heurísticas pré-desenvolvidas.



# Abstract

The technological development and emergence of new solutions in the automation and robotic field enable the creation of increasingly smarter transportation systems. This sort of evolution, when well explored, allows companies to have vast benefits over their main competitors.

Currently, there is a need to create logistics systems that are more flexible and capable of responding to the constant dynamics of production systems. In this context emerges a necessity to define an efficient behavior to the vehicle responsible for the flow of materials along the shop floor.

This dissertation presents a hybrid approach that combines evolutionary techniques, namely Neuroevolution, with discrete event simulation techniques for the development of a dynamic logistics system. From the incorporation of these techniques, the AGV's (Automated Guided Vehicle) movement in each decision timing is defined with the purpose of maximizing the productivity index.

The approach uses a neural network to model the AGV's behavior within the simulation environment. Fitness values are obtained from the evaluation of each neural network in the simulation environment at the end of each run. Following said evaluation, a genetic algorithm is used to select and evolve the individuals with better fitness values. This process results in individuals with progressively better behavior corroborated by its superior productivity values. As a demonstration of the flexibility of this method, the results are assessed by various AGV velocity parameters for different time horizons and compared with different heuristic-based approaches.



# Agradecimentos

Em primeiro lugar, quero agradecer ao Engenheiro João Basto e ao Professor Doutor José Soeiro Ferreira pela oportunidade e por todo apoio que me facultaram ao longo da realização da dissertação. Gostaria também de agradecer ao Romão Santos e ao Narciso Caldas por todos os conteúdos que me transmitiram que se revelaram chave para que o trabalho seguisse o rumo certo.

Aproveito também para agradecer ao INESC TEC – Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência – pelo incrível ambiente de trabalho facultado.

Um especial obrigado aos meus amigos Pedro Leite, Renato Silva, Alexandra Oliveira, Francisca Pereira e José Caires por tudo o que partilhamos ao longo destes 5 anos.

À minha namorada Daniela Pires, por toda a força e motivação que me deu para ultrapassar todos os obstáculos tendo sido uma peça fundamental ao longo de toda esta etapa.

Como não poderia deixar de ser, um enorme obrigado à minha família por tudo que sempre lutou para me garantir as melhores condições. Aproveito ainda para deixar um obrigado a todas as outras pessoas que duma forma ou outra contribuíram para que esta etapa fosse realmente especial. Um sincero obrigado a todos.

Vitor Carneiro





*“Nobody who ever gave his best regretted it”*

George Halas



# Conteúdo

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introdução</b>  | <b>1</b> |
| 1.1      | Contextualização da dissertação . . . . .                        | 1        |
| 1.2      | Motivação . . . . .  | 2        |
| 1.3      | Objetivos . . . . .  | 3        |
| 1.4      | Metodologia . . . . .  | 4        |
| 1.5      | Estrutura da Dissertação . . . . .                               | 5        |
| <b>2</b> | <b>Revisão bibliográfica</b>                                     | <b>7</b> |
| 2.1      | Sistemas de Produção . . . . .                                   | 7        |
| 2.1.1    | Introdução . . . . .   | 7        |
| 2.1.2    | Taylorismo . . . . .   | 7        |
| 2.1.3    | Fordismo . . . . .   | 8        |
| 2.1.4    | Sistema de Produção Toyota . . . . .                             | 8        |
| 2.2      | Sistema de produção <i>job shop</i> . . . . .                    | 11       |
| 2.2.1    | Definição formal de um <i>job shop</i> . . . . .                 | 11       |
| 2.2.2    | Características . . . . .  | 11       |
| 2.2.3    | Métricas de performance . . . . .                                | 12       |
| 2.3      | <i>Milk Run</i> . . . . .  | 14       |
| 2.3.1    | O Sistema Logístico . . . . .                                    | 14       |
| 2.3.2    | O Sistema <i>Milk Run</i> . . . . .                              | 16       |
| 2.3.3    | <i>In-plant</i> . . . . .  | 18       |
| 2.4      | <i>Material Handling System</i> . . . . .                        | 19       |
| 2.4.1    | <i>Automated Guided Vehicle</i> . . . . .                        | 20       |
| 2.4.2    | Condições ideais de aplicação . . . . .                          | 21       |
| 2.4.3    | <i>Design</i> do sistema AGV . . . . .                           | 21       |
| 2.4.4    | Controlo do Sistema AGV . . . . .                                | 21       |
| 2.5      | Otimização do <i>Material Handling System</i> . . . . .          | 22       |
| 2.5.1    | Otimização de MHS e <i>design</i> do <i>layout</i> . . . . .     | 22       |
| 2.6      | <i>Machine Learning</i> . . . . .                                | 24       |
| 2.6.1    | <i>Reinforcement Learning</i> . . . . .                          | 25       |
| 2.6.2    | <i>Reinforcement Learning</i> - Características . . . . .        | 25       |
| 2.6.3    | Técnicas de <i>Reinforcement Learning</i> . . . . .              | 27       |
| 2.6.4    | <i>Policy Gradients</i> . . . . .                                | 31       |
| 2.7      | Simulação de Eventos Discretos . . . . .                         | 33       |
| 2.7.1    | Vantagens da Simulação Baseada em Eventos Discretos . . . . .    | 35       |
| 2.7.2    | Desvantagens da Simulação Baseada em Eventos Discretos . . . . . | 35       |
| 2.8      | Considerações Finais . . . . .                                   | 36       |

|          |  |           |
|----------|--|-----------|
| <b>3</b> | <b>Problema e Método de Abordagem</b>  | <b>37</b> |
| 3.1      | Descrição do Problema . . . . .  | 37        |
| 3.2      | Características Gerais do Problema . . . . .                                 | 38        |
| 3.2.1    | Definição do Layout . . . . .  | 38        |
| 3.2.2    | Processamento e sequenciamento . . . . .                                     | 39        |
| 3.2.3    | Ordem de produção . . . . .  | 40        |
| 3.3      | Simulação de eventos discretos - Modelo e Características Base . . . . .     | 40        |
| 3.3.1    | Entidades . . . . .  | 40        |
| 3.3.2    | Estação de Trabalho . . . . .  | 41        |
| 3.3.3    | Trajetos . . . . .   | 42        |
| 3.3.4    | Carga . . . . .  | 42        |
| 3.3.5    | Descarga . . . . .   | 42        |
| 3.3.6    | Transporte . . . . .   | 43        |
| 3.3.7    | Processos . . . . .  | 43        |
| <b>4</b> | <b>Implementação de um sistema de <i>Milk Run</i> clássico</b>               | <b>45</b> |
| 4.1      | Otimização do Layout de Comparação . . . . .                                 | 45        |
| 4.1.1    | Conjuntos e Parâmetros . . . . .   | 45        |
| 4.1.2    | Variáveis de Decisão . . . . .   | 46        |
| 4.1.3    | Restrições . . . . .   | 46        |
| 4.1.4    | Função Objetivo . . . . .  | 46        |
| <b>5</b> | <b>Implementação de um sistema de transporte dinâmico</b>                    | <b>51</b> |
| 5.1      | Heurística 1 . . . . .   | 51        |
| 5.1.1    | Descrição . . . . .  | 51        |
| 5.1.2    | Objetivo . . . . .   | 51        |
| 5.1.3    | Implementação . . . . .  | 52        |
| 5.1.4    | Resultados . . . . .   | 52        |
| 5.2      | Heurística 2 . . . . .   | 52        |
| 5.2.1    | Descrição . . . . .  | 52        |
| 5.2.2    | Objetivo . . . . .   | 52        |
| 5.2.3    | Implementação . . . . .  | 53        |
| 5.2.4    | Resultados . . . . .   | 54        |
| 5.3      | Heurística 3 . . . . .   | 54        |
| 5.3.1    | Descrição . . . . .  | 54        |
| 5.3.2    | Objetivo . . . . .   | 54        |
| 5.3.3    | Implementação . . . . .  | 55        |
| 5.3.4    | Resultados . . . . .   | 55        |
| 5.4      | <i>Neuroevolution</i> (NE) . . . . .   | 55        |
| 5.4.1    | Rede Neuronal - Definição da Rede . . . . .                                  | 56        |
| 5.4.2    | Algoritmo Genético . . . . .   | 58        |
| 5.4.3    | Integração entre o <i>Neuroevolution</i> e o Ambiente de Simulação . . . . . | 60        |
| 5.4.4    | Estratégia Evolutiva 1 - Estrutura da Rede e Resultados . . . . .            | 62        |
| 5.4.5    | Estratégia Evolutiva 2 - Estrutura da Rede e Resultados . . . . .            | 64        |
| 5.4.6    | Aplicação da solução num ambiente estocástico . . . . .                      | 65        |

|          |  |           |
|----------|--|-----------|
| <b>6</b> | <b>Análise de Resultados</b>   | <b>69</b> |
| 6.1      | Estratégia Evolutiva 1 . . . . .                                       | 69        |
| 6.2      | Estratégia Evolutiva 2 . . . . .                                       | 73        |
| 6.3      | Estratégia Evolutiva 1 vs Estratégia Evolutiva 2 . . . . .             | 77        |
| 6.4      | Avaliação da Estratégia Evolutiva 2 num ambiente estocástico . . . . . | 78        |
| <b>7</b> | <b>Conclusão e Trabalho Futuro</b>                                     | <b>81</b> |
| 7.1      | Conclusões . . . . .   | 81        |
| 7.2      | Trabalho Futuro . . . . .  | 82        |
|          | <b>Referências</b>   | <b>83</b> |



# Lista de Figuras

|      |  |    |
|------|--|----|
| 1.1  | Metodologia . . . . .  | 5  |
| 2.1  | Fundamentos Lean do Sistema de Produção Toyota, <i>adaptado de [1]</i> . . . . .                               | 8  |
| 2.2  | Princípio de produção <i>Pull</i> . . . . .  | 10 |
| 2.3  | Transporte <i>point-to-point</i> . . . . .   | 15 |
| 2.4  | <i>Milk Run</i> . . . . .  | 16 |
| 2.5  | Utilização de Armazém Intermédio . . . . .   | 17 |
| 2.6  | In-Plant <i>Milk Run</i> . . . . .   | 19 |
| 2.7  | <i>Modelo Teórico de Reinforcement Learning</i> , <i>adaptado de [2]</i> . . . . .                             | 26 |
| 2.8  | <i>Diagrama ilustrativo do NE através de uma perspetiva de alto nível, adaptado de [3]</i> . . . . .           | 28 |
| 2.9  | <i>Rede Neuronal com 4 nós de input , 2 nós de output e 2 hidden layers, adaptado de [2]</i> . . . . .         | 29 |
| 2.10 | Otimização Genética . . . . .  | 30 |
| 2.11 | Procura dos melhores parâmetros seguindo o gradiente, adaptado de [4] . . . . .                                | 32 |
| 3.1  | Esquema Geral do Sistema de Produção . . . . .   | 39 |
| 3.2  | Exemplo de uma Entidade no SIMIO e conjunto de propriedades associadas . . . . .                               | 41 |
| 3.3  | Exemplo de um objeto do tipo <i>Server</i> e conjunto de propriedades associadas . . . . .                     | 41 |
| 3.4  | Exemplo de um objeto do tipo <i>Source</i> e conjunto de propriedades associadas . . . . .                     | 42 |
| 3.5  | Exemplo de um objeto do tipo <i>Sink</i> e conjunto de propriedades associadas . . . . .                       | 43 |
| 3.6  | Exemplo de um objeto do tipo <i>Vehicle</i> e conjunto de propriedades associadas . . . . .                    | 43 |
| 3.7  | Exemplo de diferentes processos com diferentes etapas no SIMIO . . . . .                                       | 44 |
| 4.1  | Comparação entre <i>layouts</i> . . . . .  | 47 |
| 4.2  | Layout não otimizado em ambiente de simulação . . . . .  | 48 |
| 5.1  | <i>Neuroevolution</i> e SIMIO num contexto de RL . . . . .   | 56 |
| 5.2  | Rede Neuronal - Estrutura Idealizada . . . . .   | 56 |
| 5.3  | Algoritmo Genético - Implementação . . . . .   | 58 |
| 5.4  | Definição da População Inicial . . . . .   | 58 |
| 5.5  | Método de Seleção Natural implementado . . . . .   | 59 |
| 5.6  | Método de <i>crossover</i> implementado entre dois indivíduos com melhor avaliação de <i>fitness</i> . . . . . | 59 |
| 5.7  | Método de mutação aplicado após processo de <i>crossover</i> . . . . .   | 60 |
| 5.8  | Exemplo de um <i>Experiment</i> . . . . .  | 61 |
| 5.9  | Diagrama Síntese . . . . .   | 62 |
| 5.10 | Evolução dos valores <i>fitness</i> . . . . .  | 63 |
| 5.11 | Conjunto de pesos não aleatórios . . . . .   | 65 |

|      |  |    |
|------|--|----|
| 5.12 | Distribuição triangular relativa aos tempos de processamento estocásticos . . . .            | 66 |
| 6.1  | Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de $0.7m/s$ . . . . . | 70 |
| 6.2  | Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de $1m/s$             | 70 |
| 6.3  | Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de $1.5m/s$ . . . . . | 71 |
| 6.4  | Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de $0.7m/s$ . . . . . | 72 |
| 6.5  | Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de $1m/s$             | 72 |
| 6.6  | Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de $1.5m/s$ . . . . . | 73 |
| 6.7  | Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de $0.7m/s$ . . . . . | 74 |
| 6.8  | Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de $1m/s$             | 74 |
| 6.9  | Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de $1.5m/s$ . . . . . | 75 |
| 6.10 | Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de $0.7m/s$ . . . . . | 75 |
| 6.11 | Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de $1m/s$             | 76 |
| 6.12 | Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de $1.5m/s$ . . . . . | 76 |



# Lista de Tabelas

|     |  |    |
|-----|--|----|
| 3.1 | Sequenciamento para cada tipo de peça . . . . .  | 39 |
| 3.2 | Tempo de processamento para cada tipo de peça . . . . .  | 40 |
| 4.1 | Resultados do melhor <i>layout</i> para um sistema <i>Milk Run</i> . . . . .                   | 46 |
| 4.2 | Layout Não Otimizado - Valores de Produtividade . . . . .                                      | 48 |
| 4.3 | Layout Otimizado - Valores de Produtividade . . . . .  | 49 |
| 4.4 | Milk Run Clássico - Valores de Produtividade . . . . .   | 49 |
| 5.1 | Heurística 1 - Valores de Produtividade . . . . .  | 52 |
| 5.2 | Heurística 2 - Valores de Produtividade . . . . .  | 54 |
| 5.3 | Heurística 3 - Valores de Produtividade . . . . .  | 55 |
| 6.1 | Valores médios de produtividade associados a cada uma das estratégias evolutivas               | 77 |
| 6.2 | Tempos médios de processamento relativos a cada uma das estratégias evolutivas                 | 77 |
| 6.3 | Avaliação da produtividade em ambiente estocástico para um horizonte temporal de 52h . . . . . | 79 |
| 6.4 | Avaliação da produtividade em ambiente estocástico para um horizonte temporal de 36h . . . . . | 79 |



# Abreviaturas e Símbolos

|      |  |
|------|--|
| AGV  | <i>Automated Guided Vehicle</i>                  |
| FEUP | Faculdade de Engenharia da Universidade do Porto |
| MHE  | <i>Material Handling Equipment</i>               |
| MHS  | <i>Material Handling System</i>                  |
| MR   | <i>Milk Run</i>                                  |
| RL   | <i>Reinforcement Learning</i>                    |
| SPT  | <i>Sistema de Produção Toyota</i>                |
| VRP  | <i>Vehicle Routing Problem</i>                   |



# Capítulo 1

## Introdução

### 1.1 Contextualização da dissertação

A tecnologia evolui a um ritmo alucinante e as práticas seguidas em regime industrial não podem ser consideradas imaculadas. Existe uma necessidade de melhoria contínua adjacente à capacidade de manter a competitividade. As exigências do público são cada vez maiores, o que anteriormente era luxo agora é tomado como mera formalidade. Esta premissa pode ser corroborada com uma análise à evolução do conceito de qualidade ao longo do tempo “*Quality is conformance to the specifications*” [5], a conformidade com as especificações é agora dada como garantida, projetando, o cliente, a palavra qualidade para outro tipo de significado “*Quality is fitness for use*” [5].

A preferência, por parte do público, por um tipo de produto personalizado em função do standardizado veio provocar oscilações na metodologia e consequente modelo de processos de negócio [6]. Esta suscetibilidade à necessidade atual do consumidor vem obrigar os fabricantes, por forma a se manterem competitivos, a aumentarem a velocidade de resposta aos pedidos. Deste modo, verifica-se uma tendência de evolução relativa ao sistema de produção, em que o sistema de planeamento dos recursos depende, cada vez mais, de uma forte ligação ao módulo respeitante às encomendas do cliente, sendo flexível o suficiente afim de atender procuras irregulares. Desta forma, evitam-se problemas como a acumulação de stocks.

Para garantia de qualidade e simultânea produtividade devem ser colocados em especial foco a forma como se desenrola o processo produtivo e os elementos que o suportam. Existindo uma inevitável correlação entre o processo produtivo e a planta de fabrico, as empresas procuram melhorar a forma como os equipamentos estão organizados de modo a evitar desperdícios, deparando-se com problemas de *Material Handling* (MH), ou seja, referentes à forma como é feito o movimento de materiais no local de produção. Sempre que se apresenta como economicamente viável deve-se recorrer a técnicas de manipulação automatizadas também por motivos de libertação de recursos humanos para realização de tarefas com outro tipo de complexidade. O *Material Handling System* (MHS) revela-se essencial no suporte do processo de produção garantindo melhorias no que

respeita a diversos fatores desde a redução de inventário, regulação do *lead-time* e menores custos globais no processo produtivo, distribuição e transporte.

O especial foco deste tipo de sistemas de gestão de materiais assenta em veículos autónomos (AGVs – *Automated Guided Vehicles*) que, sendo não tripulados, garantem, com recurso a algoritmos de controlo para gestão do tráfego, um transporte eficiente dos materiais em curso, idealmente reduzindo o número de viagens necessárias [7].

Um dos sistemas mais comuns de utilização de AGVs em chão de fábrica é o *Milk Run*. Este sistema logístico garante um transporte segundo rotas cíclicas e de reduzida periodicidade [8]. Considerando o facto de os sistemas de fabrico não serem estáticos, faz sentido o sistema de transporte não se revelar, também, estático.

Novas técnicas de otimização têm surgido adjacentes ao desenvolvimento verificado na área de *Machine Learning*. Apesar do seu uso em sistemas de produção ser relativamente recente, gradualmente se tem verificado um aumento nos casos de sucesso de aplicações desta técnica em processos de otimização, previsão de falhas e controlo. Se corretamente exploradas, as técnicas de *Machine Learning* permitem uma gestão, análise e tratamento de conjuntos de dados de elevada dimensão extraindo relações implícitas, mesmo em ambientes de elevada complexidade.

Face ao conjunto de vantagens abrangidas por métodos de *Machine Learning* para otimização do processo produtivo de uma empresa, revela-se interessante perscrutar alternativas baseadas nestes métodos para resolução de problemas de movimentação de materiais. Neste sentido, torna-se importante o desenvolvimento de soluções dinâmicas capazes de se adaptarem a toda a flexibilidade exigida pelos atuais sistemas de produção.

## 1.2 Motivação

A tecnologia está a evoluir rapidamente e existe uma necessidade de desenvolvimento de soluções logísticas dinâmicas que se revelem adequadas a diferentes paradigmas de produção.

O facto de surgirem opções cada vez mais robustas nas áreas da automação e robótica permite o desenvolvimento de sistemas de fabrico cada vez mais avançados, constituindo enormes vantagens competitivas para as empresas. O conjunto de soluções é cada vez mais apto e sofisticado na resolução de problemas e execução de tarefas de grau de complexidade elevado.

De um ponto de vista histórico, os recursos humanos vêm sendo substituídos por robôs industriais sempre que as tarefas a realizar remetem para condições de trabalho desafiadoras, exigindo a máxima precisão e segurança, constituindo funções que um ser humano teria dificuldade em satisfazer.

A substituição do homem pela máquina exige investimento, e até recentemente, os elevados preços das soluções ao nível dos robôs industriais acarretavam custos excessivos que dissuadiam países cuja mão de obra era de custo reduzido a optarem por outro tipo de soluções. As soluções progressivamente mais robustas, económicas e flexíveis constituíram uma mudança no paradigma.

Para além do aumento do grau de automação no cenário industrial, também uma outra tendência, a da “customização em massa”, tem ganho cada vez mais importância no contexto produtivo.

Sendo característico desta tendência uma necessidade de produção de uma vasta gama de produtos com sequência de operações e tempos de processamento próprios, deve existir um sistema produtivo completamente adaptado a este novo paradigma.

Considerando a necessidade de os sistemas de produção se revelarem altamente responsivos a novos dados como diferentes *mixes* de produção, é importante conceber abordagens para o delineamento de rotas a serem percorridas capazes de se adaptarem a diferentes panoramas. As heurísticas e modelos de otimização são métodos à priori, que utilizam os dados disponíveis no momento da sua execução para definir as decisões a tomar a médio-prazo, já as técnicas de *Reinforcement Learning* permitem tomar decisões de forma reativa a cada novo evento que surja no processo produtivo. Assim, o recurso a técnicas de *Reinforcement Learning* pode ser uma boa abordagem para atacar as atuais exigências da realidade industrial, onde o contexto de produção é modificado com uma frequência tal que poderá inviabilizar um planeamento à priori. Um sistema de transporte inteligente, que observa a realidade do chão de fábrica a cada momento, e que usa a sua experiência de contextos de produção anteriores para tomar novas decisões, pode conseguir atingir performances superiores às do *Milk Run* clássico. Este sistema de transporte inteligente será mais flexível e mais capaz de fornecer os materiais certos na altura certa, levando a menores tempos de inatividade nas estações de trabalho.

É interessante realçar que, com estas medidas de melhoria no contexto do transporte de materiais é possível, sem serem feitos investimentos em novos equipamentos ou reformulação do *layout*, diminuir tempos de ciclo e tempos totais de produção de um determinado produto.

Além disso, é importante notar que não é incomum que aproximadamente 85% do tempo que um produto está alocado a um sistema de produção seja despendido em ações de transporte ou à espera das mesmas, constituindo mais de 80% dos custos associados a uma atividade produtiva [9, 10].

Desta forma, a aplicação de novas técnicas de *Reinforcement Learning* neste contexto pode simultaneamente melhorar a produtividade das fábricas e diminuir os seus custos de produção, sustentando-se num baixo investimento financeiro. Assente neste fator de desenvolvimento de soluções logísticas inovadoras e no facto desta linha de pensamento poder ser aplicada a diferentes indústrias é constituída a motivação deste trabalho.

### 1.3 Objetivos

O principal objetivo e contribuição desta dissertação consiste no desenvolvimento de um método inovador e flexível baseado numa estratégia híbrida que combina técnicas de *Reinforcement Learning* e Simulação para a definição do comportamento de um sistema de transporte AGV. Este sistema logístico deve ser capaz de maximizar os índices de produtividade associados a um sistema de produção.

Considerando uma sequência produtiva definida num contexto *job shop*, é necessário delinear o melhor planeamento de transporte dos diferentes componentes sob os diferentes postos de trabalho de forma a maximizar índices de produtividade e minimizar custos associados a este processo

de transporte.

Recorrendo à simulação baseada em eventos discretos, é pretendido avaliar os comportamentos seguidos no transporte de materiais por forma a compreender e testar diferentes abordagens a seguir. O objetivo passa por implementar e testar uma solução híbrida entre um algoritmo de *Reinforcement Learning* e um ambiente de simulação para resolução de um problema de transporte de materiais. O modelo de simulação, constituirá o ambiente com o qual o agente de *Reinforcement Learning* vai interagir.

De seguida, é pretendido validar a flexibilidade e aplicabilidade da abordagem desenvolvida alterando parâmetros de índole temporal e simultaneamente as características do próprio veículo de transporte, verificando a capacidade de adaptação do método a diferentes restrições.

Posteriormente, para avaliação do desempenho da metodologia desenvolvida, será feita uma comparação entre a proposta evolucionária com outros métodos de otimização baseados em heurísticas normalmente consideradas no panorama real da indústria.

Por fim, a solução híbrida desenvolvida será sujeita a um ambiente de produção com elevada variabilidade associada aos tempos de processamento. Assim, será possível avaliar a robustez do método em ambientes altamente estocásticos e retirar as respetivas conclusões acerca da viabilidade de aplicação do mesmo em casos reais.

## 1.4 Metodologia

A dissertação desenvolvida segue uma metodologia híbrida que combina técnicas de *Reinforcement Learning* com técnicas de Simulação para a resolução de um problema de desenvolvimento de um sistema logístico inteligente associado a um sistema de produção.

Numa fase inicial será desenvolvido um sistema de transporte baseado na ideologia do *Milk Run* clássico que servirá como objeto de comparação para a solução dinâmica a desenvolver numa fase posterior. Para isso, partindo da sequência de produção de cada tipo de peça e considerando que as ordens de fabrico são geradas de forma aleatória, é avaliado o *layout* base do problema para definir a disposição ótima das estações de trabalho. Com base nesta análise é pretendido implementar um sistema de transporte baseado no conceito de *Milk Run* clássico sobre uma organização do *layout* otimizada.

O modelo de simulação permite obter informações importantes que caracterizam o sistema produtivo segundo diferentes parâmetros de entrada que podem variar desde a alteração da velocidade do AGV responsável pelo transporte de materiais até à redefinição do horizonte temporal de simulação pretendido. Recorrendo ao software SIMIO como ferramenta de simulação é possível obter-se um modelo visual que, além de ser capaz de modelar o comportamento pretendido para o sistema, fornece um conjunto de resultados para análise servindo, assim, como base para a comparação entre diferentes alternativas.

Numa instância seguinte, é pretendido modelar 3 sistemas de transporte baseados em heurísticas. Posteriormente, será desenvolvida uma solução híbrida de um sistema de transporte dinâmico,



que deve ser capaz de definir a cada momento a melhor decisão sobre que tipo de peça transportar. Este sistema não possui uma rota cíclica pré-definida e deve mostrar-se mais flexível já que apresenta um ponto direto de transporte entre estações de trabalho não adjacentes.

Recorrendo a uma técnica de *Reinforcement Learning*, nomeadamente o *Neuroevolution*, serão geradas redes neurais responsáveis por definirem a estação para a qual o veículo se deve movimentar em cada momento de decisão.

Este algoritmo será integrado por API com o sistema de fabrico simulado permitindo a estruturação de uma análise comparativa entre as diferentes alternativas, assim como entre as próprias estratégias evolucionárias desenvolvidas.

Numa fase final é pretendido avaliar o desempenho da melhor solução num ambiente estocástico para validação da robustez do método.

O conjunto de etapas mencionado encontra-se representado na Figura 1.1.

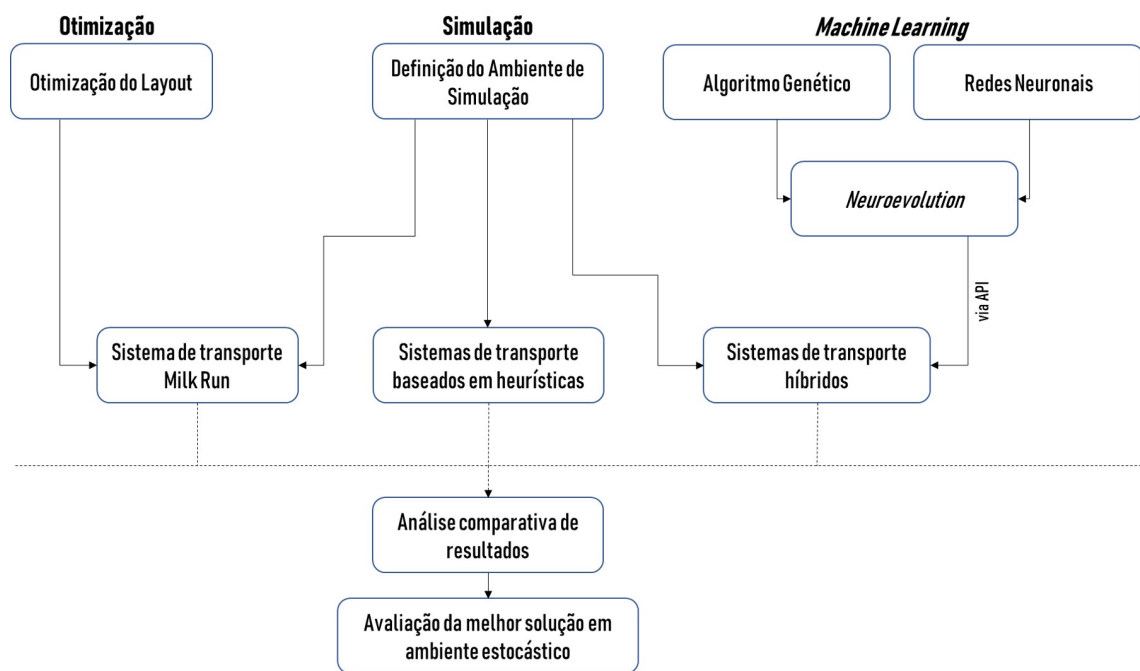


Figura 1.1: Metodologia

## 1.5 Estrutura da Dissertação

Em concordância com o trabalho desenvolvido, a dissertação está dividida num total de sete capítulos.

O primeiro capítulo é de cariz introdutório e define o contexto, motivação e objetivos que levaram à realização deste trabalho.

No capítulo 2 é apresentada a revisão bibliográfica sobre os temas relevantes para a realização desta dissertação. Neste são abordados conceitos sobre sistemas de produção e sistemas logísticos,

com especial foco no *Milk Run*. É também apresentada uma pesquisa detalhada sobre métodos de *Reinforcement Learning* e sobre a simulação de eventos discretos, evidenciando-se as vantagens e desvantagens deste tipo de abordagem quando aplicadas aos sistemas de produção.

No capítulo 3 são exibidas as características do problema em estudo. Além disso, é apresentada uma definição dos elementos de simulação, que se evidenciam como ponto comum no método de abordagem seguido para o desenvolvimento dos vários sistemas logísticos.

No capítulo 4 é exposta a metodologia seguida para o desenvolvimento de um sistema *Milk Run*.

O capítulo 5, por sua vez, apresenta as várias heurísticas e soluções baseadas em *Reinforcement Learning* desenvolvidas para a resolução do problema.

No capítulo 6 são apresentados e analisados os resultados das soluções desenvolvidas.

Por último, no capítulo 7 são referidas as conclusões do trabalho realizado e identificadas as eventuais iniciativas que seriam interessantes de explorar como trabalho futuro.

## Capítulo 2

# Revisão bibliográfica

Neste capítulo apresentam-se os temas, métodos, respetivas aplicações dos métodos e conceitos essenciais que suportam o trabalho desenvolvido ao longo da dissertação.

Nomeadamente, na secção 2.1 é apresentada a evolução dos sistemas de produção com especial foco no Sistema de Produção Toyota e a sua relevância no contexto industrial. Na secção 2.2 é esmiuçado o conceito de sistema de produção *job shop* e as suas características.

Na secção 2.3 é aprofundado o sistema de transporte *Milk Run* desde o conceito base, vantagens e variantes. A análise e estudo na área dos *Material Handling Systems* (MHS) e do próprio conceito *Automated Guided Vehicle* (AGV) é realizada na secção 2.4. Na secção 2.5 são ainda analisados alguns métodos de otimização do MHS apresentados na literatura.

Na secção 2.6 são detalhados conceitos de *Machine Learning* e examinados métodos para resolução de problemas de *Reinforcement Learning*, nomeadamente *Policy Gradients* e *Neuroevolution*, sendo este último a base da abordagem desenvolvida.

Por fim, na secção 2.7 são explorados conceitos associados à simulação de eventos discretos e a importância desta ferramenta.

## 2.1 Sistemas de Produção

### 2.1.1 Introdução

Os sistemas de produção são estratégias no âmbito da administração de empresas que organizam a maneira como é realizado o processo produtivo.

A forma como tem vindo a ser pensada a estratégia de produção tem mudado os paradigmas gerais de trabalho das organizações. Os principais tipos de produção aplicados nas cadeias produtivas ao longo do tempo são: Taylorismo, Fordismo e o Sistema de Produção Toyota.

### 2.1.2 Taylorismo

Elaborado por Frederick W. Taylor, o conjunto de premissas deste sistema baseia-se, essencialmente, em garantir a máxima produtividade através de uma ampla divisão de tarefas para que os

trabalhadores realizassem um trabalho repetitivo e standardizado para aplicação num sistema de produção em massa.

O Taylorismo ajudou a criar conflitos entre trabalhadores e gestores, entre departamentos, entre o negócio e o público, fazendo com que as operações não pudessem ser controladas por fatores temporais e que os trabalhadores se sentissem como meros fatores produtivos [11].

### 2.1.3 Fordismo

Elaborado por Henry Ford, é entendido como uma aplicação do Taylorismo ao sistema de produção fabril das empresas Ford. Mantendo as premissas Tayloristas de produção em massa este sistema destacou-se pela criação do conceito de “linha de montagem” garantindo que o produto em conceção chegasse mais rápido ao trabalhador.

Com a difusão deste sistema de produção em massa passaram a ser acumulados stocks excessivos desencadeando uma crise económica.

### 2.1.4 Sistema de Produção Toyota

Também chamado de sistema de produção flexível, o Sistema de Produção Toyota (SPT) foi criado na década de 1970 por Taiichi Ohno e Eiji Toyoda. Este conceito veio reformular a ideia de um sistema de produção.

Destacando um conjunto de ferramentas caracterizadoras, o SPT (Figura 2.1) refere também os elementos que sustentam a eficiência deste sistema produtivo.

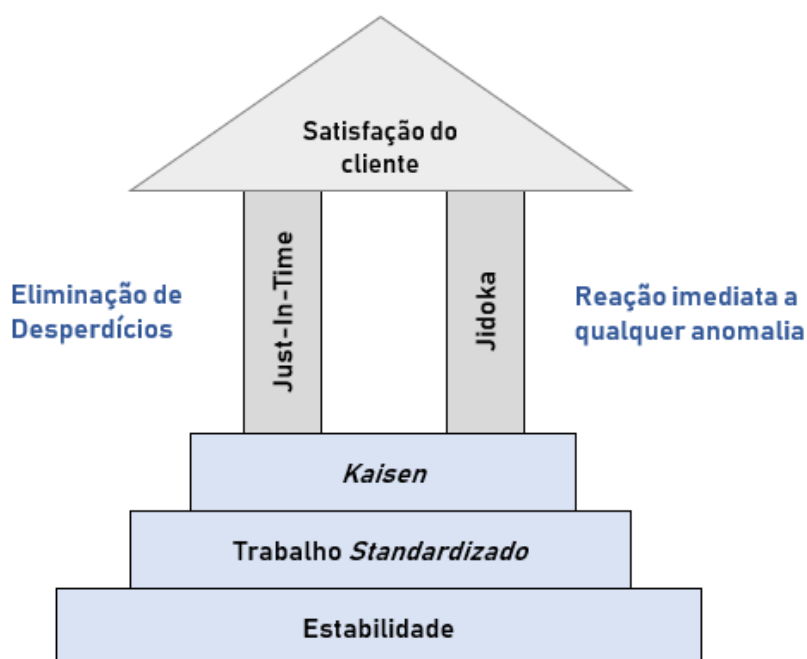


Figura 2.1: Fundamentos Lean do Sistema de Produção Toyota, adaptado de [1]

#### 2.1.4.1 *Just-in-Time (JIT)*

A base do sistema de produção Toyota assenta na eliminação de tudo o que é desperdício, considerando-se como desperdício qualquer processo que não acrescente valor [12].

O objetivo do JIT é garantir a produção de apenas a quantidade necessária a uma qualidade perfeita, sem excessos, de forma rápida, transportando produtos para os lugares certos, nas horas certas.

O JIT segue uma premissa de alocação das matérias primas na quantidade certa no tempo correto, estando relacionado com uma produção desencadeada pela procura que existe no mercado, ou seja, a adesão ao produto é o que ativa e despoleta o processo produtivo [13].

O conceito *Just-in-Time* significa que o material necessário só chega à linha de montagem quando necessário e na quantidade necessária. Seguindo, de forma rígida, esta premissa aproxima-se o valor de inventário a zero, que é uma das metas desta filosofia. Isto refere-se a um estado ideal, sendo tão mais difícil de implementar quanto mais complexo se revelar o processo produtivo.

De acordo com Denis [14], a necessidade da abordagem JIT resulta do forte desenvolvimento tecnológico e consequente aumento da competitividade verificado, apontando as entidades para um ciclo de melhoria contínua.

Maria Carravilha [15] identifica o conceito JIT como diferenciador na medida em que procura eliminar desperdícios em todas as frentes, evitando a realização de qualquer atividade que não acrescente valor ao produto final. A integridade de todos os elementos naquilo que é a empresa como um todo é fundamental para o sucesso. Para que esta filosofia funcione os trabalhadores devem ser devidamente instruídos no que respeita às suas áreas de trabalho, confiando-lhes a qualidade daquilo que produzem, ou seja, conferindo-lhes controlo sobre o seu trabalho. A autora, refere ainda que o processo de melhoria não deve ser nunca considerado concluído. Se existe margem de melhoria é exatamente com o conceito de perfeição em foco que a empresa, como um todo, deve atuar.

Os principais benefícios do *Just-In-Time* referem-se à redução do tempo de produção de toda a diversidade de produtos, conferindo maior flexibilidade ao sistema produtivo e uma redução do tamanho de lote a comprar com consequente redução do lote de produção, verificando-se uma diminuição do *Work-in-Progress* (WIP). Deste conjunto de fatores resulta uma redução do tempo que decorre desde que o produto entra no sistema de produção até à saída do mesmo [16].

#### 2.1.4.2 *Kanban*

O sistema *Kanban* é uma ferramenta que permite que um sistema de produção atinja níveis mínimos de inventário [17]. Este tipo de sistema requer um compromisso por parte dos fornecedores, garantindo uma entrega rápida de matérias primas com os índices de qualidade desejados para não comprometer o normal funcionamento do processo.

Baseando o seu funcionamento num sistema de abastecimento de um supermercado, o *Kanban*, através de cartões que controlam o fluxo de produção e transportes dentro daquilo que é o

processo produtivo (*kanban cards*) permite uma identificação ágil dos materiais a serem encomendados.

Como refere Lopes dos Reis [18], os *kanban cards* são alocados em locais específicos para identificar, com base em características do sistema de produção, como o *lead time* ou o stock de segurança, a necessidade de reposição de material. Quando o cartão ficar exposto, uma ordem de reposição deve ser gerada, garantindo-se, assim, inventários mínimos e consequentemente é evitada a necessidade de acumulação de stocks que acarretam custos de armazém.

Todo este processo é baseado no princípio *pull* – com a requisição de matérias primas a ser feita apenas quando necessária, considerando os níveis de afluência em torno do produto final como demonstra a Figura 2.2.

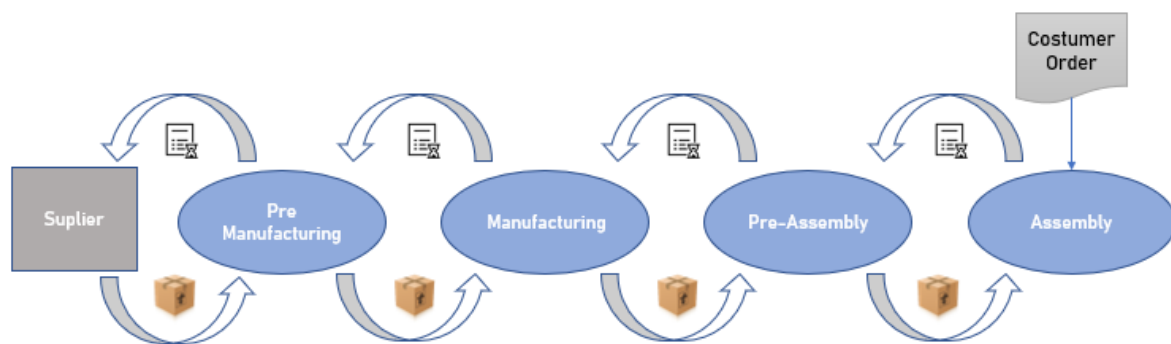


Figura 2.2: Princípio de produção *Pull*

#### 2.1.4.3 Jidoka

*Jidoka* é um sistema automático de verificação das características de uma máquina ou produto que suspende todo o processo produtivo no caso de verificação de uma não conformidade.

O processo consiste em parar todo o sistema produtivo sempre que é detetada uma anomalia. Expõe-se, desta forma, claramente o problema e viabilizam-se esforços para a compreensão do mesmo procurando-se gerar medidas corretivas por forma a que o problema não se repita ou propague.

Este tipo de práticas só vem cimentar a ideia acerca da nocividade da acumulação de stocks que podem ser usados para camuflagem de problemas de produção, desde problemas de qualidade - em que, no caso de ser verificada alguma inconformidade no *output* de algum estágio do processo, o recurso a stocks permitia que os locais de processamento posteriores continuassem o seu normal funcionamento - a problemas de avarias dos equipamentos - em que se garantia que não existia uma paragem de todo o processo produtivo até que a máquina voltasse ao seu normal funcionamento. Desta forma expõe-se de forma clara quais são os fatores limitativos do processo de produção projetando uma necessidade elevada para a resolução deste tipo de problemas e consequente melhoria.

## 2.2 Sistema de produção *job shop*

O termo *job shop* está associado a um processo produtivo de pequenos lotes de uma elevada gama de produtos. No fluxo produtivo deste tipo de sistemas, os produtos, no momento da sua produção requerem, geralmente, *set-ups* específicos e seguem uma sequência de operações devidamente estabelecida. O sistema de produção *job shop* está associado a uma gama de produtos customizados cuja finalidade não é a sua produção em massa. Considera-se o recurso a este sistema produtivo quando existe a necessidade de um sistema flexível capaz de se adaptar à variedade de produtos a fabricar.

### 2.2.1 Definição formal de um *job shop*

Formalmente, *job shop* é definido como um conjunto finito de  $n$  trabalhos,  $J = \{J_1, \dots, J_n\}$  que têm de ser escalonados num conjunto finito de  $M$  com  $m$  máquinas,  $M = \{M_1, \dots, M_m\}$ . Cada trabalho  $J_i$  é fragmentado numa série de operações  $o_{ik}$  de  $m$ , onde  $k$  representa a máquina  $M_k$  na qual a operação tem de ser processada. A ordem tecnológica das máquinas (rota de processamento) por cada trabalho  $i$  está predefinida. Cada operação  $o_{ik}$  está associada a um tempo de processamento não negativo  $p_{ik}$  [19].

### 2.2.2 Características

#### 2.2.2.1 Layout

Num sistema de produção *job shop* equipamentos semelhantes ou de funções equivalentes são agrupados juntos [20]. O *layout* desenhado pretende minimizar a movimentação de material (*Material Handling*), eliminando este tipo de desperdícios ao longo do local de produção, com a finalidade de reduzir o WIP.

#### 2.2.2.2 Routing

No mesmo sentido de evitar deslocações de material desnecessárias, é de considerar que nem todos os produtos utilizam todas as máquinas disponíveis, e à semelhança do que se verifica num hospital a designação do local para alocação das máquinas deve ser devidamente ponderada considerando as sequências de operações de cada produto. De um planeamento funcional ao nível das rotas e do *layout* a adotar é tornado o processo produtivo mais flexível permitindo a uma empresa satisfazer mais eficazmente aquilo que são as necessidades do consumidor em vez de fornecer um único produto *standard* massivamente produzido que pode não ir ao encontro daquilo que é expectável pela generalidade do mercado [21].

#### 2.2.2.3 Workers

A este tipo de sistemas de produção estão geralmente associados operários devidamente qualificados que sejam capazes de trabalhar com diferentes tipos de máquinas. Considerando as suas

qualificações, o nível de supervisão necessário diminui e é expectável que a qualidade do trabalho desenvolvido aumente. Evitando uma monotonia de trabalho adjacente a cada operário é possível manter os trabalhadores mais motivados [21].

#### 2.2.2.4 Scheduling

Cada processo produtivo alocado a cada tipo de produto é caracterizado pela sua rota, necessidades de processamento e prioridade. Num sistema *job shop* cada ordem de produção não deve ser realizada apenas com base na sua hora de chegada. Devem ser sempre considerados fatores baseados no tempo de processamento para o planeamento do processo produtivo [21].

### 2.2.3 Métricas de performance

Considerando a informação referida, são apresentadas, de seguida, um conjunto de métricas de performance de um sistema de fabrico *job shop* e os objetivos de otimização resultantes [19]:

- *Makespan*

A minimização do *makespan* é o objetivo de otimização mais popular no escalonamento de um *job shop*. Este critério reflete o período de tempo necessário para completar um conjunto em particular de ordens de trabalho. Quando é garantida a minimização deste critério, a fábrica obtém informação precisa sobre o plano produtivo. Desta forma, é possível estimar se um conjunto de tarefas pode ser finalizado dentro do horizonte de planeamento definido. Sendo o *makespan* referido como  $C_{max}$ , e  $C_i$  como o tempo para ser completado um trabalho  $J_i$ , o objetivo da otimização pode ser escrito como:

$$C_{max} = \max_{1 \leq i \leq n} C_i \rightarrow \min \quad (2.1)$$

- Utilização de máquinas

A utilização de máquinas refere o rácio entre a capacidade disponível e a capacidade necessária. Na manufatura, é desejável que cada máquina disponha de um tempo inatividade o mais reduzido possível.

A média de utilização das máquinas  $MU$  pode ser calculada por:

$$MU = \frac{\sum_{i=1}^n \sum_{k=1}^m p_{ik}}{m \cdot C_{max}} \quad (2.2)$$

O objetivo de otimização referente a esta métrica de performance passa pela maximização do tempo médio de utilização de cada máquina. Denota-se ainda que, atendendo a ambos os critérios de otimização já apresentados, a solução ótima gerada será a mesma.

- Tempo de fluxo



O tempo de fluxo  $F_i$  é definido como o período entre o lançamento  $R_i$  de um trabalho e a sua finalização  $C_i$ :

$$F_i = C_i - R_i \quad (2.3)$$

Esta métrica de performance é importante num contexto de *work-in-progress*. Quanto mais longo for o tempo de fluxo de um trabalho, maiores serão os custos de armazenamento do inventário.

A minimização do tempo total de fluxo é uma prática comum, em que são definidos, também, os pesos associados a cada trabalho sempre que necessário:

$$\sum_{i=1}^n w_i F_i \rightarrow \min \quad (2.4)$$

A minimização da soma pesada dos tempos de fluxo está intimamente relacionada com a minimização da soma pesada dos tempos de conclusão.

- Data de entrega

Sempre que as datas de entrega  $d_i$  são disponibilizadas devem ser considerados fatores como os atrasos e os desvios temporais das datas acordadas.

A medida mais simples de avaliação é o máximo atraso  $L_{max}$ , passível de ser obtido pela seguinte expressão:

$$L_{max} = \max_{1 \leq i \leq n} (C_i - d_i) \quad (2.5)$$

Um objetivo de otimização de elevada relevância passa pela minimização da soma pesada dos atrasos. Sendo um atraso  $T_i$  de um trabalho definido por:

$$T_i = \max_{1 \leq i \leq n} (0, C_i - d_i) \quad (2.6)$$

Desta forma, apenas desvios positivos relativamente à data de entrega são considerados. Pode-se, então, formular o objetivo associado da seguinte forma:

$$\sum_{i=1}^n w_i T_i \rightarrow \min \quad (2.7)$$

Seguindo a métrica previamente apresentada, a finalização antecipada dos trabalhos não tem impacto no valor da função objetivo. No entanto, seguindo uma filosofia *Just-in-Time* é desejável que a conclusão de todos os trabalhos associados a um pedido seja realizada o

mais próximo da data de entrega subjacente. Nesta situação, é considerado o desvio absoluto das datas de entrega:

$$\sum_{i=1}^n w_i |C_i - d_i| \rightarrow \min \quad (2.8)$$

Generalizando, é possível associar diferentes coeficientes  $\alpha_i$  e  $\beta_i$  relativos aos avanços e atrasos em função de uma data de entrega estabelecida, produzindo a função objetivo avanço-atraso:

$$\sum_{i=1}^n \alpha_i \max(0, d_i - C_i) + \sum_{i=1}^n \beta_i \max(0, C_i - d_i) \rightarrow \min \quad (2.9)$$

Considerando os pontos assinalados, crucial para o bom desempenho ao nível das métricas de performance do sistema é a definição da forma como os materiais devem ser transportados entre as máquinas responsáveis pelo processamento.

## 2.3 *Milk Run*

### 2.3.1 O Sistema Logístico

Bowersox [22] refere que nenhuma outra área envolve tanta complexidade de negócios ou extensão geográfica como a logística.

Meyer [23] define 3 conceitos chave no processo de transporte em instalações fabris:

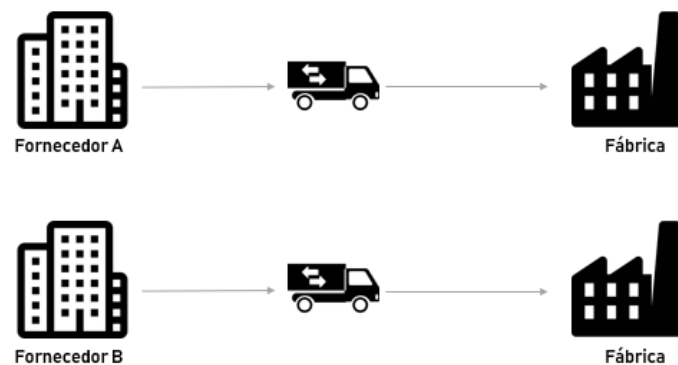
- *Area Forwarding Services*
- *Point-to-point*
- *Milk Run*

#### 2.3.1.1 *Point-to-Point*

No método de transporte *point-to-point* a recolha de produtos é feita pela transportadora no fornecedor e entregue diretamente ao cliente, como mostra a Figura 2.3.

Este método de transporte deve ser considerado quando o volume de produtos a encomendar é excessivo, alocando-se assim um meio de transporte exclusivo ao material [23].

Os custos adjacentes a este tipo de transporte estão ligados ao volume de encomendas e distância de entrega dos produtos.

Figura 2.3: Transporte *point-to-point*

### 2.3.1.2 Area Forwarding Services

Meyer [23] refere este método de transporte como um agrupamento especial de serviços em que uma empresa passa a responsabilidade sobre o transporte de produtos de uma determinada gama a uma fornecedora de serviços logísticos, deixando à mercê da empresa subcontratada o planeamento e execução dos transportes necessários.

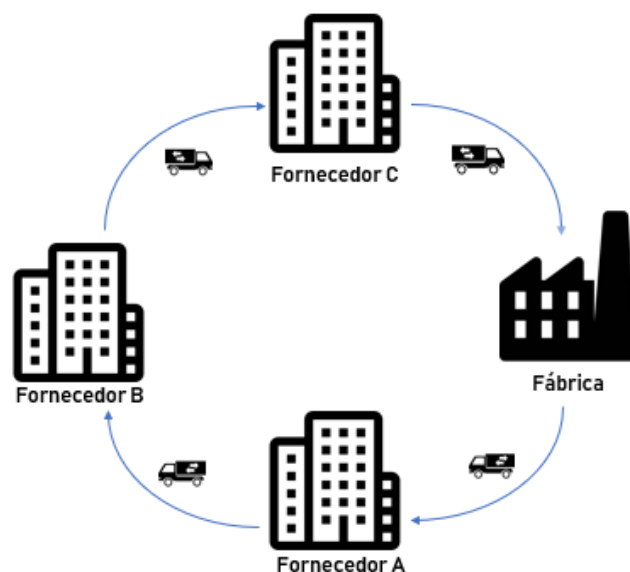
### 2.3.1.3 Milk Run

Baudin [24] refere-se ao conceito de *Milk Run* como um modelo de transporte que permite a movimentação de quantidades pequenas de materiais, prevendo o *lead time* de entrega dos mesmos sem que se verifique uma multiplicação dos custos de transporte.

Recorrendo a este método de transporte são planeadas, ciclicamente, viagens a vários fornecedores garantindo a entrega de diversos materiais em simultâneo na fábrica, com uma periodicidade bem estabelecida.

Este tipo de regularidade no processo de transporte permite uma redução da variabilidade associada ao mesmo. Como resultado, é permitido às empresas reduzirem os seus stocks de segurança, o que pode gerar preços reduzidos no produto final [23]. Este método de transporte facilita, também, as operadoras logísticas que podem utilizar esta regularidade nos processos de transporte para garantir taxas de ocupação elevadas nos seus meios de transporte minimizando desperdícios.

Comparado ao transporte *point-to-point* o *Milk Run* permite um transporte mais frequente e em menores lotes de materiais.

Figura 2.4: *Milk Run*

#### 2.3.1.4 Síntese

Considera-se o *Milk Run* mais complexo para implementação pela necessidade de desenho e planeamento de métodos eficientes que permitam uma ligação coerente e vantajosa para todos os intervenientes. Este método é adotado em situações que existe uma necessidade de recolha e entrega de material regular.

O conceito pode ainda ser aplicado ao interior da fábrica definindo rotas específicas ou estabelecidas dinamicamente para o transporte de materiais por forma a serem suprimidas as necessidades de cada máquina considerando sempre o plano de produção a seguir.

#### 2.3.2 O Sistema *Milk Run*

Considerando a flexibilidade da produção adjacente a um sistema de produção do tipo *job shop* é importante explorar um tipo de transporte flexível de materiais que seja capaz de se adaptar às necessidades produtivas momentâneas.

Neste tipo de sistema, o departamento de logística da empresa fabricante define quais os fornecedores que devem ingressar uma determinada rota e a necessidade em termos de periodicidade da recolha de materiais.

O fornecedor deve estar preparado para a recolha de peças com base nas informações recebidas respeitando os horários pré-estabelecidos.

Utilizando este tipo de sistema, que acarreta um transporte diverso de materiais, a descarga do veículo deve ser devidamente preparada por forma a não serem verificados desperdícios de índole temporal.

### 2.3.2.1 Vantagens que resultam da adoção deste método

- Redução do custo unitário:

Pelo facto de serem recolhidos exclusivamente os materiais necessários no horário definido, a quantidade desejada chega apenas quando solicitada evitando acumulações de stock e respetivos custos de armazém.

- Máxima utilização dos veículos de transporte:

Recorrendo ao mesmo meio de transporte é possível colmatar diferentes necessidades que surjam no que respeita ao material necessário.

O objetivo é que o veículo tenha sempre uma ocupação máxima do espaço disponível permitindo uma economia de tempo e dinheiro [25].

- Maior flexibilidade na receção de materiais:

Reduzindo-se o fluxo de veículos que chegam à fábrica e existindo um planeamento cuidado do horário em que se pretende a ocorrência da descarga é possível delinear exatamente os recursos a alocar para o respetivo processo aumentando a eficácia do mesmo.

### 2.3.2.2 Requisitos para implementação do Sistema Milk Run

- A Localização Geográfica dos Fornecedores:

Os fornecedores não devem estar geograficamente distantes do local a abastecer. Caso esta situação se verifique é necessária a adoção de um armazém intermédio (Figura 2.5) que não comprometa a rota planeada, sendo necessária a avaliação da rentabilidade da inserção do fornecedor.

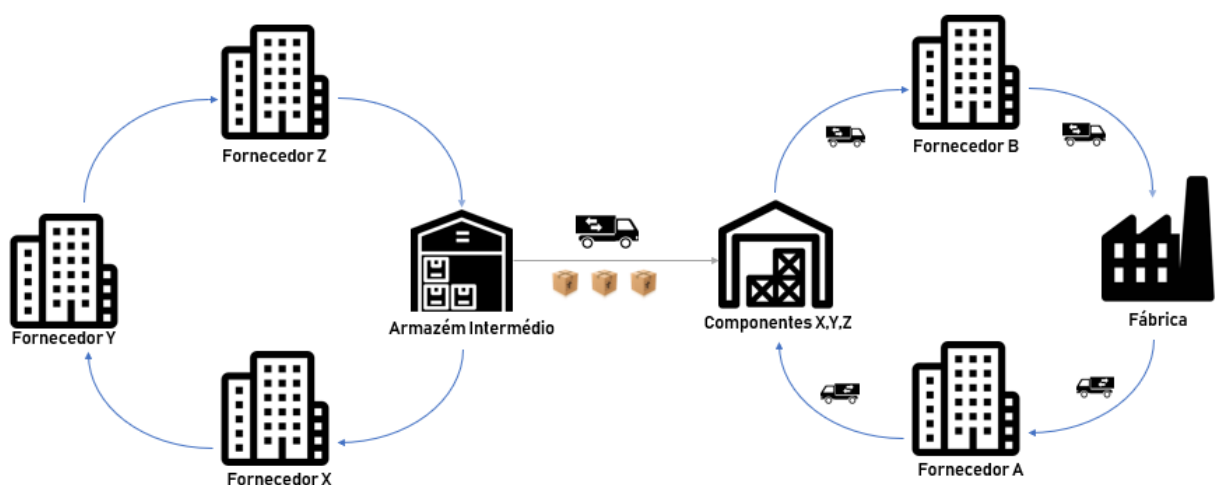


Figura 2.5: Utilização de Armazém Intermédio

### 2.3.2.3 Cumprimento das Janelas Temporais Delineadas

A transportadora deve ser capaz de respeitar, estritamente, a janela temporal alocada para entrega de materiais [26]. Incumprimentos, devido ao conceito de minimização de stocks adjacentes a este tipo de sistema, podem resultar numa paragem completa na linha de produção.

### 2.3.2.4 Cumprimento das especificações de qualidade

Pelo mesmo motivo, é da responsabilidade dos fornecedores o abastecimento de materiais dentro das especificações de qualidade acordadas [25].

### 2.3.2.5 Situações Não Aconselháveis Para *Milk Run*

O *Milk Run*, é preferencialmente utilizado para o transporte de componentes com frequência razoável.

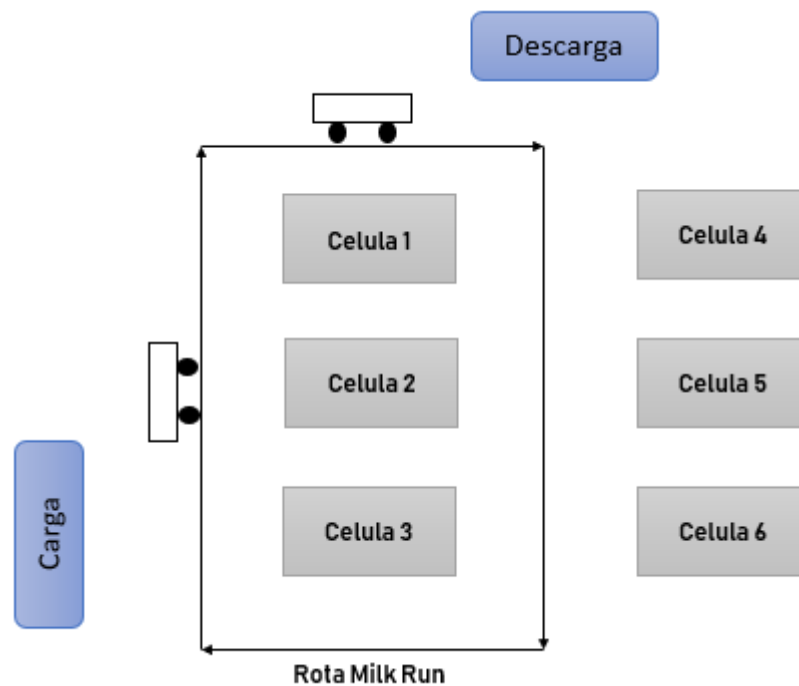
Atendendo à premissa supracitada é de evitar recorrer a este sistema de transporte nas seguintes situações:

- A necessidade de utilização de um determinado componente é de elevado volume. Nesta situação é razoável alocar um único meio de transporte exclusivo para supressão das necessidades.
- A necessidade de utilização de um determinado componente revela-se esporádica. Nesta situação o recurso a uma transportadora comum deve ser a solução.

## 2.3.3 *In-plant*

Todas estas características detalhadas para um sistema de transporte *outbound* podem ser mobilizadas para o próprio sistema de fabrico.

Domingo et al.[27] referem-se ao *Milk Run* como o sistema de transporte adequado para o movimento horizontal de materiais com recurso a veículos autónomos. O objetivo passa pela recolha de pacotes vazios de materiais e substituição por pacotes completos. Para o caso de não existirem materiais para recolha ou fornecimento o veículo segue a sua rota.

Figura 2.6: In-Plant *Milk Run*

Baudin [24] argumenta que a utilização dos métodos tradicionais de transporte no chão de fábrica, como por exemplo o transporte de materiais recorrendo a empilhadores, vai contra o conceito de *lean manufacturing*. Salienta, também, que grandes melhorias podem resultar da eliminação de viagens para transporte de materiais sobre a planta da fábrica, mais do que a intenção pela redução de distâncias.

O *Milk Run* reduz os desperdícios de logística interna no que se refere a inventários desnecessários ou movimentos excessivos de materiais ao longo da planta [27], isto sem ser necessária uma reforma da filosofia ou *layout* de produção.

## 2.4 Material Handling System

O MHS é referido como o processo e o sistema que, respetivamente, tratam da movimentação e gestão de materiais de um local para outro [28].

Na atualidade a competitividade é geral em qualquer área de negócio, os consumidores exigem *lead times* cada vez mais reduzidos e mais qualidade e flexibilidade de trabalho. De forma a manter a competitividade é importante atender dois fatores, a satisfação do cliente e a redução dos custos de produção. Um MHS bem desenvolvido é o caminho para atingir estes objetivos [29].

Se o fornecimento de materiais não for eficiente, repercussões são sentidas no processo de fabrico [30].

O desenho de um MHS envolve vários fatores, desde a seleção do tipo de equipamento a utilizar (Material Handling Equipment - MHE), o(s) tipo(s) de carga(s) a movimentar e o *pathing* a seguir pelos equipamentos [31].

Os diferentes MHE podem ser classificados em 4 categorias principais [32]:

- Armazenamento
- Posicionamento
- Formação de lotes
- Transporte

Chu et al.[32] referem que equipamentos de transporte são responsáveis pela movimentação de materiais entre locais enquanto os equipamentos de posicionamento são utilizados para manipulação do material num ponto fixo específico.

Banks et al.[9] referem três formas para a movimentação de materiais: *conveyors*, *unconstrained vehicles* e *guided vehicles*.

*Conveyors* são definidos como dispositivos fixos para o transporte de materiais ponto-a-ponto, através de rotas fixas com pontos de paragem e processamento definidos. Estes podem ainda ser classificados segundo as repercussões provocadas pela paragem dos itens que transportam. Consideram-se acumulativos se o tapete estiver em contínuo movimento ou não acumulativos quando a paragem do tapete provoca a paragem de toda a linha.

*Unconstrained Vehicles* são capazes de optar por rotas alternativas ao passo que os *Guided Vehicles* circulam por caminhos pré-estabelecidos com capacidade de decisão limitada no que respeita a obstáculos e congestionamentos.

Nomura e Takakuwa [33] definem duas categorias de classificação para os métodos de transporte de componentes:

- *Manual-based* - sistema no qual o transporte de materiais é da responsabilidade dos trabalhadores;
- *Automated-based* - sistema recente baseado num método de funcionamento automatizado que adota equipamentos de transporte de materiais para a transferência de produtos entre processos e armazém, como por exemplo o AGV (*Automated Guided Vehicle*).

Considerando ambientes em que são exigidas constantes reconfigurações do *layout* face à alteração dos produtos a conceber e consequentes métodos de fabrico, é recomendável a abordagem manual face aos custos de replaneamento de rotas e redefinição de parâmetros para uma movimentação ótima dos AGVs.

#### 2.4.1 *Automated Guided Vehicle*

Um sistema de AGVs corresponde a um conjunto de veículos autónomos e à sua movimentação ao longo de rotas planeadas na planta da fábrica. Sem intervenção humana, é alocada a estes



veículos a responsabilidade de transporte de produtos nas zonas de produção e de armazenamento. Recorrendo a este tipo de sistema são expectáveis melhorias na eficiência dos processos e reduções de custos face ao investimento em mão de obra humana.

É característico destes veículos apresentarem velocidades de transporte adequadas ao tipo de material que transportam e aos locais específicos de paragem. Além disso, este tipo de sistema exige um conjunto de interrupções para que seja feito o carregamento da bateria do equipamento de transporte.

#### **2.4.2 Condições ideais de aplicação**

A Dematic [34] refere as seguintes condições como ideais para implementação de um sistema de AGVs:

- Transportes em ambientes desafiadores;
- Transporte de e para células de produção e armazéns;
- Transportes repetitivos e de longas distâncias;
- Fluxos de Produção de elevada complexidade.

O foco passa pelo desenvolvimento de soluções dinâmicas para alocação dos veículos para que possam ser concebidas soluções mais robustas sem a necessidade de reformulação dos *layouts*.

#### **2.4.3 Design do sistema AGV**

Numa fase inicial de planeamento é importante identificar todos os pontos de recolha e libertação de materiais. Posteriormente, devem ser definidas as rotas a serem percorridas segundo os seguintes aspetos:

- As rotas devem seguir uma orientação unidirecional segundo uma filosofia cíclica.
- O recurso a rotas cruzadas é admissível se apresentar um conjunto de vantagens temporais significativas.

#### **2.4.4 Controlo do Sistema AGV**

À medida que o número de veículos aumenta a frota pode tornar-se cada vez mais complexa e dispendiosa. Devem, então, ser discutidas um conjunto de estratégias operacionais com o objetivo de alcançar níveis de utilização e produtividade ótimos:

- Regras de seleção de veículo – Por forma a serem evitadas viagens longas sem carga desaproveitando-se os recursos, sempre que existe uma carga a ser transportada e mais do que um AGV estiver disponível, a seleção passa pelo equipamento mais próximo do ponto de recolha.

Em circunstâncias de igualdade a opção deve recair pelo equipamento com menor volume de utilização.

- Regras de seleção de trabalho – Sempre que um AGV fica disponível e mais de que uma carga aguarda transporte o processo de decisão sobre qual deve ser movimentada primeiro deve ser baseado nos seguintes fatores:
  - Cargas com maior tempo de espera;
  - Cargas com menor tempo de espera;
  - Cargas com maior índice prioritário;
  - Maior número de cargas num mesmo ponto de recolha.
- Regras de paragem dos veículos – Se não existir, após entrega de uma carga, necessidade de movimentação de outras cargas, a decisão do local de paragem do equipamento deve considerar as seguintes ações:
  - Manter a posição atual;
  - Deslocação para um ponto de carregamento de bateria;
  - Movimentação para o local de maior probabilidade para a próxima requisição de transporte.

## 2.5 Otimização do *Material Handling System*

O transporte de materiais é um processo chave no que se refere à competitividade de qualquer entidade produtiva. Esta atividade acarreta elevados desperdícios, e por forma a manter a competitividade é essencial o desenho de um sistema que permita a eliminação de todos os processos que não acrescentam valor. Os objetivos de um sistema de transporte devem ser os seguintes [35] :

- Aumento da produtividade;
- Diminuição dos custos de transporte;
- Fluxo de entregas eficiente segundo uma filosofia JIT;
- Melhoria das condições de trabalho e segurança.

Baudin [24] agrupa os processos logísticos em três grupos de acordo com a sua aplicação: logística *outbound* (da fábrica para o cliente), logística *inbound* (do fornecedor para a fábrica), logística *in-plant* (logística no interior da fábrica). O problema em estudo enquadra-se numa logística *in-plant*.

### 2.5.1 Otimização de MHS e *design do layout*

A conceção de um MHS está diretamente relacionada com os *Vehicle Routing Problems* (VRPs). Apesar das semelhanças verificadas nos VRPs de um sistema de produção e os VRPs clássicos no que se refere às entregas, carregamentos e capacidade dos equipamentos, é importante considerar que, *in-plant*, as quantidades necessárias de stock são alteradas de acordo com

o ciclo periódico do veículo de transporte. *In-plant*, os pontos físicos de acumulação de stocks apresentam limitações, os níveis de inventário estão dependentes do tempo de ciclo do veículo e tudo isto considerado apresenta um conjunto de dificuldades à aplicação de um modelo *Milk Run* neste ambiente face às logísticas *in* ou *outbound*.

Do objetivo comum de minimização do número de equipamentos para movimentação de materiais e consequente eliminação do conjunto de atividades que não acrescentam valor, é subentendida uma relação entre os problemas de desenho de um MHS e os problemas de definição de *layout*.

A literatura apresenta um conjunto de abordagens distintas para a geração de soluções a este tipo de problemas. Como exemplificação faz-se uma breve nota das seguintes:

Hwang [36] aplicou, na definição de um *layout* industrial, um problema de roteamento de veículos (VRP) com o objetivo de determinar o número mínimo de viaturas necessárias para que estas não constituíssem o fator limitativo do sistema produtivo. A abordagem proposta agrupou um modelo de otimização de veículos a um modelo heurístico integrado face à não linearidade do problema abordado.

Satoglu e Sahin [37] propuseram um modelo matemático não linear e uma abordagem heurística adequada para a conceção de um sistema de fornecimento de materiais baseado no sistema *Milk Run*.

Domingo et al.[27] desenvolveram um sistema de distribuição *Milk Run* para transporte capaz de solucionar a acumulação de materiais em pontos de stock intermédios de uma linha de montagem. Baseando-se em princípios *lean*, a solução desenvolvida permitiu evitar transportes desnecessários e uma redução do inventário sem a necessidade de reformulação do *layout* ou da própria filosofia produtiva. Os autores acrescentam ainda a necessidade de adaptação do método face às diferenças de cada organização no que se refere ao *layout*, fluxo de materiais ou outras características produtivas.

Teschemecher et al.[38] desenvolveram um algoritmo baseado no estudo do comportamento de formigas, que libertando feromonas quando se verifica um caminho válido até um alimento permite o seguimento de toda a colónia que pode também divergir deste primeiro caminho delineado na deslocação até ao alimento. Neste algoritmo desenvolvido, os autores seguem duas abordagens:

- Primeiro, partindo de uma solução aleatória, procuram uma solução mais eficiente em termos temporais utilizando os mesmos recursos de transporte.
- Com a segunda abordagem é pretendido gerar uma solução igualmente válida alocando menos um recurso de transporte às restrições do problema.

Sempre que é verificada uma solução válida o algoritmo deixa de se focar na otimização da solução aleatória, procurando gerar otimizações para a nova solução encontrada. Com este sistema de otimização os autores focam-se principalmente na minimização de recursos a alocar em função da minimização do *lead time* total.

## 2.6 *Machine Learning*

As heurísticas e modelos de otimização definidos à priori utilizam dados disponíveis no momento. No enquadramento daquilo que deve ser o futuro e com o foco no desenvolvimento de soluções dinâmicas, é importante explorar técnicas de *Machine Learning*, mais concretamente *Reinforcement Learning* (RL), capazes de viabilizar soluções robustas no que se refere ao transporte de materiais num sistema *job shop*.

Num contexto mais geral, *Machine Learning* pode ser definido, segundo Arthur Samuel [39], como o campo de estudo que permite a aprendizagem sem uma programação explícita. Tom Mitchell [39], num contexto mais orientado à engenharia, refere que se pode afirmar que um programa computacional aprende sempre que a partir uma determinada experiência  $E$  relativa a uma tarefa  $T$  e uma medida de desempenho  $D$ , se verificar uma melhoria de desempenho na tarefa  $T$ , segundo o parâmetro  $D$  resultante da experiência  $E$ .

De acordo com a forma como a aprendizagem é realizada, *Machine Learning* pode ser caracterizado segundo 3 diferentes tipos, *Supervised Learning*, *Unsupervised Learning* e *Reinforcement Learning*[2].

- *Supervised Learning*

A aprendizagem neste método é realizada seguindo um conjunto de dados devidamente rotulados fornecidos por um supervisor externo. Cada exemplo é, essencialmente, a descrição de uma situação e consequente ação a seguir pelo sistema, sendo apropriada para aplicações de classificação [2].

- *Unsupervised Learning*

Este tipo de aprendizagem parte de um conjunto de dados não rotulados com a função de descobrir padrões e agrupá-los devidamente segundo um conjunto de características comuns. A utilização deste tipo de aprendizagem remete essencialmente para efeitos de *clustering*, sendo bastante relevante na exposição de situações anómalas, isto é, dados que não se enquadrem com nada do conjunto [39].

- *Reinforcement Learning*

Este tipo de aprendizagem, ao contrário do *supervised learning* não parte de um conjunto de dados rotulados. Através da experiência absorvida baseada num conjunto de recompensas, à semelhança do que se verifica com o ser humano, é explorada a ação a tomar. Essa ação não é explícita e deve ser inferida com base no maior conjunto de recompensas resultantes do feedback de um vasto conjunto de diferentes ações [2].

Dadas as características dos tipos de aprendizagem e do facto do transporte de materiais depender de uma situação momentânea não pré-definida é importante explorar o *Reinforcement Learning*, mais concretamente as técnicas de *Neuroevolution* (NE) e *Policy Gradients* (PG).

### 2.6.1 Reinforcement Learning

*Reinforcement Learning* pode ser caracterizado como uma abordagem computacional para perceber e autonomamente aprender a melhor decisão a ser tomada numa determinada situação dado o objetivo de um problema [2]. O caráter diferenciador desta abordagem revela-se pela ausência de necessidade de supervisão ou conhecimento total de um ambiente de aplicação para chegar a um conjunto de resultados satisfatórios. Partindo de um conjunto de objetivos específicos e seguindo um paradigma causa-efeito em relação a cada tomada de decisão é possível aferir o conjunto de decisões relevantes para a solução de um determinado problema.

Um dos desafios impostos ao RL e não aos outros tipos de *Machine Learning* é a decisão acerca do *trade-off* entre explorar novas decisões que levarão a novos resultados, não necessariamente melhores, ou manter decisões tomadas no passado que garantem resultados satisfatórios mas limitam a capacidade de encontrar novas soluções [2]. O RL deve, então, ser capaz de aproveitar as decisões que levaram a resultados positivos e decidir quando explorar novas decisões por forma a melhorar a solução obtida (maximizar conjunto de recompensas) mas sempre considerando que uma decisão que gerou resultados satisfatórios não possa ser subjugada por outra capaz de gerar resultados ainda melhores. Géron [39] dá um exemplo neste sentido referindo que quando vamos a um restaurante pela primeira vez e todos os pratos têm um aspeto igualmente apelativo a decisão revela-se aleatória. Repetindo a ida ao restaurante e o prato escolhido na primeira vez se ter revelado uma boa escolha vai aumentar a probabilidade de se repetir a escolha mas não até 100% ou desta forma privaria toda a opção por restantes pratos que se podiam revelar melhores opções que, então, nunca seriam exploradas.

### 2.6.2 Reinforcement Learning - Características

Em problemas de *Reinforcement Learning*, a descrição do ambiente nem sempre é completa e como consequência não é possível delinear automaticamente uma política baseada nessa descrição. Desta forma, é importante que o agente seja capaz de aprender a melhor política baseando-se na experiência resultante de um conjunto de interações com o ambiente [40]. Para perceber melhor este tipo de problemas devem estar claros um conjunto de conceitos associados a sistemas RL assim como as suas interações que se apresentam na Figura 2.7.

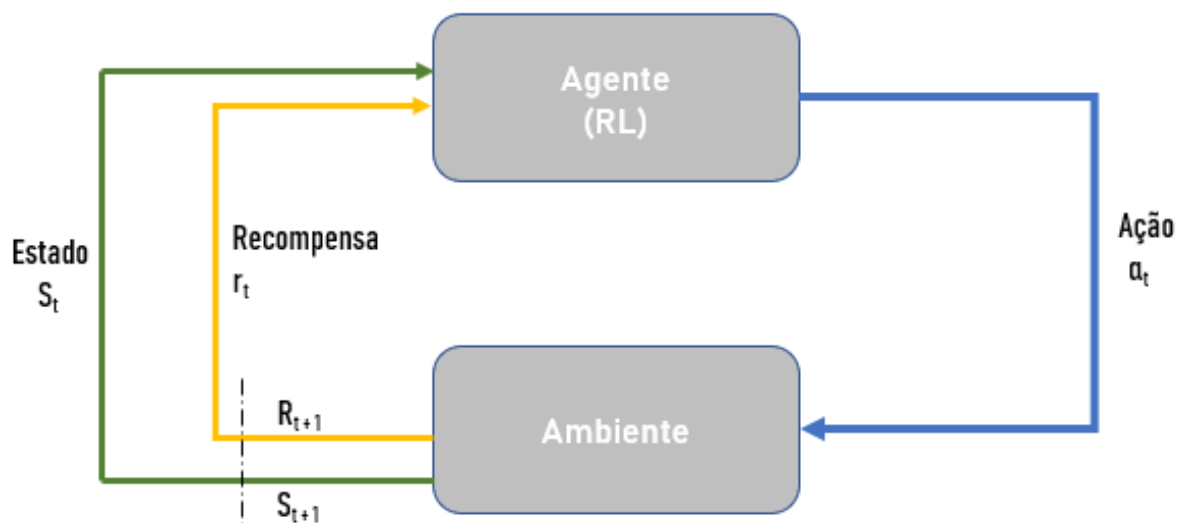


Figura 2.7: *Modelo Teórico de Reinforcement Learning*, adaptado de [2]

#### 2.6.2.1 Agente

É caracterizado como agente o elemento responsável pela aprendizagem e tomada de decisão [2]. Considerando o estado do sistema  $S_t$  e o conjunto de recompensas geradas pelas suas ações o agente é responsável pela melhoria da sua política [ver abaixo].

#### 2.6.2.2 Ambiente

O ambiente pode ser caracterizado como o conjunto de coisas com as quais o agente interatua [2]. Este é responsável por indicar ao agente o estado atual e o conjunto de ações possíveis [40]. Posta a tomada de decisão por parte do agente, o ambiente também fornece o valor da recompensa  $R_t$  gerada para posterior avaliação.

#### 2.6.2.3 Ação

As ações modelam a forma como as decisões do agente vão afetar o ambiente do problema. Pelo facto de ações diferentes poderem gerar recompensas diferentes cabe ao agente encontrar a sequência ótima de ações a serem tomadas por forma a maximizar o conjunto de recompensas obtidas.

#### 2.6.2.4 Recompensa e Objetivo

A recompensa representa a realimentação que permite avaliar o sucesso ou falha de qualquer ação seguida pelo agente. Em RL, o objetivo do agente pode ser formalizado em função de um conjunto de recompensas, isto é, o objetivo passa por uma acumulação de recompensas num horizonte temporal a longo termo, salientando a ideia de que uma ação capaz de gerar uma recompensa imediata pode levar a uma situação não ideal e descontextualizada do objetivo do agente [2].

### 2.6.2.5 Política ( $\pi$ )

Política pode ser traduzida como a estratégia seguida por um agente para determinar a próxima ação a tomar com base no estado atual. Formalmente, Sutton [2] define uma política como o mapeamento entre um estado e a probabilidade de seleção de cada uma das ações possíveis adjacentes a esse mesmo estado. Como o objetivo passa pela maximização de recompensas, então política ótima pode ser considerada aquela que num determinado panorama temporal consegue maximizar a acumulação de recompensas, ou seja, apresentar resultados iguais ou melhores que qualquer outra política.

Técnicas de *Reinforcement Learning* podem, então, ser consideradas orientadas ao objetivo, apontando um conjunto sequencial de ações a serem seguidas para que o agente consiga maximizar um conjunto de recompensas.

O tempo pode ser considerado um fator limitativo na obtenção de resultados a partir da experiência e é neste sentido que este tipo de aprendizagem se diferencia dos humanos. Podendo correr os algoritmos de aprendizagem em processadores de alto desempenho é possível avaliar um conjunto de situações que seria impensável no tempo de vida útil humana, limitando aquele que poderia ser o conhecimento obtido a partir da experiência.

Abordados os conceitos relativos a métodos de RL é importante explorar um conjunto de técnicas utilizadas neste sentido para definição de políticas com desempenhos ótimos na solução de problemas derivados de uma ausência de conhecimento total do ambiente de aplicação.

## 2.6.3 Técnicas de *Reinforcement Learning*

### 2.6.3.1 Neuroevolution

*Neuroevolution* é uma técnica de *Machine Learning* aplicada a algoritmos evolucionários para construção de redes neurais artificiais (RNA). Esta técnica permite a aprendizagem sem especificar um conjunto de dados à priori, apenas com base em *feedback*, um modelo neuronal arbitrário e a estrutura da rede. O método permite reformular pesos ou a própria estrutura da rede para encontrar soluções mais robustas ou aprender uma determinada tarefa cujo desempenho é medido através de uma função de *fitness* que se pretende maximizar [3].

Métodos baseados em NE atingem especial relevância em domínios de aplicação de *Reinforcement Learning* já que se baseiam num conceito de seleção natural que consiste na seleção de indivíduos com um desempenho geral superior [3]. Desta forma, pode ser gerada uma solução robusta naquilo que é a definição da política ótima no contexto de um problema de RL.

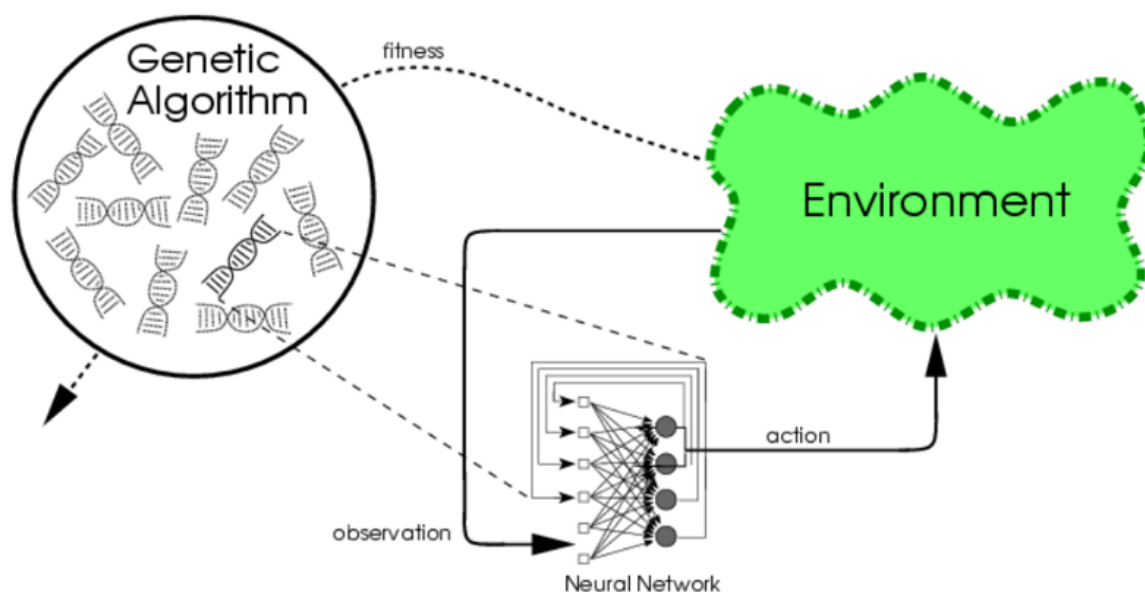


Figura 2.8: Diagrama ilustrativo do NE através de uma perspectiva de alto nível, adaptado de [3]

Embora o NE não consista num método de RL tão complexo e desenvolvido como métodos mais recentes, na medida em que faz uma procura "cega" por novas soluções, estudos recentes indicam que a sua rapidez de execução e de geração de novas soluções o tornam num método capaz de competir com aqueles que constituem o estado-da-arte, como exemplifica o trabalho feito pela Uber AI [41].

Para consolidação de ideias relativas ao funcionamento do método NE é importante explorar um conjunto de conceitos base relativos a Redes Neurais Artificiais assim como Algoritmos Genéticos para desenvolvimento e evolução das redes.

### 2.6.3.2 Redes Neurais Artificiais

Redes Neurais Artificiais podem ser definidas como modelos computacionais implementados em software ou em dispositivos de hardware especializados que procuram capturar o comportamento e características de um sistema biológico nervoso [42]. Estas são geralmente compostas por um conjunto de unidades de processamento que podem ser denominados por neurónios. Em termos matemáticos, estas podem ser vistas como um grafo direcionado em que os nós representam os neurónios e as arestas de ligação entre nós podem ser vistas como a influência que a ativação de um neurónio vai causar noutro neurónio adjacente. O número, tipo e possíveis interligações entre neurónios definem a arquitetura da rede.

O treino da rede, para obtenção de melhores resultados, relaciona as suas entradas e saídas podendo atualizar a estrutura da rede ou apenas os pesos entre arestas que ligam os nós, sendo que para isso não necessita de um conjunto de regras idealmente definidas mas apenas de um conhecimento do ambiente de aplicação da rede e do conjunto de informação disponível para a mesma.



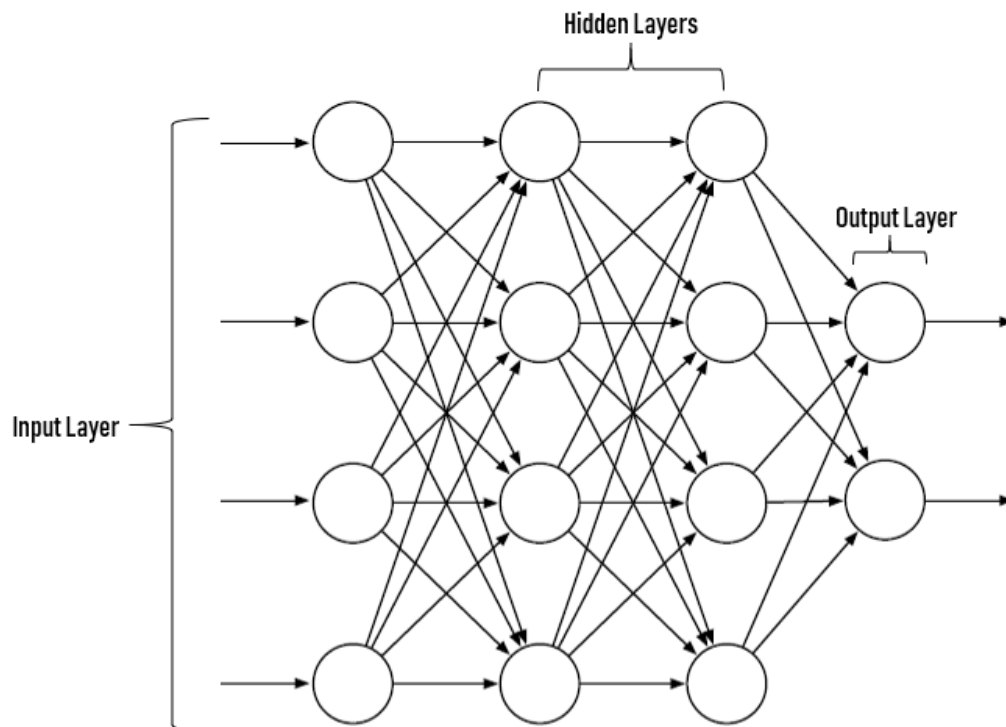


Figura 2.9: Rede Neuronal com 4 nós de input , 2 nós de output e 2 hidden layers, adaptado de [2]

### 2.6.3.3 Algoritmo Genético

Para solução de problemas computacionais ou de engenharia com redes neurais são utilizados algoritmos genéticos para encontrar parâmetros ideais para a rede que modela o problema [42].

Estes algoritmos são baseados na população [42] seguindo um conjunto de princípios evolutivos que partem de uma representação aleatória abstrata de uma solução em busca de resultados melhores. Para isso, é considerada a evolução das soluções ao longo de diferentes gerações que são selecionadas ou descartadas segundo a qualidade que apresentam do ponto de vista da função de *fitness* que o algoritmo tenta otimizar no contexto do problema. Esta evolução nas soluções deve-se ao facto de serem introduzidas mutações ou recombinações entre gerações para gerar uma nova população.

- Recombinação Genética

Esta operação genética consiste na combinação do material genético de duas ou mais soluções para gerar uma solução igualmente válida mas não necessariamente melhor na ótica do objetivo do problema.

- Mutação

Recorre-se à inserção de mutações no código genético para que o algoritmo genético altere a sua área de pesquisa podendo atingir soluções que de outra forma estaria privado de alcançar. Este processo verifica-se após a recombinação genética podendo ocorrer de forma intencional ou aleatória.

Neste contexto de aprendizagem, entende-se a população como um próprio *input* para evolução deste tipo de algoritmos.

Através desta base, as características das redes neuronais podem ser codificadas em genomas artificiais (nós) e desenvolvidas segundo critérios de desempenho relevantes para o problema envolvente.

As vantagens da utilização de algoritmos evolutivos passam pela codificação e evolução genética simultânea das características que definem a rede neuronal [42].

#### 2.6.3.4 Implementação do Neuroevolution

Considera-se o objetivo de aplicação da metodologia NE encontrar uma rede neuronal que melhor desempenhe uma tarefa específica. Tipicamente, como abordagem inicial é gerado, aleatoriamente, um modelo e um conjunto de descendentes baseados nesse mesmo modelo que codificam geneticamente várias redes neuronais.

Partindo desta população original é desencadeado um ciclo de otimização genética (Figura 2.10) baseado nos algoritmos genéticos previamente apresentados.

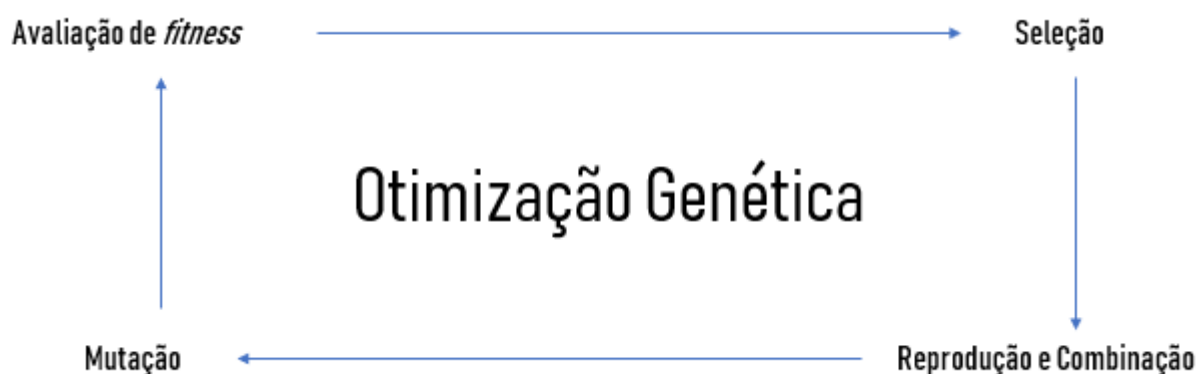


Figura 2.10: Otimização Genética

Primeiramente, é feita uma avaliação de *fitness* com a intenção de verificar quais os modelos que apresentam melhores índices de desempenho. De seguida, é feita uma seleção baseada nos valores de *fitness*, sendo estes valores definidos com base num acumular de recompensas. Reduz-se, assim, a descendência a um único modelo que apresente melhores avaliações de *fitness*. Observa-se então que este processo se assemelha a uma pesquisa *structured random*.

Após a fase de seleção, um único modelo é considerado sendo realizadas operações de reprodução e combinação genética para gerar nova descendência. Repetindo-se o processo descrito mas agora com a melhor solução observada da iteração imediatamente anterior.

Tipicamente, no contexto de otimização genética, são inseridas mutações genéticas por forma a aumentar a variedade da descendência gerada. Este processo é iterativo até que uma rede com valor de *fitness* favorável seja gerada.

Numa abordagem convencional de NE, os pesos da rede neuronal são afinados e otimizados considerando-se a arquitetura da rede fixa, o que facilita a implementação do método em diferentes áreas [3].

Quando é pertinente considerar a modificação da estrutura da rede para obtenção de melhores resultados, outro tipo de abordagens são consideradas. Um poderoso método desenvolvido por Stanley e Miikkulainen denominado *NeuroEvolution of Augmenting Topologies* (NEAT) demonstra as vantagens das alterações da topologia para que seja alcançado um balanceamento entre o *fitness* das soluções obtidas e a sua diversidade. Desta forma, reduz-se o espaço de pesquisa e consequentemente são gerados ganhos ao nível da velocidade de aprendizagem [43].

A literatura vem utilizando algoritmos de NE em diversas áreas de aplicação. Christian Igel [44], recorrendo a estratégias evolutivas para adaptação dos pesos das RN em tarefas de RL, obteve melhores resultados no que respeita ao problema do pêndulo invertido face a outras abordagens que adaptam simultaneamente o peso e a própria estrutura da rede.

Filipe Such et al.[41] demonstraram que Algoritmos Genéticos podem ser uma solução rápida no treino de redes neurais para jogarem *ATARI games* - jogos que mapeiam diretamente pixels em ações. Os autores documentaram também a competitividade deste tipo de abordagens na atualização de pesos de redes neurais face a algoritmos mais populares enaltecendo a vantagem que advém da aliança deste método às elevadas capacidades de processamento dos tempos modernos. Rogier Koppejan e Shimon Whiteson [45] recorreram a técnicas de NE para demonstrar que estas podem ser eficazes na descoberta de políticas eficientes no que respeita ao controlo de um helicóptero face a diferentes condições ambientais a que são sujeitos.

#### 2.6.4 Policy Gradients

Métodos baseados em *Policy Gradients* (PG) são algoritmos de RL que mantêm uma política parametrizada ação-seleção e atualizam os parâmetros da política movendo-os na direção de um gradiente estimado a partir de uma medida de performance baseada num conceito de recompensas [46]. Em métodos baseados em políticas a aprendizagem não incide sobre uma função de valor, que indica o que é expectável no que respeita a um acumular de recompensas, mas sobre aprendizagem e otimização de políticas que mapeiam estados em ações. Uma função de valor pode, também, ser utilizada na aprendizagem do melhor parâmetro de uma política, mas não é necessária para a decisão acerca da ação a tomar [2].

A utilização de métodos PG evidencia um conjunto de vantagens:

- Os algoritmos baseados nestes métodos apresentam melhores propriedades de convergência [4]. Se por um lado métodos baseados em valor podem apresentar grandes oscilações durante o treino dada a ação a seguir poder ser alterada dramaticamente devido a pequenas mudanças na estimação de valores de ação, com PG é seguido o gradiente para serem encontrados os melhores parâmetros. Suavizando a atualização da política a cada passo é garantida a convergência para um máximo local (no pior caso) ou um máximo global (no melhor caso) [4].

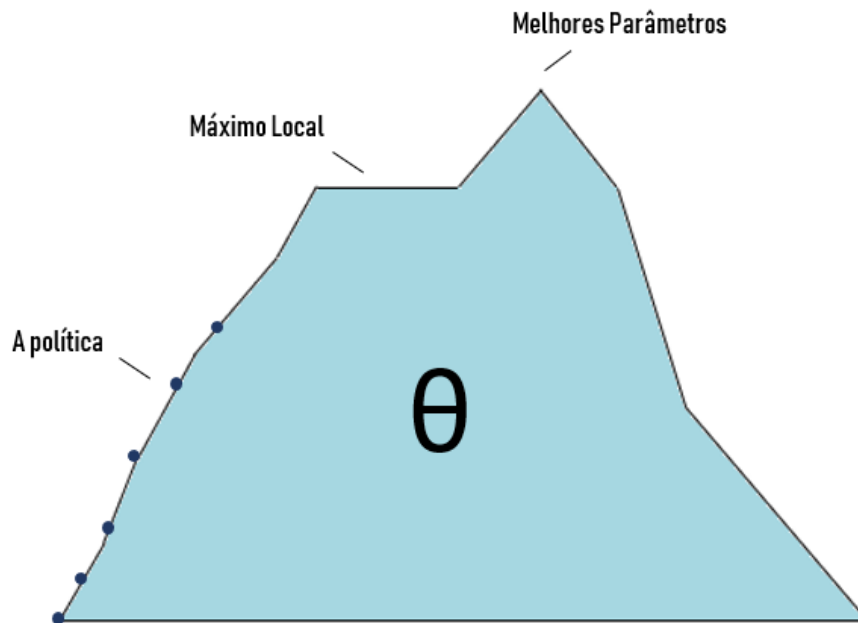


Figura 2.11: Procura dos melhores parâmetros seguindo o gradiente, adaptado de [4]

- Outra vantagem evidenciada vai ao encontro da maior eficiência dos métodos PG em espaços de ação de elevadas dimensões. Se por um lado, políticas baseadas em valor estimam, computacionalmente, o máximo acumular de recompensas a cada passo, métodos baseados em PG melhoram a política a cada passo ajustando os parâmetros na direção do gradiente estimado [46].
- Uma terceira vantagem é a capacidade destes métodos serem capazes de aprender políticas estocásticas, permitindo que o agente seja capaz de explorar o espaço de estados sem que seja tomada sempre a mesma ação. Isto resulta do facto de serem distribuídas probabilidades de decisão sobre diferentes ações a tomar não limitando o conjunto de ações por explorar.

#### 2.6.4.1 Pesquisa da política

PG indicam como deve ocorrer a mudança na política alterando um parâmetro  $\theta$  de forma a ser obtido um maior acumulado de recompensas. Isto ocorre aumentando, sistematicamente, a probabilidade de opção por ações que potenciem esse acumular de recompensas e, simultaneamente, diminuindo a probabilidade de seleção de ações que reflitam um resultado contrário.

Partindo de uma política  $\pi$  e um parâmetro  $\theta$  está associada uma distribuição de probabilidades de escolha sobre uma determinada ação  $a$  dado um estado  $s$ .

$$\pi_{\theta}(a|s) = P[a|s] \quad (2.10)$$

Considerando a otimização da política, o objetivo passa por encontrar o melhor parâmetro que permita maximizar um acumular de recompensas  $J(\theta)$ .

Para avaliação do quão positiva é a política recorre-se à função objetivo  $J(\theta)$  que calcula o acumular de recompensas esperado pela política. Partindo de um conjunto de episódios e causando um oscilação  $\Delta\theta$  nos parâmetros da política é verificada uma variação  $\Delta R$  no aglomerado de recompensas geradas.

Podendo a variação ser aproximada pela seguinte expressão:

$$\Delta R \approx \nabla_{\theta} J(\theta) * \Delta\theta \quad (2.11)$$

Sendo  $\nabla_{\theta} J(\theta)$  o gradiente do valor de recompensa em relação aos parâmetros da política.

Considerando que o vetor gradiente aponta na direção de maior crescimento no que respeita a um acumular de recompensas, alterando o parâmetro  $\theta$  na direção do gradiente, com um *step* pequeno o suficiente é garantido um maior acumular de recompensas.

Se iterativamente se atualizar o parâmetro de acordo com a seguinte expressão

$$\theta_{k+1} = \theta_k + \alpha_k * \nabla_{\theta} J(\theta) \quad (2.12)$$

é garantida a convergência, no pior caso, para um máximo local [47].

Métodos baseados em PG vêm sendo aplicados em diversas áreas. Nate Kohl e Peter Stone [48] recorreram a técnicas de RL baseadas em PG para automaticamente definirem um conjunto de possíveis parâmetros de forma a encontrarem a maneira mais rápida de andamento de um *robot* de quatro pés. Em apenas 3 horas os resultados obtidos, sem interação humana, mostraram ser melhores que o conjunto de soluções pré-existentes.

Num contexto mais atual, a OpenAI [49] desenvolveu um sistema inteligente capaz de aprender utilizando um método denominado *Proximal Policy Optimization* (PPO) que se insere na família de métodos PG. Este método foi aplicado no treino de uma equipa de cinco *bots* modelados por redes neurais para que aprendessem a jogar DOTA2. O método mostrou-se bastante eficaz demonstrando ser capaz de derrotar alguns dos melhores jogadores do mundo do referido videojogo.

## 2.7 Simulação de Eventos Discretos

A simulação baseada em eventos discretos representa a modelação, simulação e análise de sistemas utilizando técnicas matemáticas e computacionais que permitem descrever um sistema e o seu comportamento [50]. Pela capacidade de representar quantitativamente ambientes reais, simulando as dinâmicas de um determinado sistema partindo de uma ideia baseada em eventos [50],

este método de simulação revela-se uma poderosa ferramenta de análise capaz de avaliar fluxos produtivos, desde as atividades envolvidas ao impacto das mesmas na globalidade do processo. Considerando este ponto, este tipo de simulação pode ser utilizado como ferramenta de auxílio para as empresas, avaliando que tipo de repercussões pode uma determinada ação causar sem interferir, realmente, no processo da empresa. Métodos de simulação podem então ser considerados como uma das principais ferramentas de apoio à decisão face ao elevado poder computacional dos tempos modernos [50].

Associados a este tipo de simulação estão um conjunto de conceitos que permitem perceber melhor o funcionamento deste tipo de softwares de auxílio à tomada de decisão [9]:

- Sistema: Coleção de entidades (e.g., pessoas ou máquinas) que interagem entre si para atingirem um determinado objetivo.
- Modelo: Representação abstrata do sistema, contendo relações lógicas, estruturais ou matemáticas capazes de descrever o sistema em termos de estado, atributos, processos ou atividades.
- Estado do Sistema: Representa o conjunto de variáveis capazes de descrever o sistema num determinado instante de tempo.
- Entidade: Qualquer componente do sistema que requer uma representação explícita no modelo (e.g., máquinas, clientes).
- Atributos: Propriedades de uma dada entidade.
- Evento: Ocorrência instantânea que muda o estado do sistema.
- Atividade: Duração de tempo específica em que é conhecido o seu momento de início.

Um modelo de simulação baseado em eventos discretos pode ser definido pelo conjunto de ações resultantes dos valores das variáveis de estado em momentos temporais discretos onde é verificada a ocorrência de um evento, consequente de uma atividade. As entidades competem por recursos, ou aguardam pela disponibilização dos mesmos até à libertação por parte de outras entidades.

Percebendo os conceitos anteriormente explicitados é possível analisar que o desenvolvimento ao nível de software para este tipo de simulação evoluiu progressivamente desde 1960. Se por um lado, inicialmente, o modelador do sistema era obrigado a programar todo o modelo lógico e o código para controlar os eventos e atividades tornando demorada e complexa cada situação a simular, atualmente, um conjunto de "packages" permite que os modelos sejam modificados a qualquer momento acelerando o tipo de análise "what-if". Este cenário confere uma maior abrangência quanto ao número de situações a analisar, facilitando a chegada a situações mais benéficas segundo um determinado conjunto de pontos definidos pelas empresas [50].

Dado o carácter flexível deste tipo de ambientes de simulação, progressivamente se verifica um maior recurso a este tipo de técnicas em ambientes produtivos. Cada vez mais, as empresas

investem abundantemente para tornarem os seus processos produtivos mais flexíveis, como é o caso da definição de rotas dos AGVs, naquilo que respeita ao transporte de materiais. Desta forma, dado o conjunto de cenários disponibilizados pelo ambiente de simulação é possível chegar a um processo de transporte mais inteligente, reduzindo *lead times*, o WIP e aumentando a taxa de utilização de cada recurso, resultando numa maior produtividade para as empresas que adotam este tipo de abordagens [50].

### 2.7.1 Vantagens da Simulação Baseada em Eventos Discretos

Tomando como base o mencionado anteriormente, inúmeras vantagens podem ser evidenciadas do recurso a técnicas de simulação:

- Permite o estudo e análise de sistemas de elevada complexidade;
- Permite avaliar a ocorrência de um determinado fenómeno e inferir o responsável pelo mesmo;
- Permite que sejam testados novos *layouts*, sistemas de transporte ou outras modificações ao processo produtivo por forma a tirar conclusões e delinear ações de melhoria sem que seja necessário despender recursos na aquisição de material;
- Avaliando diferentes circunstâncias de simulação alterando entradas e verificando o resultado consequente é possível obter informações valiosas;
- Permite o manuseamento do tempo, podendo este ser acelerado para ver com mais detalhe uma ação ou um determinado conjunto de ações.

### 2.7.2 Desvantagens da Simulação Baseada em Eventos Discretos

Apesar das vantagens, este tipo de modelação também tem desvantagens associadas:

- É necessário treino e *expertise* para a construção de modelos de simulação;
- É passível de existirem entradas de carácter aleatório num modelo de simulação e os resultados observados podem resultar exatamente dessas aleatoriedades expressando falsos resultados;
- Dada a complexidade dos sistemas a simular, estes processos podem ser muito dispendiosos em fatores monetários e temporais.

Apesar das desvantagens previamente mencionadas, por forma a garantir competitividade deve existir uma procura pela inovação ou por um melhor aproveitamento dos recursos e é neste contexto que os modelos de simulação se revelam extremamente importantes, já que permitem a avaliação de um conjunto de cenários enorme que de outra forma não poderiam ser explorados.

Através da literatura existente, é possível verificar a aplicação de métodos de simulação na indústria, mais concretamente a sua utilização em problemas relativos ao transporte de materiais. Korytkowski e Karkoszka [51] estudaram, recorrendo a métodos de simulação, a eficiência, interações e impacto de um modelo de transporte *Milk Run* aplicado a uma linha de produção tradicional. Desta forma, os autores foram capazes de avaliar, rapidamente, o desempenho do sistema produtivo segundo um vasto conjunto de indicadores. Jeffrey S. et al. [52] aplicaram métodos de simulação baseada em eventos discretos para controlo do chão de fábrica de um sistema de produção. Os autores recorreram a estes métodos, não só como ferramenta de análise de evolução mas, também, como elemento de controlo. Simulando um conjunto de diferentes ordens produtivas e analisando os resultados consequentes foi possível obter previsões de desempenho do sistema produtivo de elevada fiabilidade e gerar o conjunto de tarefas adequadas a cada situação verificada.

## 2.8 Considerações Finais

Face às temáticas abordadas neste capítulo é possível retirar um conjunto de conclusões. Dada a complexidade associada aos sistemas de produção atuais e aos sistemas logísticos adjacentes é possível afirmar que técnicas de RL podem ser uma alternativa interessante de explorar em busca de melhores soluções ao nível dos sistemas de transporte.

Uma vez que a capacidade de processamento tem vindo a crescer, a utilização da técnica *Neuroevolution* desperta especial interesse. Considerados os resultados apresentados pela literatura resultantes deste método, este é visto como uma boa opção a explorar para resolver problemas ao nível da logística *in-plant*. Em virtude da necessidade de existência de um ambiente para que o algoritmo possa ser avaliado e do conjunto de vantagens associadas à simulação de eventos discretos, este método assume especial relevância dada a sua capacidade de replicar com elevada credibilidade características dos sistemas de produção reais.



## Capítulo 3

# Problema e Método de Abordagem

Este capítulo descreve o problema e formula uma análise detalhada ao mesmo por forma a definir a informação necessária para a estruturação de um sistema de transporte de materiais (MHS) otimizado baseado em técnicas de RL. Neste capítulo estão também detalhadas características gerais do sistema de produção idealizado para o desenvolvimento do sistema de transporte de materiais, assim como uma definição dos elementos de simulação gerais ao conjunto de implementações a serem executadas.

### 3.1 Descrição do Problema

O problema em estudo incide sobre a estruturação de um novo sistema logístico flexível capaz de responder à constante dinâmica dos sistemas de produção atuais e maximizar índices de produtividade adjacentes. O sistema de produção em que atua o sistema logístico a desenvolver apresenta um conjunto de características aproximadas àquilo que se verifica em ambientes de produção reais. O problema pode ser definido por um conjunto de 5 tipos de peças distintas a serem produzidas num ambiente de produção constituído por 7 máquinas, também diferentes entre si, e ainda duas estações extra responsáveis por introduzirem e retirarem peças do sistema assim que se dê por terminada a sequência de tarefas associada. Partindo de um conjunto de ordens de fabrico pré-existentes é desencadeada a produção de um conjunto de diferentes produtos, sendo que cada produto possui uma sequência de produção e tempo de processamento característicos:

- Cada máquina apresenta tempos de processamento adequados ao tipo de produto que processa;
- Cada produto é apenas processado no conjunto de máquinas definidas na sua sequência de produção.

O transporte será realizado com recurso a um AGV que deve ser capaz de mostrar um comportamento inteligente nas suas deslocações. Desta forma, quando desencadeada a produção, é importante estabelecer uma rotina de transporte adequada que maximize a produtividade do sistema produtivo. Assim, surge a necessidade de explorar e definir um sistema de transporte, de

características otimizadas, para o conjunto de ordens produtivas que são lançadas no sistema. A otimização no fluxo de transporte passa pela definição da rota a ser percorrida, à decisão sobre o tipo de produto a transportar a cada momento e tem como objetivo aumentar a produtividade do sistema de produção sem que sejam necessárias reestruturações ao nível do *layout*. Para isso, é importante explorar técnicas de RL que sejam capazes de definir o conjunto de ações que devem ser tomadas pelo veículo de transporte, a cada momento, com base na experiência obtida.

## 3.2 Características Gerais do Problema

### 3.2.1 Definição do Layout

A planta da fábrica encontra-se organizada de forma a que as estações de trabalho adjacentes estejam equidistantes.

Cada estação de trabalho contém dois *buffers*: *buffer* de entrada - responsável pela retenção das peças antes de estas serem sujeitas a processamento - e um *buffer* de saída - responsável pela acumulação das peças que aguardam transporte.

O AGV responsável pela movimentação das peças ao longo do sistema atravessa, em primeira instância, o *buffer* de entrada seguido do *buffer* de saída. Desta forma, permite uma libertação de recursos numa dada estação e recolha das peças que aguardam transporte na mesma, caso se verifique alguma indicação neste sentido. Se não existirem indicações neste sentido o veículo de transporte segue o seu percurso. Como possui capacidade de transporte limitada, se esta é atingida, as peças no *buffer* de saída de qualquer estação de trabalho irão manter-se nesse estado até que o veículo por elas passe com capacidade para as transportar.

Pelo facto do *layout* se poder revelar um fator capaz de limitar a produtividade será importante avaliar a solução atual, representada na Figura 3.1 para definir um ponto de comparação válido para o sistema de transporte a idealizar.

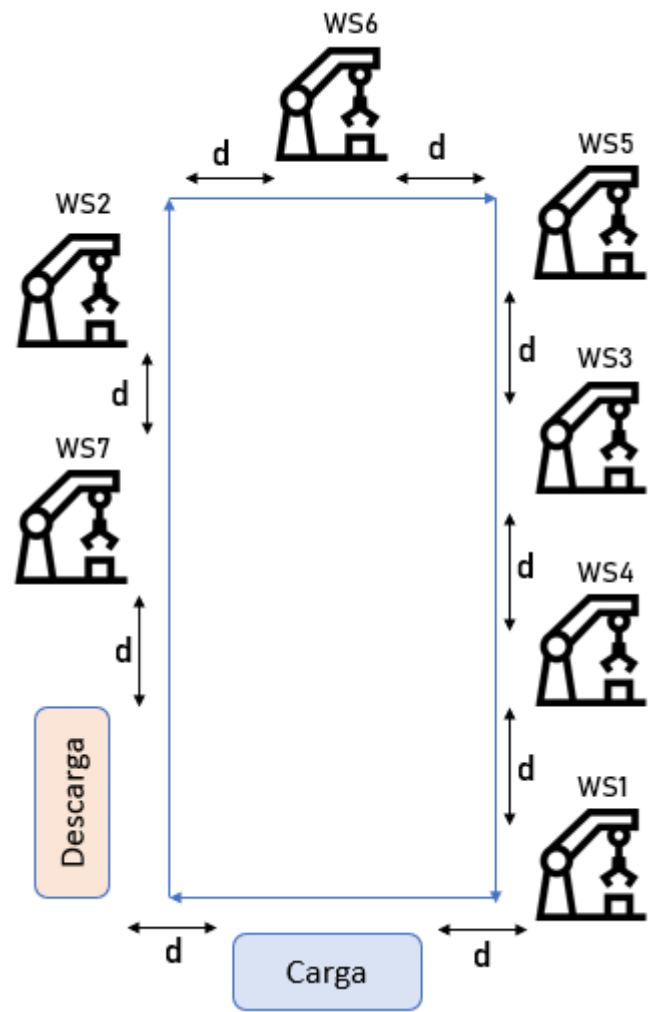


Figura 3.1: Esquema Geral do Sistema de Produção

3.2.2 Processamento e sequenciamento

Assim que dá entrada no sistema de produção, cada tipo de peça contém associado a si uma lista que define a sequência de máquinas em que deve ser processada e o tempo de processamento em cada máquina. Esta informação encontra-se apresentada nas Tabelas 3.1 e 3.2, respetivamente.

Tabela 3.1: Sequenciamento para cada tipo de peça

|       | Sequenciamento  |
|-------|-----------------|
| Peça1 | WS2 WS1 WS7 WS6 |
| Peça2 | WS4 WS2 WS5 WS1 |
| Peça3 | WS2 WS4 WS7 WS3 |
| Peça4 | WS7 WS3 WS2 WS4 |
| Peça5 | WS4 WS5 WS1 WS2 |

Tabela 3.2: Tempo de processamento para cada tipo de peça

|              | WS1 | WS2 | WS3 | WS4 | WS5 | WS6 | WS7 |
|--------------|-----|-----|-----|-----|-----|-----|-----|
| <b>Peça1</b> | 9   | 2   | -   | -   | -   | 8   | 7   |
| <b>Peça2</b> | 5   | 9   | -   | 10  | 5   | -   | -   |
| <b>Peça3</b> | -   | 6   | 7   | 10  | -   | -   | 7   |
| <b>Peça4</b> | -   | 7   | 7   | 5   | -   | -   | 3   |
| <b>Peça5</b> | 9   | 2   | -   | 10  | 6   | -   | -   |

Cada máquina processa uma única peça a cada momento. A ordem de processamento segue uma lógica FIFO (First-In-First-Out). Desta forma, a próxima peça a ser processada em cada estação de trabalho é definida pela sua ordem de chegada ao *buffer* de entrada.

### 3.2.3 Ordem de produção

A entrada de peças no sistema é realizada de forma periódica. Desta forma, a cada 90 segundos dará entrada uma nova peça no sistema.

Cada tipo de peça dá entrada no sistema segundo uma ordem pré-definida.

## 3.3 Simulação de eventos discretos - Modelo e Características Base

Uma vez que é pretendido avaliar o comportamento geral do sistema a desenvolver, um modelo de simulação baseado em eventos discretos é importante dada a sua capacidade de representar as principais características dos sistemas de produção. Este tipo de ferramenta revela-se útil para análise do comportamento dos sistemas de transporte, descritos com mais detalhe nos Capítulos 4 e 5. Por seu lado, este capítulo expressa apenas os pontos da metodologia comuns entre os sistemas de transporte a serem desenvolvidos, como evidenciado na Figura 1.1.

O SIMIO, como ferramenta de simulação capaz de representar modelos caracterizadores do sistema, permite não só avaliar resultados referentes a cada modelo, mas também visualizar o seu comportamento através de animações visuais em duas ou três dimensões. No contexto da abordagem a seguir, o SIMIO permite uma modelação das estações de trabalho, trajetos entre elas, assim como uma definição de *como e por quem* é realizado o transporte de materiais. Além do mencionado, é capaz de modelar os comportamentos ao nível de sequenciamento e processamento pretendidos.

Para cada modelo simulado pode ser definido um tempo de simulação fixo ou dependente de um *trigger* pré-definido pelo utilizador, permitindo avaliar de que forma alterações nos parâmetros de cada objeto afetam o comportamento do modelo.

### 3.3.1 Entidades

Este tipo de objetos, denominados pelo software como "*ModelEntity*" (Figura 3.2), são capazes de modelar o comportamento de uma peça e assumir o conjunto de características pretendidas para as mesmas, desde o conhecimento do estado de processamento da sequência total em que se

encontram à definição sobre a próxima estação de trabalho em que devem ser processadas. De notar a flexibilidade deste tipo de entidades que, a partir de apenas um objeto, permitem a adição de símbolos conferindo uma capacidade de modelação relativa a todos os tipos de peças que devem dar entrada no sistema.

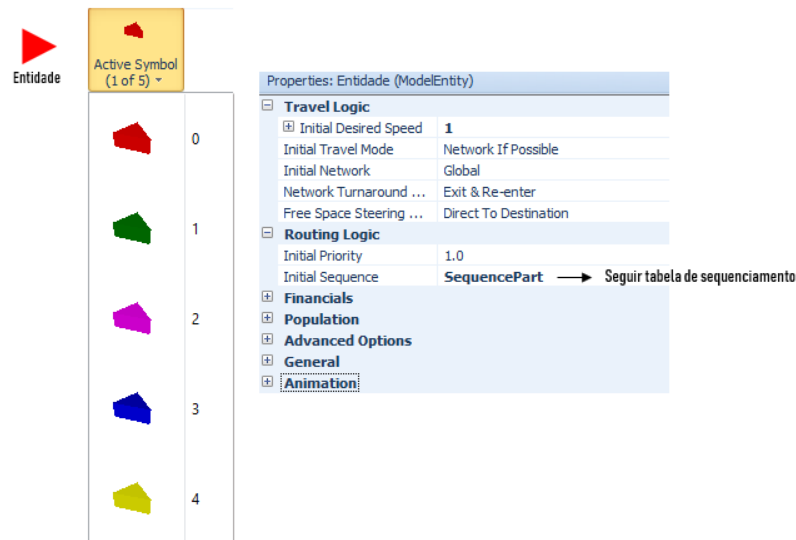


Figura 3.2: Exemplo de uma Entidade no SIMIO e conjunto de propriedades associadas

### 3.3.2 Estação de Trabalho

A estação de trabalho é definida no simulador por "Server" (Figura 3.3), este representa o objeto responsável pelo processamento de cada tipo de peça. A este objeto estão associados os respetivos *buffers* de entrada e saída assim como a lógica de processamento para cada peça que assim o necessite. Na estação de trabalho simulada, é permitida tanto uma definição de tamanho limite dos *buffers*, assim como a definição do número de peças passíveis de serem processadas em simultâneo, que para o caso em estudo será limitado a 1 peça.

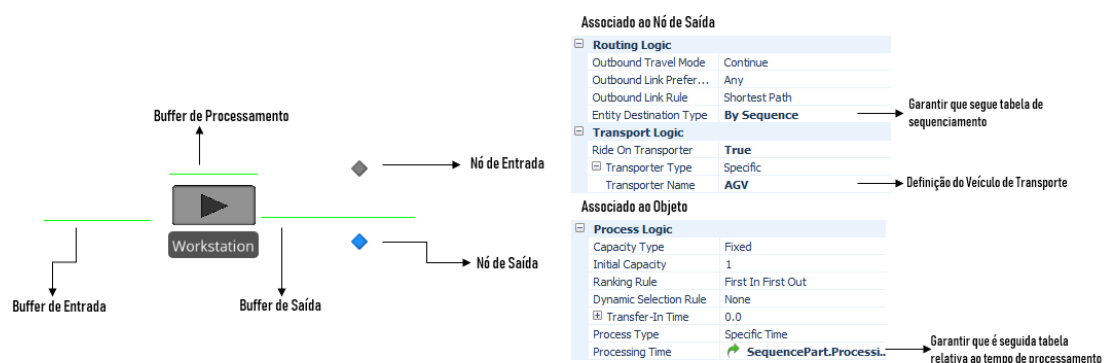


Figura 3.3: Exemplo de um objeto do tipo *Server* e conjunto de propriedades associadas

### 3.3.3 Trajetos

Trajetos, do ponto de vista do simulador, refletem caminhos válidos de ligação entre os objetos simulados. Estes podem ser caracterizados em termos da sua distância e do sentido em que permitem a movimentação de entidades, transportadas por AGV, sobre si próprios.

### 3.3.4 Carga

A entrada de entidades no sistema é definida por um objeto do tipo "Source" (Figura 3.4). Este objeto é capaz de modelar tanto o local em que é dada a entrada de uma peça no sistema (Carga), como a lógica de entrada de peças no sistema. A *Source* é capaz de seguir uma determinada ordem de produção e lançar uma peça para o sistema apenas no momento pretendido. A este tipo de objetos está apenas associado um *buffer* de saída que reflete a lista de espera das peças que aguardam para ser transportadas.

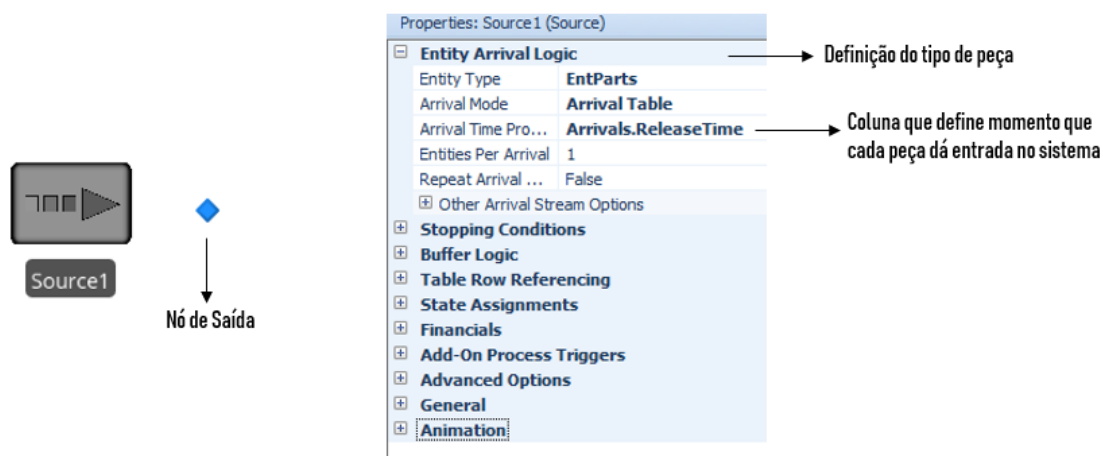


Figura 3.4: Exemplo de um objeto do tipo *Source* e conjunto de propriedades associadas

### 3.3.5 Descarga

A saída de entidades do sistema é definida por um objeto do tipo "Sink" (Figura 3.5). Este objeto reflete o último ponto respeitante à rota de cada peça simulada, acabando, do ponto de vista do simulador, por ser visto não só como o local que elimina peças do sistema, mas também como um ponto com tempo de processamento nulo.

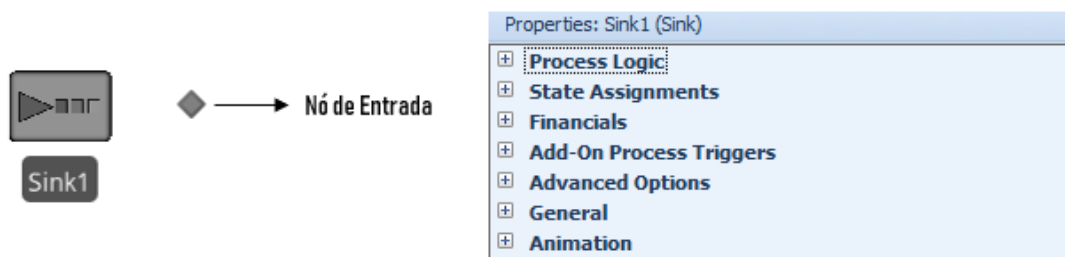


Figura 3.5: Exemplo de um objeto do tipo *Sink* e conjunto de propriedades associadas

### 3.3.6 Transporte

A movimentação de materiais ao longo do ambiente simulado é realizada com recurso a um objeto do tipo "Vehicle" (Figura 3.6). Este pode ter comportamentos pré-definidos distintos podendo a sua movimentação ser fixa, e portanto cíclica e periódica. O movimento do veículo pode reger-se segundo *triggers* externos, dependendo da definição por parte de outro objeto do sistema para definir o próximo local a se movimentar.

Este objeto possui um conjunto de parâmetros alteráveis (ver Figura 3.6) que permitem, rapidamente, modelar diferentes comportamentos e desta forma avaliar diferentes cenários.

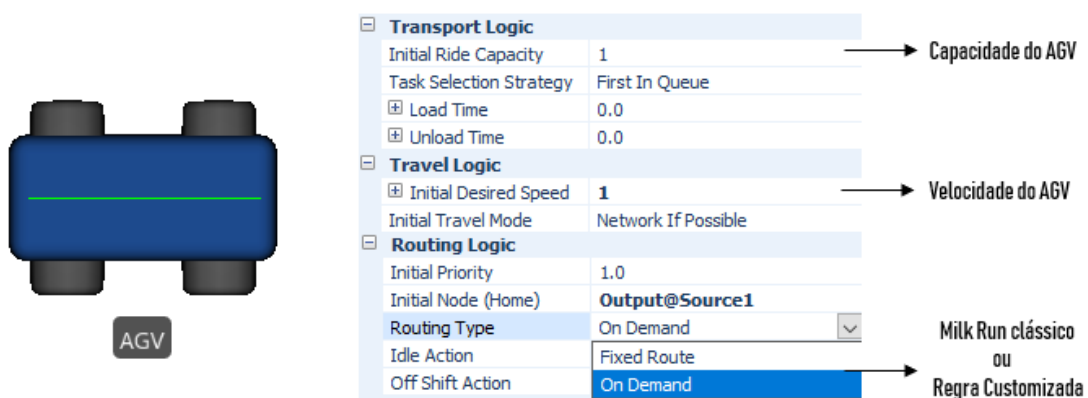


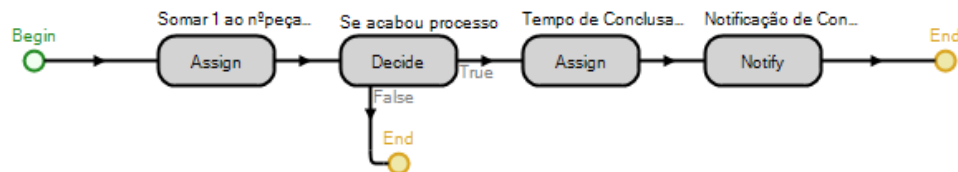
Figura 3.6: Exemplo de um objeto do tipo *Vehicle* e conjunto de propriedades associadas

### 3.3.7 Processos

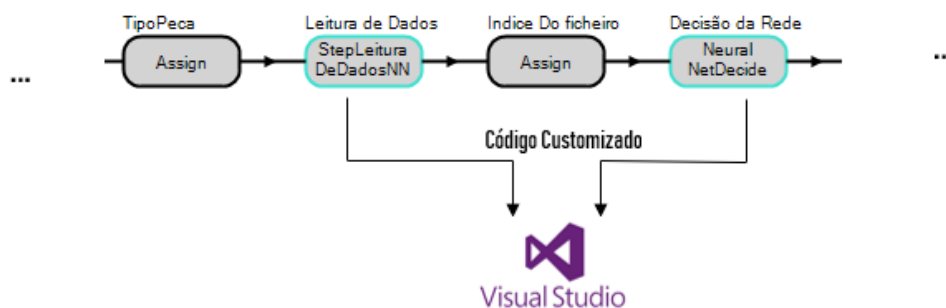
Um processo, do ponto de vista do simulador, é a ferramenta que permite modelar toda a lógica associada ao modelo. A estruturação de cada processo permite definir a sequência temporal em que cada ação acontece, que acontecimento despoleta o processo e de que forma o processo vai afetar o resto do modelo.

Cada processo é constituído por *steps* que podem realizar funções de atribuição, execução de outros processos, pesquisa, entre outras, destacando-se a flexibilidade de modelarem comportamentos definidos por código externo ao simulador via API (explorado com detalhe posteriormente), conferindo uma infinidade de aplicações para o software.

#### Count\_Destroyed\_Entities



#### Algoritmo



#### Algoritmo

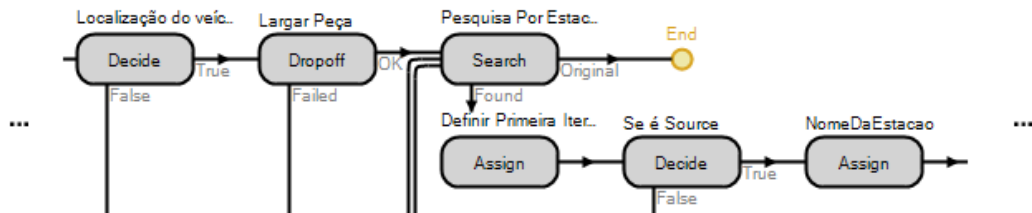


Figura 3.7: Exemplo de diferentes processos com diferentes etapas no SIMIO

### 3.3.7.1 Resultados

Uma característica importante do software utilizado é refletida pela sua capacidade de, a qualquer momento, permitir uma análise aos vários parâmetros relativos ao modelo do sistema de produção. A disponibilidade de filtragem de dados de acordo com o tipo de objeto(s) a simular facilita e acelera o processo de análise, assim como permite uma avaliação do que é realmente pertinente para o caso em estudo. Paralela a toda esta disponibilidade de informação, é fornecida, ainda, uma opção gráfica de visualização de dados segundo *dashboards*.



## Capítulo 4

# Implementação de um sistema de *Milk Run* clássico

Neste capítulo é demonstrada a metodologia de implementação de um sistema de transporte *Milk Run*, desde o pensamento sobre o *layout* à definição de características do AGV de transporte utilizado. O processo de transporte abordado neste capítulo é cíclico, garantindo que o AGV responsável segue uma rota fixa e pré-definida.

### 4.1 Otimização do Layout de Comparação

O *layout* inicialmente definido para o problema, tem a disposição evidenciada na figura 3.1. Este, no entanto, pode não se revelar o mais adequado para o conjunto de ordens de produção despoletadas no sistema.

Com o objetivo de viabilizar uma solução robusta para comparação com o sistema de transporte a desenvolver, é importante pensar na disposição das próprias estações de trabalho, como se verifica em grande parte dos ambientes industriais. Para isso, foi avaliada a sequência de trabalho de cada uma das ordens de produção.

Pretendendo definir a disposição ótima das estações de trabalho, foi estruturado o problema de otimização referido na literatura como Problema do Caixeiro Viajante. Neste caso, este procura, essencialmente, dispor estações de trabalho com maior número de interações diretas mais próximas entre si de forma a serem minimizados custos associados ao transporte e consequentemente aumentada a eficiência de cada ciclo do *Milk Run*.

#### 4.1.1 Conjuntos e Parâmetros

- $N$  Número de Estações
- $S$  Conjunto de Estações do Sistema Logístico,  $S = \{1, \dots, N\}$
- $s_{ij}$  Força da ligação entre a estação  $i$  e a estação  $j$

### 4.1.2 Variáveis de Decisão

$$x_{ij} = \begin{cases} 1, & \text{se existe uma ligação da estação } i \text{ para a estação } j \\ 0, & \text{caso contrário} \end{cases}$$

### 4.1.3 Restrições

1 - Qualquer estação é antecedida e sucedida por uma única estação

$$\sum_{i=1, i \neq j}^N x_{ij} = 1, \quad \forall j \in S \quad (4.1)$$

$$\sum_{j=1, j \neq i}^N x_{ij} = 1, \quad \forall i \in S \quad (4.2)$$

2- Um único circuito cobre todas as estações não permitindo a ocorrência de sub-circuitos

$$\sum_{i \in Q} \sum_{j \in Q} x_{ij} \leq |Q| - 1, \quad \forall Q \subseteq \{2, \dots, N\} \quad (4.3)$$

### 4.1.4 Função Objetivo

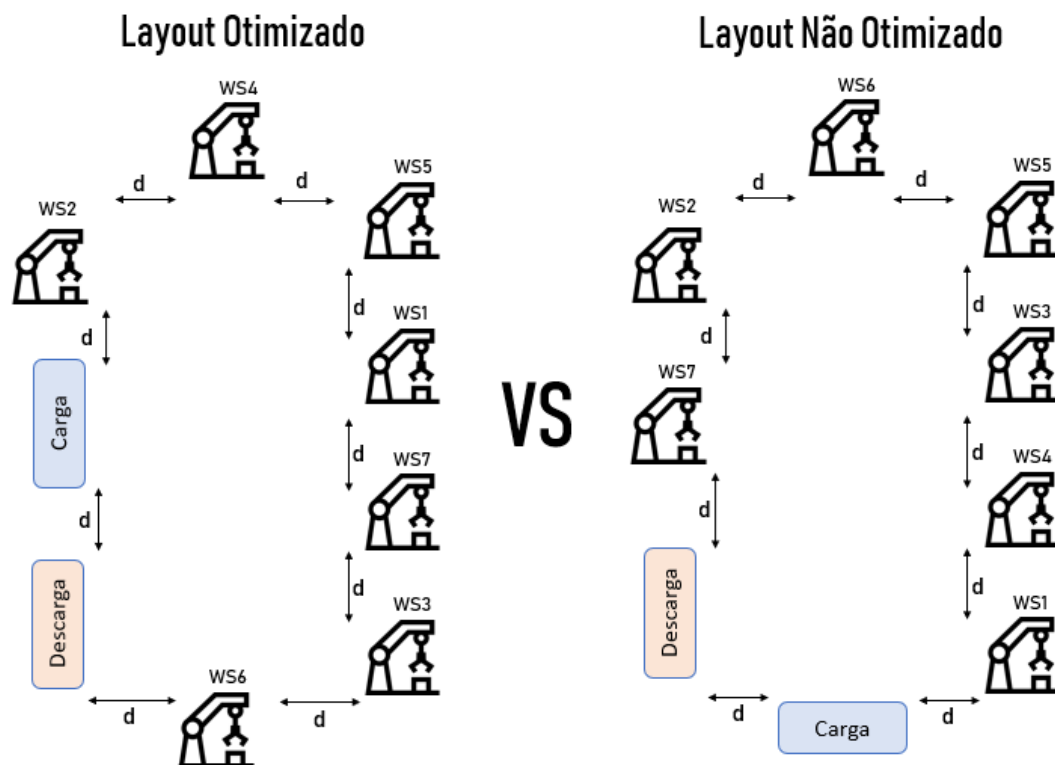
$$\max \sum_{i=1}^N \sum_{j=1, j \neq i}^N x_{ij} s_{ij} \quad (4.4)$$

Segundo o objetivo definido, recorrendo ao software CPLEX para modelar o problema obteve-se a seguinte disposição de estações de trabalho:

Tabela 4.1: Resultados do melhor *layout* para um sistema *Milk Run*

| De     | Para |
|--------|------|
| Source | WS2  |
| WS2    | WS4  |
| WS4    | WS5  |
| WS5    | WS1  |
| WS1    | WS7  |
| WS7    | WS3  |
| WS3    | WS6  |
| WS6    | Sink |

Obtida a sequência ótima das estações de trabalho, foram simulados os dois *layouts* de forma a corroborar, numa fase seguinte, a importância da própria organização das estações de trabalho através de uma análise aos dados relativos às duas disposições referidas que se encontram apresentadas na Figura 4.1.

Figura 4.1: Comparação entre *layouts*

Pelo facto de ser pretendido introduzir melhorias no sistema de transporte, é importante verificar que o sistema de transporte limita o próprio sistema produtivo. Com este propósito, foi seguido o seguinte encadeamento de ideias:

- Desenho no simulador do *layout* não otimizado com velocidade do veículo de transporte infinita, sendo que este segue o trajeto definido para o *Milk Run*.
- Fornecer ao sistema simulado uma sequência de produção conhecida e avaliar resultados segundo o critério de *Makespan*.
- Redefinir a velocidade do AGV para valores realistas e verificar se estes afetam o desempenho do sistema produtivo.
- Avaliar no *layout* otimizado o desempenho do sistema produtivo segundo os mesmos critérios e características do veículo de transporte.

Partindo do *layout* não otimizado definido na Figura 4.1, recorrendo ao software SIMIO, resultou o seguinte sistema de produção simulado:

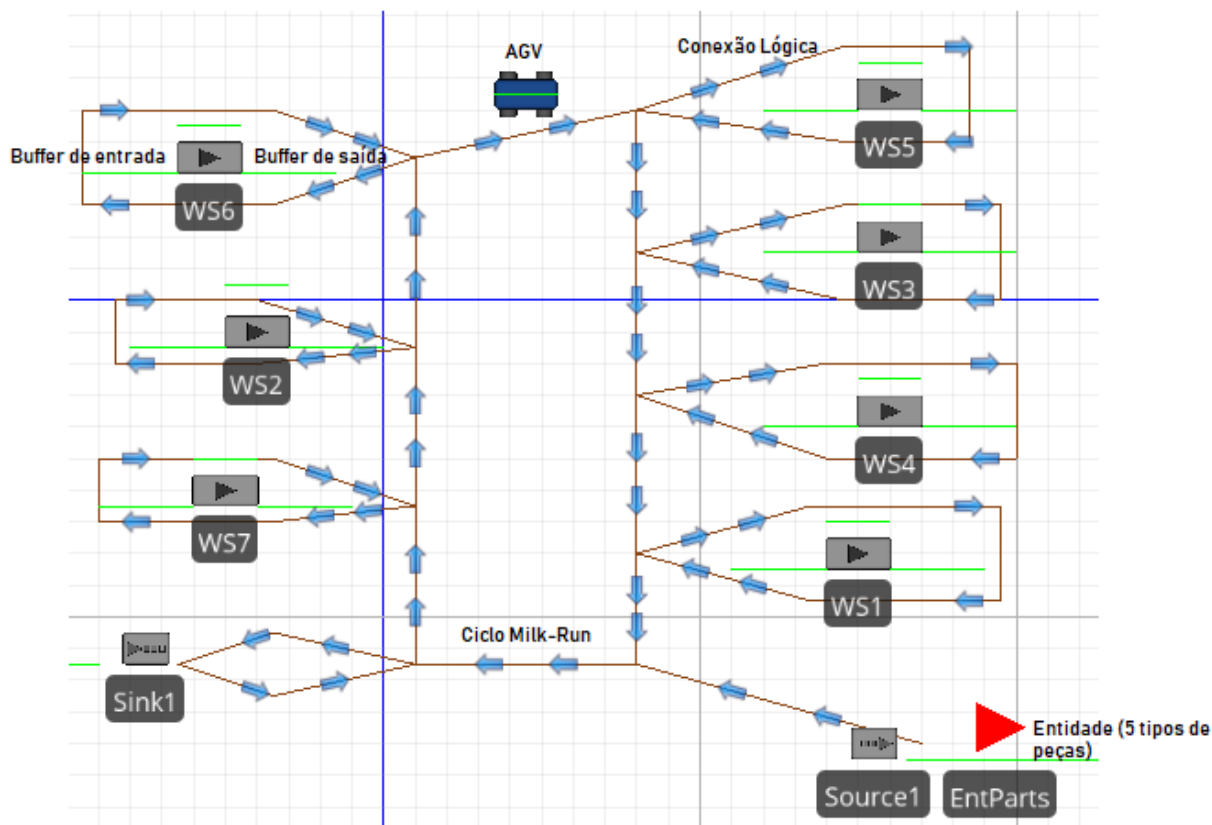


Figura 4.2: Layout não otimizado em ambiente de simulação

Definindo uma velocidade infinita ao AGV responsável por seguir o trajeto cíclico sobre as estações de trabalho é garantido que o sistema de transporte em nada limita a produção.

Pré-definindo uma ordem de produção de 1000 peças é possível avaliar, segundo o critério de *Makespan*, o tempo total que o sistema leva a finalizar o pedido se não existissem restrições ao nível de transporte.

Segundo as características apresentadas, avaliando os resultados do sistema simulado, foi verificada a saída da última peça do sistema após 113.88h.

Limitando a velocidade segundo valores pertinentes em contexto industrial para, respectivamente 1m/s e 0.7m/s foram observados os seguintes resultados para um tempo de simulação limitado a 113.88h:

Tabela 4.2: Layout Não Otimizado - Valores de Produtividade

| Velocidade | Nº Peças Produzidas |
|------------|---------------------|
| 1m/s       | 715                 |
| 0.7 m/s    | 499                 |

Sendo verificada uma diminuição nos índices de produtividade, é comprovado o comportamento do sistema de transporte como *bottleneck* do sistema produtivo.

Alterando no ambiente de simulação a disposição das estações de trabalho, segundo os resultados do modelo de otimização desenvolvido, foram verificados os valores de produtividade apresentados na Tabela 4.3.

Tabela 4.3: Layout Otimizado - Valores de Produtividade

| <b>Velocidade</b> | <b>Nº Peças Produzidas</b> |
|-------------------|----------------------------|
| <b>1 m/s</b>      | 996                        |
| <b>0.7 m/s</b>    | 980                        |

À medida que o veículo de transporte apresenta velocidades inferiores, este revela-se um fator mais limitativo ao sistema produtivo enfatizando a importância da definição de um *layout* adequado neste tipo de problemas.

Pelo conjunto de valores de produtividade apresentados pode-se observar uma melhoria significativa neste indicador, resultante da reformulação do *layout* efetuada.

Admite-se que uma empresa que disponha de um sistema *Milk Run* implementado tenha feito as devidas otimizações à disposição das estações de trabalho. Desta forma, faz sentido utilizar o *layout* otimizado na comparação entre o *Milk Run* clássico e o sistema de transporte dinâmico a desenvolver.

Tendo em conta que num contexto real o planeamento é feito em horizonte rolante, com novas ordens a surgirem frequentemente e originando um reescalonamento da produção, faz sentido utilizar um indicador de performance capaz de avaliar a produtividade do sistema num horizonte de tempo de mais curto prazo. Assim, como critério de comparação entre as diversas abordagens para a definição do comportamento do AGV, utilizar-se-á a produtividade num horizonte temporal de 36h e 52h. Estes valores para o *Milk Run* clássico estão apresentados segundo diferentes parâmetros de velocidade para o AGV na Tabela 4.4:

Tabela 4.4: Milk Run Clássico - Valores de Produtividade

| <b>Velocidade</b> | <b>Nº Peças Produzidas</b> |            |
|-------------------|----------------------------|------------|
|                   | <b>36h</b>                 | <b>52h</b> |
| <b>1.5 m/s</b>    | 298                        | 453        |
| <b>1 m/s</b>      | 328                        | 500        |
| <b>0.7 m/s</b>    | 297                        | 433        |

Desta forma, fica completamente definido o sistema *MilK Run* que servirá como primeiro ponto de avaliação para as soluções dinâmicas a serem desenvolvidas.



## Capítulo 5

# Implementação de um sistema de transporte dinâmico

Neste capítulo é demonstrada a metodologia de implementação de um sistema de transporte dinâmico, desde as abordagens heurísticas que utilizam dados à priori para a definição dos movimentos a serem realizados, à solução híbrida que deve ser capaz de ajustar o seu comportamento a diferentes variações associadas ao sistema de produção.

Após verificação dos resultados que o *Milk Run* tradicional é capaz de apresentar e antes de recorrer a um algoritmo de natureza evolutiva para o desenvolvimento de uma solução flexível para o problema, foram concebidas as diferentes heurísticas apresentadas abaixo.

Relativamente às características mencionadas no capítulo anterior, a única diferença significativa refere-se à liberdade de movimentos do AGV por toda a área livre entre as estações.

### 5.1 Heurística 1

#### 5.1.1 Descrição

Partindo das funcionalidades disponibilizadas pelo SIMIO, a heurística acarreta os pedidos segundo a ordem pela qual dão entrada.

Para desencadear o processo produtivo, um pedido de transporte é efetuado quando a primeira peça dá entrada no sistema. Os restantes pedidos para movimentação são efetuados no momento em que qualquer peça termina o seu processamento. O movimento após a recolha de uma peça é sempre dirigido ao próximo local designado na sua tabela de sequenciamento.

#### 5.1.2 Objetivo

Avaliação do impacto das movimentações pelo espaço livre segundo um indicador de produtividade.

### 5.1.3 Implementação

---

**Algoritmo 1: *First in First Out***


---

**Entrada:** Lista de pedidos de transporte

**Saída:** Seleção Da Peça a Transportar

**início**

**enquanto** *Ordem produção não terminada* **faça**

**se** *Contém pedidos na lista de pedidos* **então**

            Designar o primeiro pedido como aquele a ser satisfeito (lógica FIFO)

**fim**

**fim**

**fim**

---

### 5.1.4 Resultados

Tabela 5.1: Heurística 1 - Valores de Produtividade

| Velocidade     | Nº Peças Produzidas |     |
|----------------|---------------------|-----|
|                | 36h                 | 52h |
| <b>1.5 m/s</b> | 250                 | 365 |
| <b>1 m/s</b>   | 87                  | 186 |
| <b>0.7 m/s</b> | 42                  | 66  |

## 5.2 Heurística 2

### 5.2.1 Descrição

A heurística desenvolvida incide, principalmente, sobre dois parâmetros. O primeiro é relativo à distância a que o veículo de transporte se encontra de cada estação de trabalho, a cada momento. O segundo parâmetro é respeitante ao tempo associado a cada *buffer* de entrada, isto é, o tempo desde que uma peça dá entrada numa estação de trabalho até que, efetivamente, é processada. Este tempo é respetivo à soma dos tempos de processamento de cada peça que se encontra no *buffer* de entrada à espera de ser processada.

A heurística irá atuar em duas situações:

- Sempre que o veículo dá entrada num nó relativo a um objeto do modelo;
- Se não existir nenhuma peça num dos *buffers* de saída, a heurística atua assim que uma peça termina o seu processamento.

### 5.2.2 Objetivo

Num contexto *just-in-time*, para evitar que o AGV transporte peças que não possam ser imediatamente processadas é sempre verificado o horizonte temporal de processamento de cada máquina.



O local onde for verificado que uma peça é processada o mais cedo possível é então definido como o próximo local a entregar peça, se assim for possível. Desta forma, minimizam-se os estados de "starvation" relativos a cada estação.

### 5.2.3 Implementação

Recorrendo à ferramenta de construção de processos disponibilizada pelo simulador foi modelada a seguinte lógica para o modelo:

---

#### Algoritmo 2: Minimizar Tempo Dos *Buffers* De Entrada

---

**Entrada:** Distância do AGV para cada estação , Tempo Do *Input Buffer*

**Saída:** Seleção Da Estação A Movimentar

**início**

**para cada** *Estação de trabalho* **faça**

**se** *Contém peça no output buffer* **então**

            Avaliar o tempo do *buffer* de entrada da estação onde a peça será processada de seguida;

**fim**

**se** *Última estação na lista de estações && Não encontrou nenhuma peça* **então**

            Mantém a posição;

            Termina Processo;

**fim**

**fim**

**se** *Encontrou peças* **então**

        Selecionar a peça com menor tempo de espera associado como a próxima a ser transportada;

**fim**

**se** *Empate* **então**

        Avaliar a distância a que o veículo de transporte se encontra de cada uma das peças anteriormente selecionadas;

**para cada** *Estação de trabalho com tempo do input buffer mínimo* **faça**

            Avaliar o tempo do *buffer* de entrada da estação onde a peça será processada de seguida;

**fim**

**fim**

    Escolher estação mais próxima como o local a movimentar;

**fim**

---

## 5.2.4 Resultados

Tabela 5.2: Heurística 2 - Valores de Produtividade

| Velocidade | Nº Peças Produzidas |     |
|------------|---------------------|-----|
|            | 36h                 | 52h |
| 1.5 m/s    | 212                 | 326 |
| 1 m/s      | 97                  | 244 |
| 0.7 m/s    | 90                  | 123 |

## 5.3 Heurística 3

### 5.3.1 Descrição

A heurística visa replicar o comportamento do *Milk Run*, com a exceção do trajeto não ser cíclico. Desta forma, evitam-se movimentos para estações que não contenham peças para serem transportadas.

A partir do local onde se encontra o AGV é avaliado o conteúdo das estações subsequentes e opta-se pela movimentação para a estação mais próxima que contém material à espera de ser transportado.

A heurística irá atuar nas mesmas situações descritas para a *Heurística 2*.

### 5.3.2 Objetivo

Avaliar o impacto da minimização de movimentos em vazio.

Seguindo a heurística serão evitadas deslocações em vazio eliminando desperdícios associados ao sistema de transporte, quer do ponto de vista da utilização do AGV, assim como desperdícios relativos à distância percorrida pelo mesmo.

### 5.3.3 Implementação

---

**Algoritmo 3:** *Milk Run* com eliminação de deslocações em vazio

---

**Entrada:** Distância do AGV para cada estação

**Saída:** Seleção Da Estação A Movimentar

**início**

Ordenar estações por ordem de distância (incluindo a própria estação);

**enquanto** *Não tiver peça para transportar* **faça**

    Pesquisar no vetor de estações ordenadas por distância:

**se** *Contém peça no output buffer* **então**

        Definir a estação como local a movimentar;

**fim**

**senão**

        Pesquisar na próxima estação mais próxima;

**fim**

**se** *Última estação no vetor Ordenado && Não encontrou nenhuma peça* **então**

        Mantém a posição;

        Termina Processo;

**fim**

**fim**

**fim**

---

### 5.3.4 Resultados

Tabela 5.3: Heurística 3 - Valores de Produtividade

| Velocidade | Nº Peças Produzidas |     |
|------------|---------------------|-----|
|            | 36h                 | 52h |
| 1.5 m/s    | 265                 | 380 |
| 1 m/s      | 295                 | 441 |
| 0.7 m/s    | 334                 | 500 |

## 5.4 Neuroevolution (NE)

Considerando a complexidade associada aos problemas de transporte, é de enorme valor o desenvolvimento de uma solução flexível que seja capaz de se adaptar às diferentes características deste tipo de problemas.

Partindo da premissa associada ao *Reinforcement Learning*, aprendizagem a partir da experiência, irá recorrer-se a um algoritmo de origem evolutiva. A solução será desenvolvida através da interação de um agente (Rede neuronal que modela o comportamento de um AGV) com o ambiente (Sistema de produção simulado) tendo como base um conjunto de recompensas.

Como estratégia evolutiva irá recorrer-se a um algoritmo genético que deve ser capaz de gerar indivíduos (redes neuronais) cada vez mais competentes, com base numa função objetivo. A estratégia evolutiva passa por uma maximização de recompensas, sinónimo de soluções consequentemente de maior qualidade.

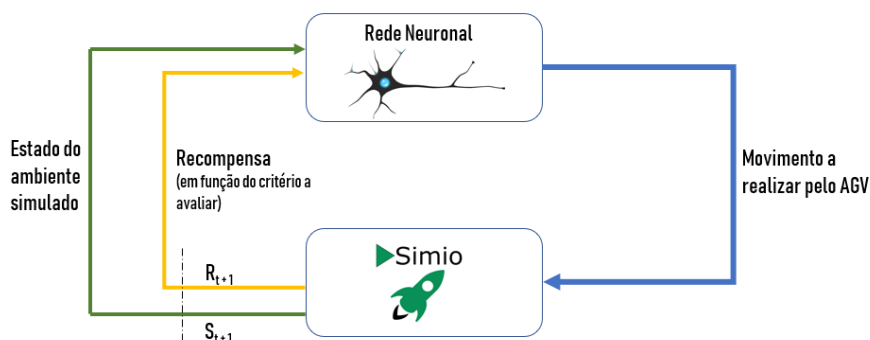


Figura 5.1: *Neuroevolution* e SIMIO num contexto de RL

#### 5.4.1 Rede Neuronal - Definição da Rede

Dada a complexidade do problema foi idealizada a necessidade de uma camada oculta para a definição de uma resposta não linear face aos dados de entrada para a rede.

A rede terá o formato apresentado na Figura 5.2, sendo que o número de neurónios da camada oculta será variável de acordo com a abordagem seguida.

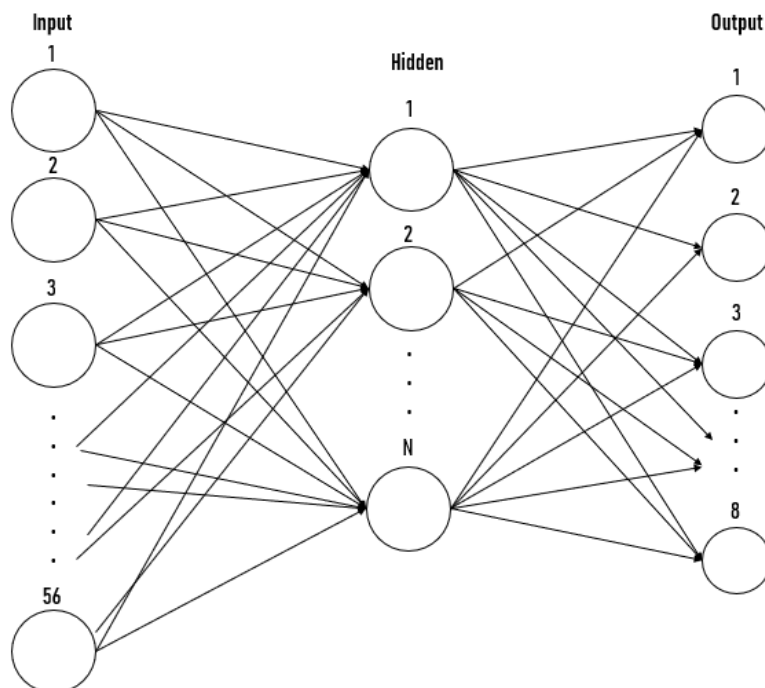


Figura 5.2: Rede Neuronal - Estrutura Idealizada

### Entradas

Os neurónios de entrada devem fornecer informação à rede que a permitam chegar a uma resposta em concordância com a função objetivo. A rede modela a resposta através de valores probabilísticos apresentados na camada de saída.

No contexto do problema, para auxílio na tomada de decisão sobre a estação a movimentar-se por parte do AGV, foram considerados 3 tipos de entradas obtidas e fornecidas pelo simulador:

- Tempo do *buffer* de entrada de cada estação + *source* (8);
- Distância do AGV a cada uma das estações + *source* + *sink*(9),
- Tipo de Peça no *buffer* de saída de cada estação, incluindo a *source* (8\*5)

O último tipo de entrada tem de ser representado com 5 neurónios de carácter binário. Desta forma, o segmento [0 0 0 0 1] significa que a estação tem uma peça do tipo 5. Se a responsabilidade de passagem deste tipo de informação fosse remetida a um neurónio com valores entre 1 e 5 estar-se-ia a dar uma informação falsa à rede. De forma errada, seria considerado que uma peça do tipo 5 teria uma importância diferente face a uma peça do tipo 2.

### Pesos

Os pesos representam os valores responsáveis por definir o tipo de influência que os neurónios de uma camada têm nos neurónios da camada seguinte. Estes estão compreendidos entre -1 e 1 aos quais se acrescenta também um peso de *bias* que existe para uma melhor adaptação, por parte da rede, à informação que lhe é transmitida.

### Saídas

A camada de saída terá 8 neurónios, representando cada neurónio uma estação para a qual o AGV se pode deslocar, sendo que o último neurónio diz respeito à *source*. O valor de cada neurónio estará compreendido entre 0 e 1 respeitando a sua função de ativação. A definição da estação a movimentar passa pela escolha do neurónio que apresenta valor probabilístico máximo.

### Função de Ativação

Com fundamento na literatura [39], a escolha da função de ativação a utilizar está dependente da camada.

Na camada oculta é utilizada a função RELU que pode ser modelada como:

$$Y = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (5.1)$$

Na camada de saída é utilizada a função de ativação Softmax. Esta pode ser representada como:

$$\theta(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, \quad j = 1, \dots, k \quad (5.2)$$

A função descrita transforma valores de saída em probabilidades entre 0 e 1. O maior valor de probabilidade, no contexto do problema, será a estação a deslocar. Quanto mais elevados os valores, maior a certeza sobre a decisão acerca da estação determinada para movimento.

### 5.4.2 Algoritmo Genético

Utilizando como base os conceitos definidos no Capítulo 2, naquilo que é o contexto do problema, foi desenvolvido o algoritmo genético descrito na Figura 5.3.

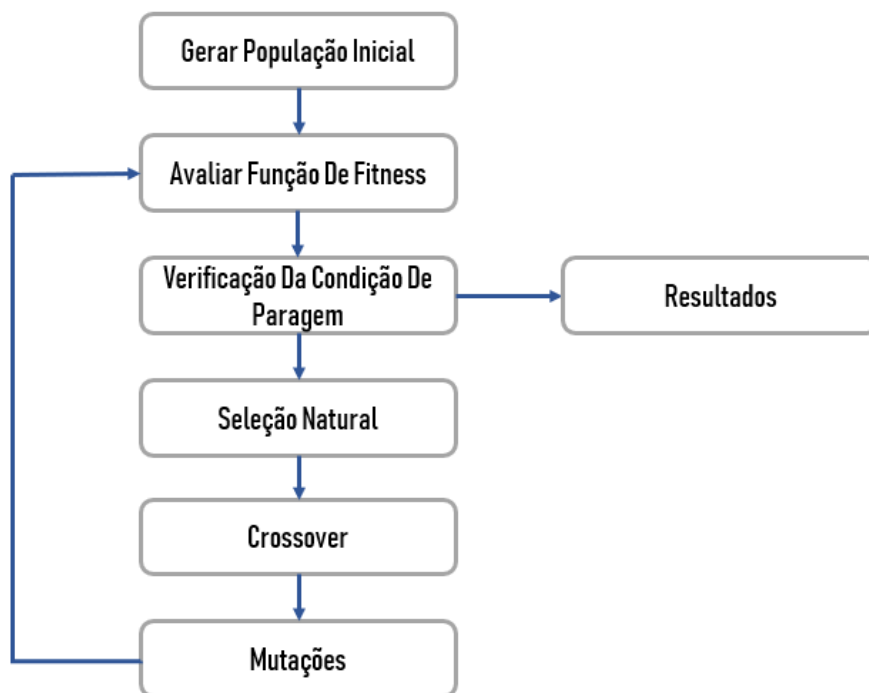


Figura 5.3: Algoritmo Genético - Implementação

#### População Inicial

Na definição da população inicial são gerados, aleatoriamente, 10 indivíduos que representam os pesos de cada uma das redes responsáveis por modelar o comportamento do AGV.

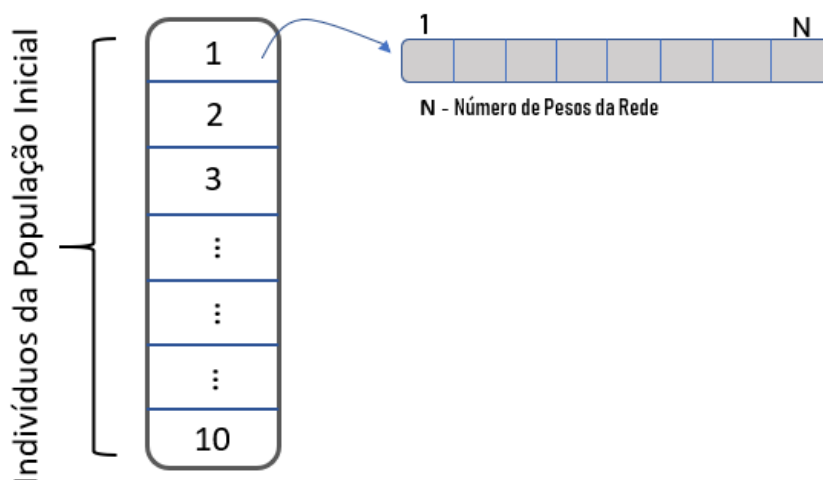


Figura 5.4: Definição da População Inicial

### Seleção Natural

Partindo da população existente, é avaliada a função de *fitness* de cada um dos indivíduos. Segundo um critério de maximização da produtividade para diferentes horizontes temporais e velocidades do AGV é atribuída, por parte do SIMIO, uma pontuação (recompensa) a cada indivíduo. Os 5 que apresentam melhores resultados são mantidos para as próximas etapas e iterações do algoritmo e os 5 piores indivíduos são descartados.



Figura 5.5: Método de Seleção Natural implementado

### Crossover

A partir das 5 melhores soluções obtidas no processo de seleção natural imediatamente anterior é gerada a descendência. Este processo parte da seleção aleatória de 2 das 5 melhores soluções e é repetido cinco vezes para que seja reposto o número de indivíduos pretendidos para a população. De seguida, utilizando um critério de combinação em dois pontos e para conferir maior variedade às novas gerações, cada ponto de corte é definido aleatoriamente [19]. Apenas uma restrição deve ser seguida - os pontos de corte devem estar definidos na primeira e segunda metade do código genético, respetivamente.

A descendência gerada contém, então, a primeira parte do  $Pai_1$ , a segunda parte do  $Pai_2$  e a terceira parte é, novamente, herdada do  $Pai_1$ .

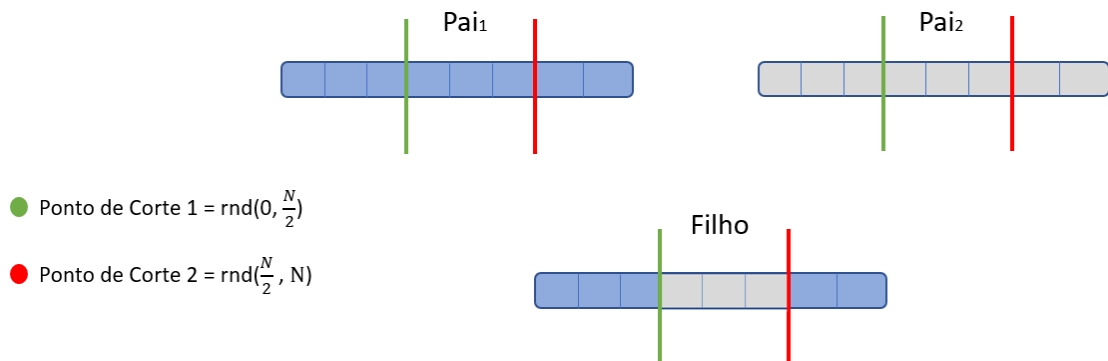


Figura 5.6: Método de *crossover* implementado entre dois indivíduos com melhor avaliação de *fitness*

### Mutações

O processo de mutação é fixo e garante divergência na descendência gerada assegurando que o espaço de procura não fica limitado pelo código genético da população inicial. Desta forma, a percentagem de pesos da rede a sofrerem mutações é de 10%. Os pesos mutados são todos relativos às dendrites de um mesmo neurónio antes de serem efetuadas mutações noutros neurónios [53]. A elevada taxa de mutações justifica-se pelo reduzido tamanho da população [19].

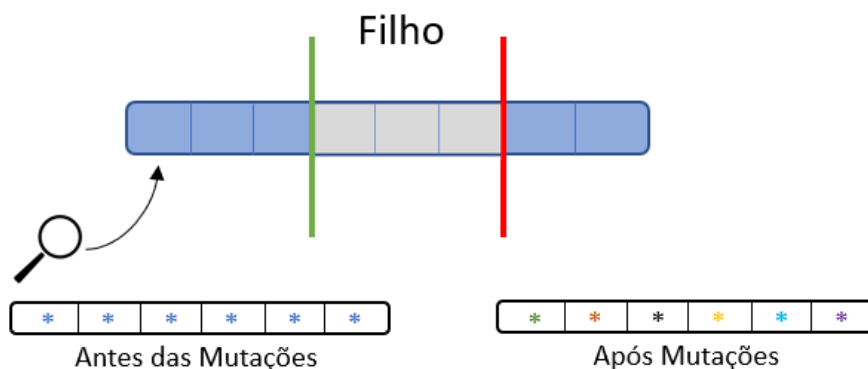


Figura 5.7: Método de mutação aplicado após processo de *crossover*

### Condição de Paragem

Em termos de eficiência, o critério de paragem é verificado quando durante 10 iterações consecutivas não são verificadas melhorias na melhor solução [19].

## 5.4.3 Integração entre o *Neuroevolution* e o Ambiente de Simulação

Definido o modelo de simulação e a estrutura da rede responsável pelas decisões de movimentação relativas ao veículo de transporte, é importante perceber de que maneira estas ferramentas distintas interatuam.

### 5.4.3.1 API

Aliado a todo o tipo de objetos que permitem ao simulador ser capaz de replicar com elevada exatidão o comportamento de um sistema produtivo, a *Application Programing Interface* (API) permite alargar este conjunto de funcionalidades.

Além de permitir a modelação de ambientes de simulação com recurso a programas externos, destaca-se a flexibilidade para a construção de *steps* customizados, recorrendo à linguagem C#, conferindo outro tipo de dinâmica aos processos do simulador.

Para integrar o ambiente de simulação com o algoritmo genético foi definido o seguinte conjunto de etapas:

- Definição de uma estrutura de dados responsável pela transmissão de dados entre o simulador e o Visual Studio, IDE utilizado para conexão por API;



- Definição de um *step* customizado responsável pela leitura dos dados de entrada para a rede neuronal;
- Definição de um novo *step* responsável por tomar a decisão, com base na camada de saída da rede, sobre a estação para a qual o AGV se deve movimentar.

#### 5.4.3.2 Experiments

O simulador tem a funcionalidade de corrida rápida, em que aglomerando um conjunto de variáveis de controlo e de respostas (resultados de simulação) é constituído um cenário. Definindo diferentes variáveis de controlo sobre diferentes cenários é definido o *experiment*. No caso de existir estocasticidade associada ao problema, é possível definir um conjunto de replicações, permitindo avaliar o impacto que o valor tomado pelas variáveis estocásticas pode ter no sistema de produção.

No contexto do problema tratado, os pesos associados a cada uma das redes neuronais estão guardados num ficheiro correspondente. Para o *experiment* customizado desenvolvido, a variável de controlo é o índice do ficheiro responsável por representar o comportamento do AGV.

As variáveis de resposta estão dependentes da função objetivo. No contexto do problema, estas estão diretamente ligadas aos valores de produtividade que são pretendidos maximizar. Todos os parâmetros dos *experiments* podem ser transmitidos por API, evitando a necessidade de abrir o simulador sempre que seja pretendido avaliar o desempenho de uma rede neuronal diferente. Este fator permite otimizar todo o processo de interligação Simulação/*Machine Learning*.

| Scenario                            |            |        | Replications |           | Controls                 | Responses |               |
|-------------------------------------|------------|--------|--------------|-----------|--------------------------|-----------|---------------|
| <input checked="" type="checkbox"/> | Name       | Status | Required     | Completed | IndiceDoFicheiroDaRedeNN | Makespan  | Produtividade |
| <input checked="" type="checkbox"/> | Scenario 1 | Idle   | 1            | 0 of 1    | 1                        |           |               |

↓  
 Redefinir se existir estocacidade

↓  
 Definição do Índice da Rede que modela o comportamento do AGV

↓  
 Definição da resposta segundo o critério a avaliar

Figura 5.8: Exemplo de um *Experiment*

#### 5.4.3.3 Síntese

Integrando os diferentes elementos desenvolvidos, as respostas obtidas segundo métodos evolucionários resultam do seguinte encadeamento de ideias:

- Definição/Ajuste do conjunto de pesos que determinam o código genético da população a atuar no ambiente de simulação;

- Simulação do comportamento de cada um dos 10 indivíduos (10 redes neuronais) e retorno das respectivas respostas por parte do simulador.
- Via API é transmitido o *fitness* de cada uma das redes neuronais. Estes valores servem como *input* para as próximas etapas do algoritmo genético.
- Esta sequência de procedimentos constitui uma iteração do algoritmo. O ciclo é repetido até verificação da condição de paragem, finalizando uma corrida.

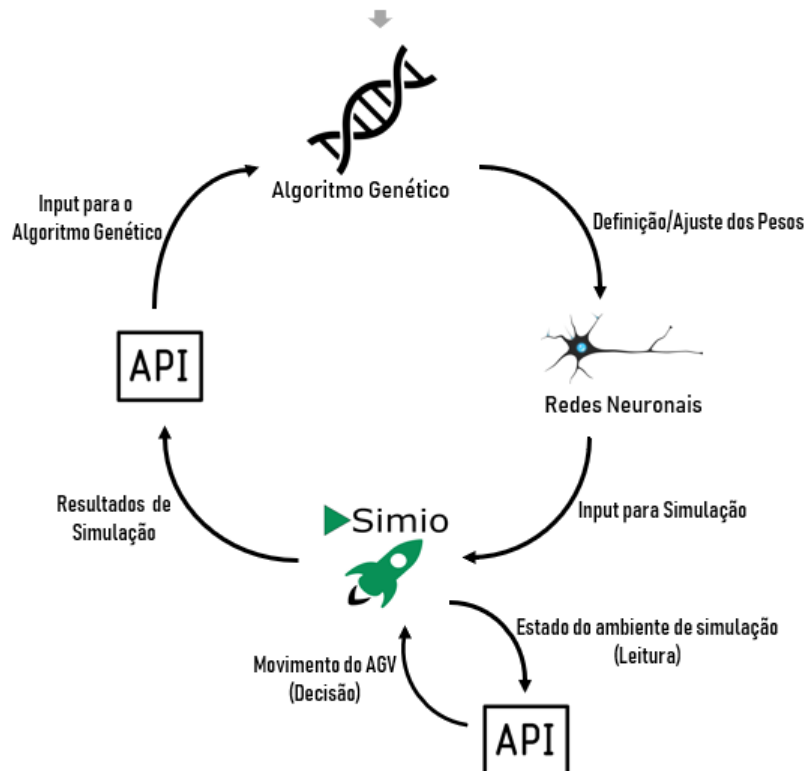


Figura 5.9: Diagrama Síntese

#### 5.4.4 Estratégia Evolutiva 1 - Estrutura da Rede e Resultados

Como previamente mencionado, as camadas de entrada e saída têm um número de neurónios fixos. As diferentes abordagens desenvolvidas atuam, então, sobre a definição da camada oculta.

O recurso a uma única camada é o suficiente para aproximar qualquer função que contenha um mapeamento contínuo de um espaço finito para outro [54].

A camada oculta, nesta abordagem inicial, é composta por 48 neurónios respeitando as "regras de ouro" definidas por Jeff Heaton [54].

- O número de neurónios da camada oculta está compreendido entre o número de neurónios das camadas de entrada e saída;
- O número de neurónios da camada oculta deve ser aproximadamente 2/3 do número de neurónios da camada de entrada, mais os neurónios da camada de saída;

- O número de neurónios é inferior a duas vezes o tamanho da camada de entrada.

A opção por este número de neurónios prende-se pela garantia de um amplo espaço de pesquisa pela solução ótima.

Para avaliar a qualidade da solução desenvolvida foram estipulados os seguintes pontos:

- Definição da população inicial (conjunto de pesos) de forma completamente aleatória;
- Definição da média pesada entre o Número de Movimentos com Peça e o Número de Peças Produzidas como função de *fitness* da rede ( $1 * \text{Número de Movimentos com Peça} + 2000 * \text{Número de Peças Produzidas}$ );

Do algoritmo genético desenvolvido e considerando os parâmetros anteriores, verificaram-se os seguintes resultados:

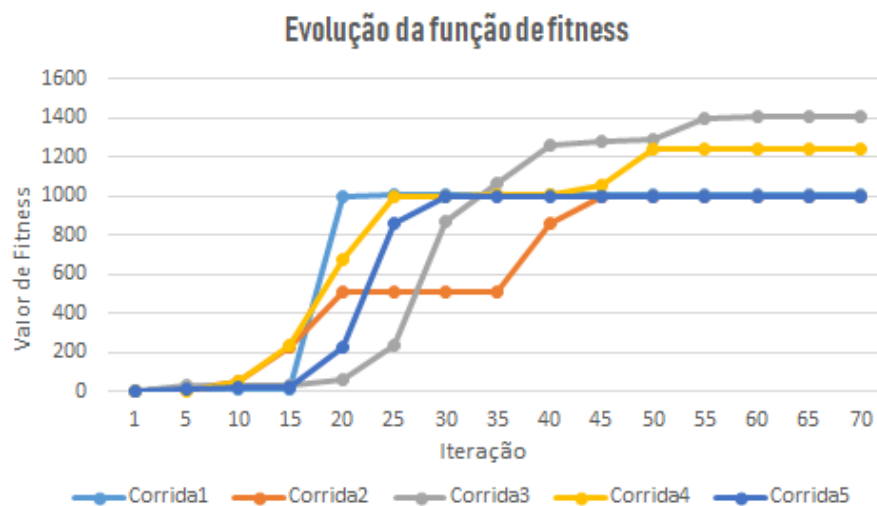


Figura 5.10: Evolução dos valores *fitness*

Na Figura 5.10 é visível uma melhoria nos valores de *fitness* da melhor solução de cada corrida ao longo das iterações. A evolução ao nível dos resultados apresentados permite corroborar o funcionamento do algoritmo. Ainda assim, avaliando o comportamento resultante do melhor indivíduo gerado a partir do algoritmo genético (*Corrida3*), verificou-se uma limitação no comportamento do AGV. Dado o facto de durante as primeiras horas de simulação existirem sempre peças a dar entrada no sistema, a rede define que o AGV se deve movimentar sempre para a *source*. Não sendo capaz de se adaptar ao momento em que todas as peças que deram entrada no sistema já foram transportadas para as respetivas estações de processamento, a rede chega a um máximo local. Neste ponto, as mutações inseridas na descendência gerada não são capazes de evitar este comportamento viciado.

Como solução a este problema, o *fitness* da rede foi redefinido para ser relativo apenas aos valores de produtividade associados. Ou seja, a resposta dos *experiments* ficou restringida ao número de peças produzidas no horizonte temporal definido.

Dado o amplo espaço de pesquisa resultante do elevado número de pesos da rede, não se verificou, ainda assim, qualquer tipo de melhoria nos valores relativos ao indicador de produtividade. Para ultrapassar a situação descrita, a definição da população inicial deixou de ser completamente aleatória. Os pesos das dendrites relativas aos neurónios da camada oculta foram estruturados de forma a que a rede fosse, pelo menos, capaz de replicar o comportamento expresso pela *heurística3* e até subjugá-la.

Por esse motivo, os neurónios foram divididos em 2 tipos de acordo com o comportamento que devem ter face aos dados de entrada:

- Tipo 1 : Tomam o valor 1 se a estação de trabalho  $W$  não contiver peça para ser processada. Isto é feito manipulando o valor de *bias* associado ao neurónio.
- Tipo 2 : Valores após função de ativação superiores a 0 sempre que uma estação  $W$  contém peça. O valor é tanto maior quanto mais longo for o tempo de espera associado à estação para a qual a peça contida na estação  $W$  pretender ser processada. Valor ao qual é somado o tempo relativo à deslocação do AGV para fazer a recolha da peça.

Este tipo de abordagem visa garantir valores de probabilidade mais baixos para estações que não contenham peça no *buffer* de saída. Para o caso de conter peça, o valor de probabilidade de escolha será tão mais baixo quanto maior for a distância a percorrer e o tempo até que a peça seja processada na estação seguinte.

Para uma velocidade de  $0.7m/s$  e um horizonte temporal de 36h, após atuação do algoritmo, foi verificada a produção de 328 unidades. Ainda assim, comparando a melhor solução da estratégia evolucionária (328) com o melhor resultado apresentado até ao momento para este conjunto de parâmetros (334), verifica-se uma diferença entre os valores de produtividade apresentados. O motivo da estratégia evolucionária apresentar valores de produtividade inferiores pode ser explicado pelo facto de um elevado número de neurónios tornar o espaço de pesquisa demasiado vasto para que seja encontrada a solução ótima [55].

É importante observar que com os devidos valores para os seus pesos a rede seria capaz de replicar a *heurística*. Esta observação prova que a rede não atinge melhores desempenhos devido ao elevado espaço de pesquisa e não por incapacidade da própria estrutura.

#### 5.4.5 Estratégia Evolutiva 2 - Estrutura da Rede e Resultados

Considerando as ilações retiradas da *Estratégia Evolutiva 1*, o tamanho da camada oculta foi reduzido diminuindo, consequentemente, o espaço de pesquisa de soluções.

A camada oculta nesta abordagem é composta por 8 neurónios. A opção por este número de neurónios prende-se pela avaliação do desempenho que uma rede mais simples é capaz de demonstrar.

Para garantir que o comportamento do AGV não se remete apenas ao transporte de peças que se encontram na *source*, como foi verificado numa fase inicial da abordagem anterior, 5 dendrites de cada neurónio da camada oculta foram manipuladas:

- O neurónio de índice  $i$  da camada oculta, em vez de partir de um conjunto de pesos completamente aleatórios, passa a apresentar 5 pesos de valor 1 para os neurónios da camada de entrada referentes à presença de peça para ser transportada na estação  $i$ .

A manipulação deste conjunto de pesos garante que a rede parte de uma solução inicial em que é assegurada importância à existência de peça numa estação de trabalho. Este processo está representado na Figura 5.11.

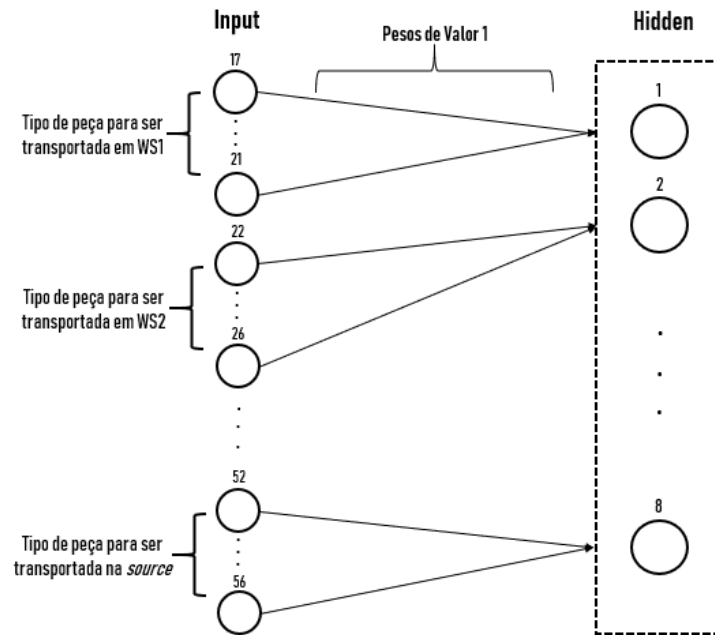


Figura 5.11: Conjunto de pesos não aleatórios

Com o objetivo de maximização dos valores de produtividade e segundo o horizonte temporal e velocidade do AGV definidos para a *Estratégia Evolutiva 1*, foi observada a produção de 339 peças. Considerando estes valores, a abordagem demonstra ser capaz de apresentar melhores resultados que qualquer um dos métodos utilizados anteriormente. Estes serão analisados e explorados com maior detalhe no Capítulo 6 referente à Análise de Resultados.

#### 5.4.6 Aplicação da solução num ambiente estocástico

Após a avaliação da solução num modelo de simulação determinístico, o próximo passo consiste em introduzir estocasticidade no próprio modelo simulado. Nesse sentido, foi associada variabilidade aos tempos de processamento de cada tipo de peça, à semelhança do que se verifica num sistema de produção real. A introdução deste parâmetro permite ajuizar a robustez da solução desenvolvida uma vez que possibilita a avaliação do comportamento do agente de RL num ambiente de simulação estocástico.

Considerando  $T_{ij}$  o tempo de processamento da peça  $i$  na estação  $j$ , este foi redefinido para seguir a distribuição triangular apresentada na Figura 5.12

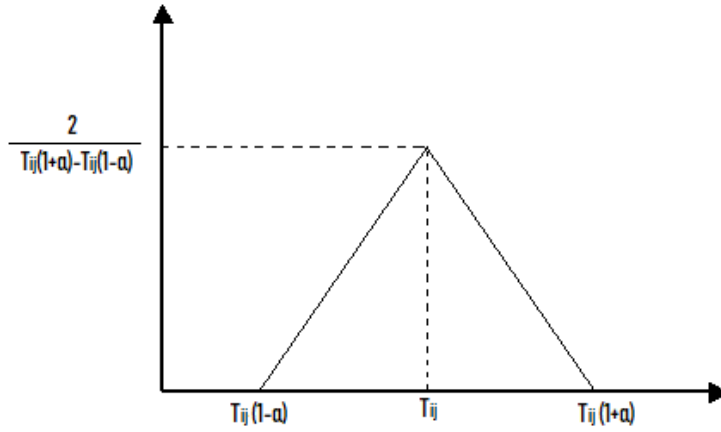


Figura 5.12: Distribuição triangular relativa aos tempos de processamento estocásticos

Matematicamente, a distribuição associada ao tempo de processamento de cada peça pode ser representado pela seguinte expressão:

$$f(x|a,b,c) = \begin{cases} \frac{2(x-(1-\alpha)T_{ij})}{((1+\alpha)T_{ij}-(1-\alpha)T_{ij})(T_{ij}-(1-\alpha)T_{ij})}, & (1-\alpha)T_{ij} \leq x < T_{ij} \\ \frac{2}{(1+\alpha)T_{ij}-(1-\alpha)T_{ij}}, & x = T_{ij} \\ \frac{2((1+\alpha)T_{ij}-x)}{((1+\alpha)T_{ij}-(1-\alpha)T_{ij})(T_{ij}(1+\alpha)-T_{ij})}, & T_{ij} < x \leq (1+\alpha)T_{ij} \\ 0, & \text{para outros casos} \end{cases} \quad (5.3)$$

O coeficiente  $\alpha$  está associado à variabilidade do tempo de processamento. Quanto maior  $\alpha$  maior o desvio padrão relativo à variável tempo. Isto pode ser corroborado pela expressão de Variância associada a este tipo de distribuição:

$$Var(X) = \frac{a^2 + b^2 + c^2 - ab - ac - bc}{18} \quad (5.4)$$

Onde ,  $a$ ,  $b$  e  $c$  tomam os seguintes valores:

$$\begin{cases} a = T_{ij} - T_{ij}\alpha \\ b = T_{ij} + T_{ij}\alpha \\ c = T_{ij} \end{cases} \quad (5.5)$$

Desta forma, a Variância, em função de  $T_{ij}$  e  $\alpha$ , é representada da seguinte forma:

$$\begin{aligned} Var(X) &= (T_{ij} - T_{ij}\alpha)^2 + T_{ij}^2 + (T_{ij} + T_{ij}\alpha)^2 - [(T_{ij} - T_{ij}\alpha) * (T_{ij} + T_{ij}\alpha)] \\ &\quad - [(T_{ij} - T_{ij}\alpha) * T_{ij}] - [(T_{ij} + T_{ij}\alpha) * T_{ij}] \\ \Leftrightarrow Var(X) &= \frac{(T_{ij}\alpha)^2}{6} \end{aligned} \quad (5.6)$$

O desvio padrão pode, então, ser evidenciado pela seguinte expressão:

$$\sigma = \frac{T_{ij}\alpha}{\sqrt{6}} \quad (5.7)$$

Isto permite concluir que quanto maior o  $\alpha$ , maior é o desvio padrão associado ao tempo de processamento. Por consequência, maior é a exigência de robustez relativa à estratégia evolutiva desenvolvida.

Os resultados relativos a este encadeamento de ideias encontram-se apresentados no Capítulo 6.





## Capítulo 6

# Análise de Resultados

Este capítulo apresenta e analisa os resultados computacionais da integração entre estratégias baseadas no *neuroevolution* e a simulação baseada em eventos discretos. Os resultados surgem da execução de diferentes cenários via API que tomam como variáveis a velocidade do AGV e o horizonte temporal de produção de forma a corroborar a flexibilidade associada à abordagem evolucionária.

Com os resultados será possível analisar a aplicabilidade da metodologia e, ao mesmo tempo, avaliar a sua eficácia comparando-a diretamente com as melhores soluções não evolucionárias apresentadas nos Capítulos 4 e 5.

Considerando os tempos de processamento que se pretendem analisar neste capítulo, é de referir que os resultados foram obtidos utilizando o Visual Studio 2017 e a versão 10.18 do SIMIO. Todos os resultados foram processados num computador ASUS K556U com um processador Intel Core i5, dual core até 2,71GHz e 12GB de memória RAM.

### 6.1 Estratégia Evolutiva 1

Em primeira análise, é avaliada a capacidade de modelar o comportamento do AGV por parte de uma rede com 48 neurónios na camada oculta.

Para um horizonte temporal de 36h e segundo diferentes parâmetros de velocidade do AGV, os índices de produtividade apresentados pelo método em relação à melhor solução não evolucionária podem ser observados nas Figuras 6.1, 6.2 e 6.3.

- Velocidade  $0.7m/s$

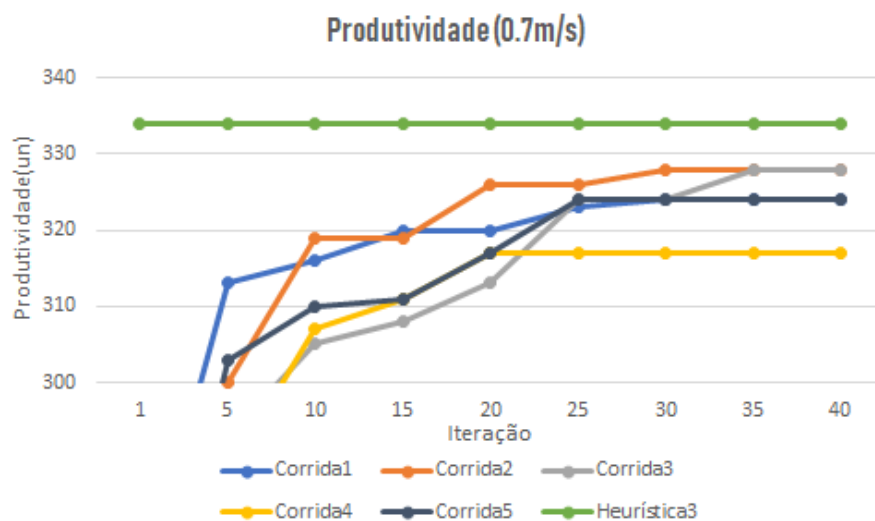


Figura 6.1: Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de  $0.7m/s$

- Velocidade  $1m/s$

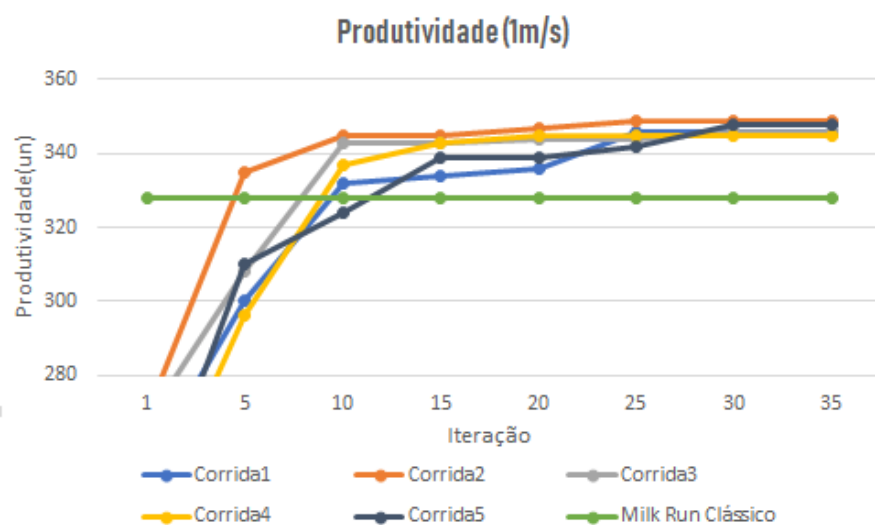


Figura 6.2: Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de  $1m/s$

- Velocidade  $1.5m/s$

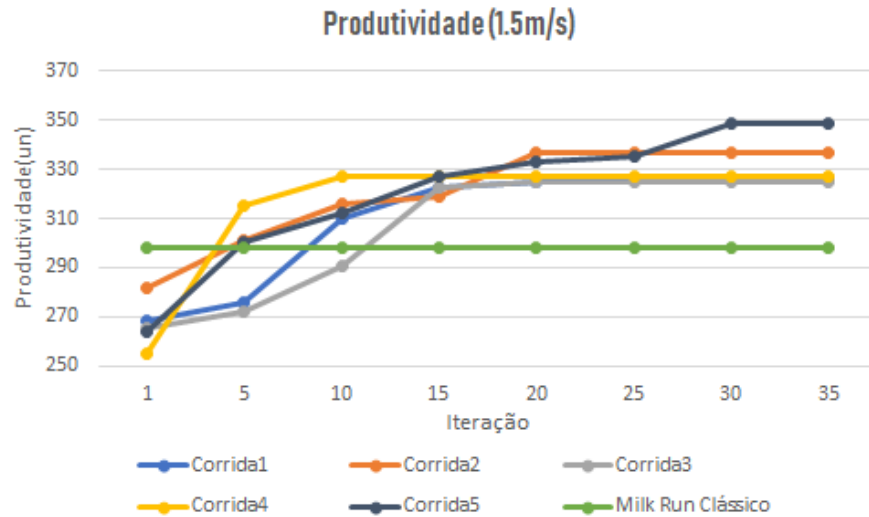


Figura 6.3: Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de  $1.5m/s$

Observando cada uma das respostas apresentadas nas figuras anteriores, é possível perceber que as características do veículo afetam diretamente o desempenho do sistema de transporte quando o seu comportamento está predefinido. Isto permite constatar que a regra de transporte que define a melhor solução de comparação está dependente da própria velocidade do AGV.

Para uma velocidade de  $0.7m/s$  a estratégia evolucionária apresenta índices de produtividade 5% inferiores à solução de comparação. Estes valores estão diretamente ligados ao amplo espaço de pesquisa pela solução ótima conferido pelo número de neurónios da camada oculta. Como resultado desta definição da camada oculta, é verificado o critério de paragem sem a chegada a valores de produtividade superiores.

Para velocidades mais elevadas, contrariamente aos valores apresentados pelas soluções não evolutivas, a estratégia evolucionária consegue aumentar a produtividade do sistema aproveitando características superiores do AGV para satisfazer as necessidades de produção.

Observando as figuras 6.2 e 6.3, a estratégia evolucionária foi capaz de apresentar índices de produtividade 6% e 17% superiores, respetivamente.

Avaliada a responsividade da abordagem evolucionária a alterações nos parâmetros do veículo, foi reformulado o horizonte temporal. Esta alteração tem a finalidade de ajuizar a dinâmica da solução segundo diferentes restrições.

Para um horizonte temporal de 52h e utilizando as características de velocidade da experimentação anterior, os resultados podem ser verificados nas Figuras 6.4, 6.5 e 6.6.

- Velocidade  $0.7m/s$

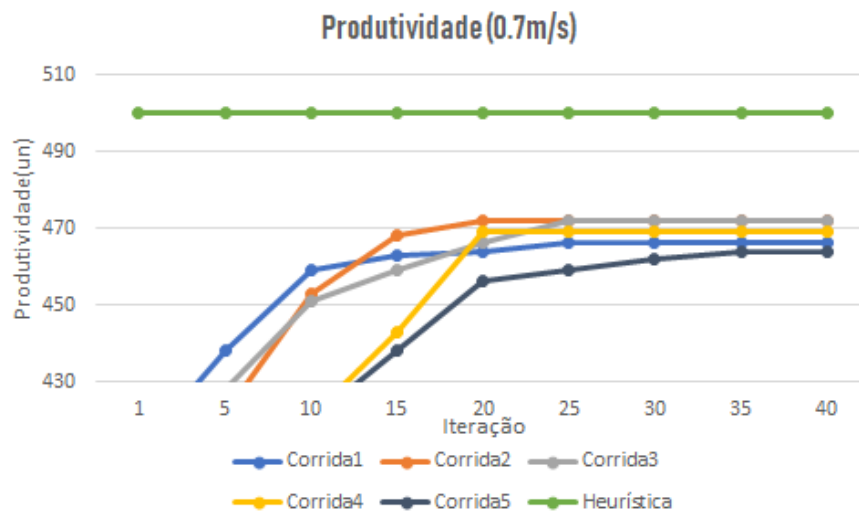


Figura 6.4: Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de  $0.7m/s$

- Velocidade  $1m/s$

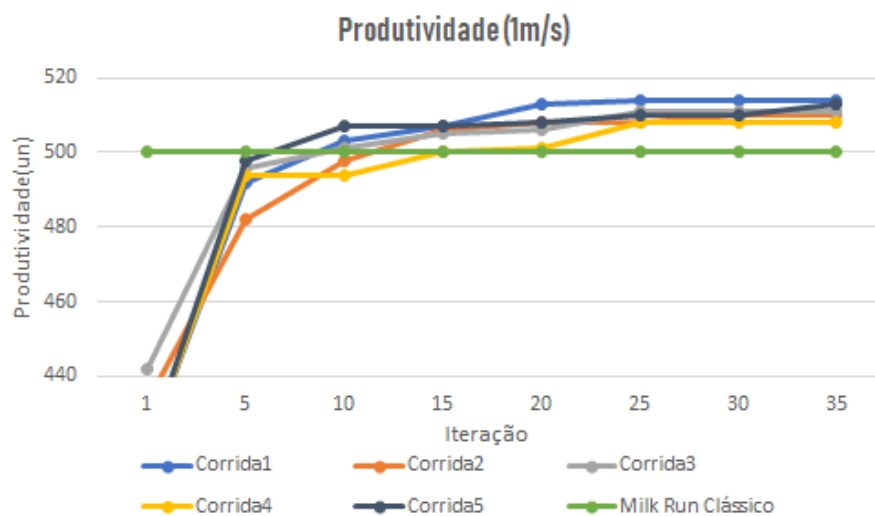


Figura 6.5: Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de  $1m/s$

- Velocidade  $1.5m/s$

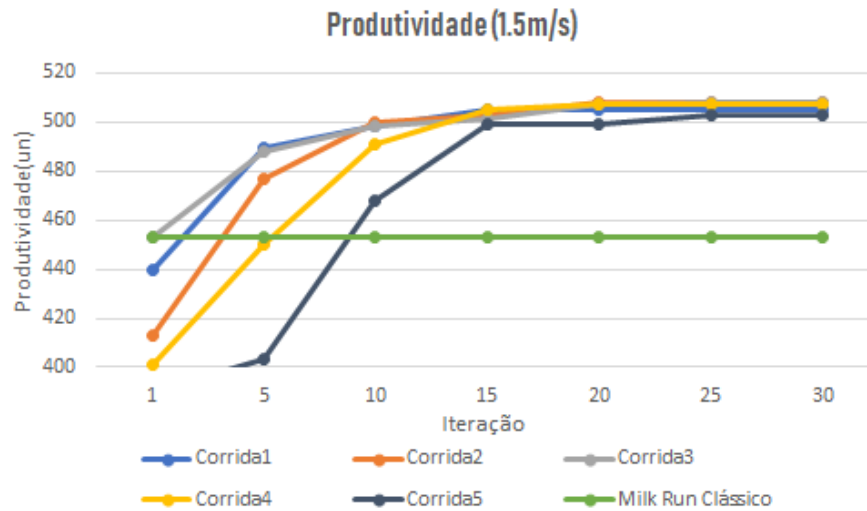


Figura 6.6: Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de  $1.5m/s$

A estratégia evolucionária demonstra ser capaz de se adaptar à nova restrição temporal aplicada. Para uma velocidade de  $1m/s$  apresenta índices de produtividade 3% superiores, por sua vez, para  $1.5m/s$  os valores são incrementados em 12%.

## 6.2 Estratégia Evolutiva 2

Para corroborar a flexibilidade e aplicabilidade do método, surgiu a necessidade de se encurtar o espaço de pesquisa para demonstrar que os menores valores de produtividade obtidos para uma velocidade de  $0.7m/s$  eram dependentes da própria estrutura da rede e não da abordagem proposta. Neste contexto, redefiniu-se o número de neurónios da camada oculta para 8.

Para um horizonte temporal de 36h e segundo as mesmas características de velocidade exploradas anteriormente, foram verificados o conjunto de resultados apresentados nas Figuras 6.7, 6.8 e 6.9.

- Velocidade  $0.7m/s$

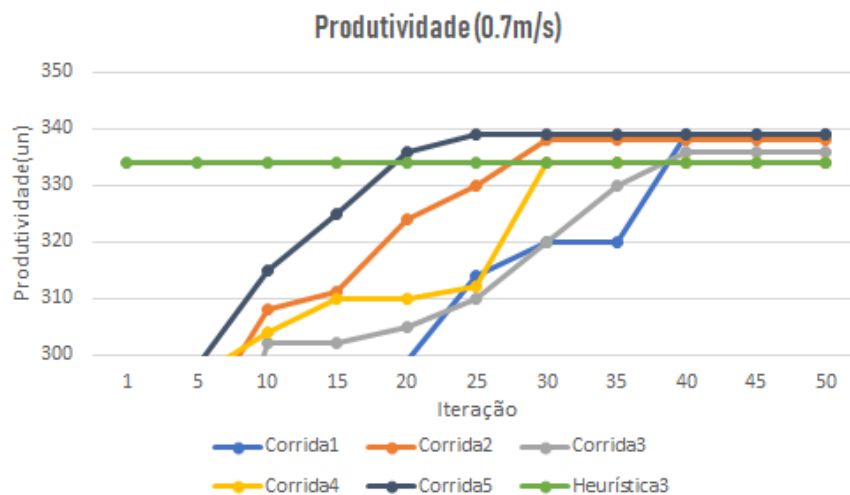


Figura 6.7: Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de  $0.7m/s$

- Velocidade  $1m/s$

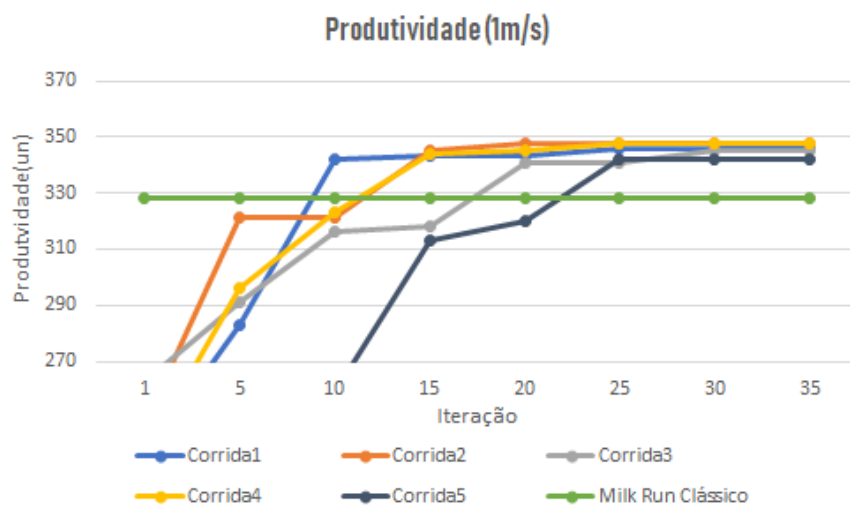


Figura 6.8: Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de  $1m/s$

- Velocidade  $1.5m/s$

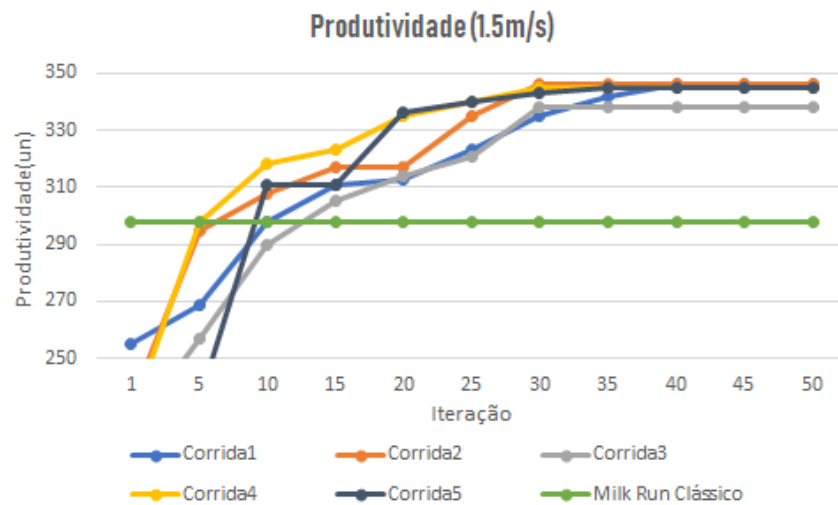


Figura 6.9: Produtividade segundo um horizonte temporal de 36h e velocidade do AGV de  $1.5m/s$

A solução evolucionária demonstrou ser capaz de subjugar todos os resultados anteriores demonstrando a importância da restrição do espaço de pesquisa para o aumento do desempenho apresentado pelo algoritmo genético.

Os índices de produtividade são 1%, 6% e 16% superiores aos apresentados pelas anteriores melhores soluções para as velocidades de  $0.7m/s$ ,  $1m/s$  e  $1.5m/s$  respetivamente.

Redefinindo o horizonte temporal para 52h, os resultados obtidos encontram-se nas Figuras 6.10, 6.11 e 6.12 abaixo representadas:

- Velocidade  $0.7m/s$

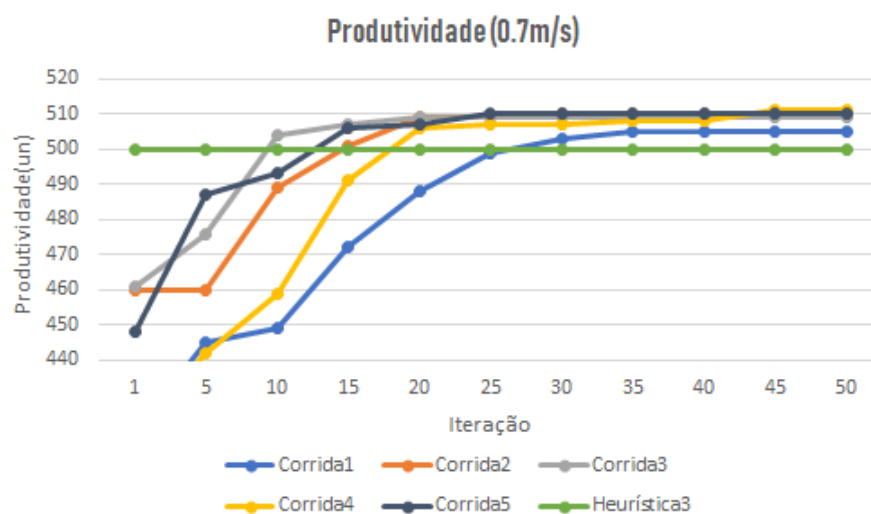


Figura 6.10: Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de  $0.7m/s$

- Velocidade  $1m/s$

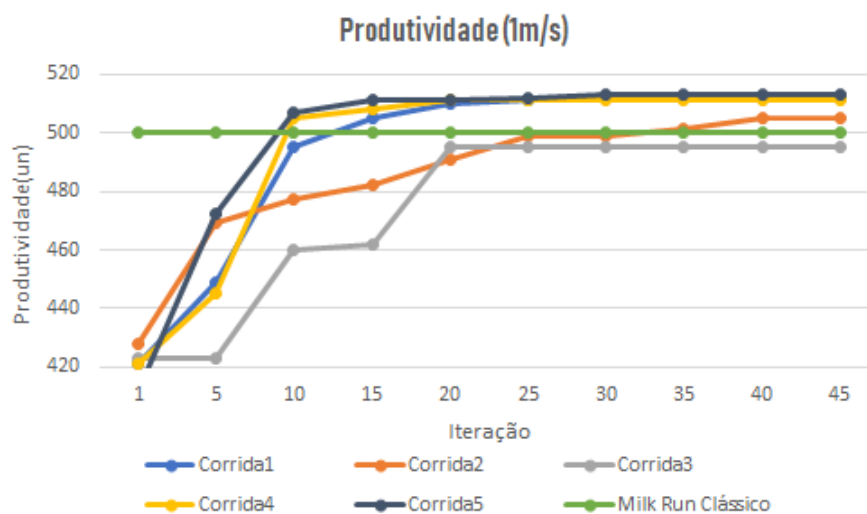


Figura 6.11: Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de  $1m/s$

- Velocidade  $1.5m/s$

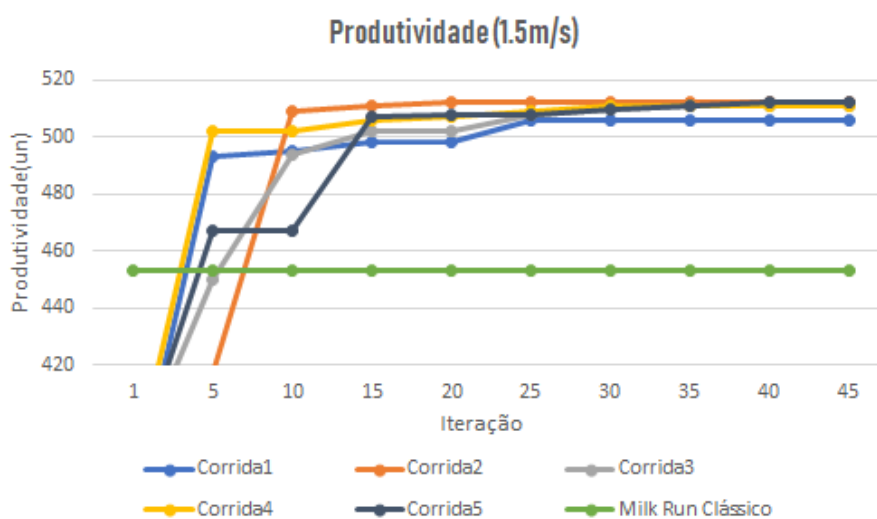


Figura 6.12: Produtividade segundo um horizonte temporal de 52h e velocidade do AGV de  $1.5m/s$

Apresentando resultados 2, 3 e 13% superiores para cada uma das velocidades, a estratégia evolucionária demonstra ser flexível e adaptável a diversas conjunções de variáveis num sistema de produção.

À medida que se verifica um aumento da velocidade do AGV, a discrepância nos valores de produtividade entre a melhor solução não evolucionária e as estratégias evolutivas aumenta. Como a *heurística 3* e o *Milk Run clássico* atuam, preferencialmente, sobre ambientes em que a distância



é um fator limitativo para o sistema de transporte, quando o tempo de viagem entre estações deixa de ser tão restrigente a performance do sistema de transporte é afetada. Deste modo, aumentando a velocidade do AGV os índices de produtividade apresentados acabam por ser inferiores aos apresentados pelas estratégias evolucionárias.

Opondo-se a esta causalidade, a abordagem evolucionária é capaz de se adaptar às variáveis introduzidas e reformular um comportamento adequado para o AGV. Desta forma, revela-se uma solução flexível para a maximização da produtividade segundo diferentes parâmetros do veículo e restrições temporais a que é sujeita.

### 6.3 Estratégia Evolutiva 1 vs Estratégia Evolutiva 2

Demonstrada a aplicabilidade do método num sistema de produção *job shop*, surge a necessidade de comparar diretamente cada uma das abordagens no que se refere à sua eficácia e ao tempo de processamento associado.

No que respeita aos valores de produtividade apresentados por cada uma das estratégias, os resultados médios obtidos foram os seguintes:

Tabela 6.1: Valores médios de produtividade associados a cada uma das estratégias evolutivas

| Velocidade (m/s) | Produtividade Média (un.) |     |     |     |
|------------------|---------------------------|-----|-----|-----|
|                  | 36h                       |     | 52h |     |
|                  | EV1                       | EV2 | EV1 | EV2 |
| <b>0.7</b>       | 324                       | 337 | 468 | 507 |
| <b>1</b>         | 347                       | 346 | 511 | 507 |
| <b>1.5</b>       | 333                       | 344 | 506 | 510 |

Do ponto de vista do esforço computacional, o tempo de processamento associado a cada uma das abordagens, em média, foi o seguinte:

Tabela 6.2: Tempos médios de processamento relativos a cada uma das estratégias evolutivas

| Velocidade (m/s) | Tempo Médio de Processamento (min) |     |     |     |
|------------------|------------------------------------|-----|-----|-----|
|                  | 36h                                |     | 52h |     |
|                  | EV1                                | EV2 | EV1 | EV2 |
| <b>0.7</b>       | 51                                 | 30  | 44  | 31  |
| <b>1</b>         | 48                                 | 28  | 51  | 29  |
| <b>1.5</b>       | 46                                 | 36  | 46  | 34  |

Os resultados apresentados na Tabela 6.1 permitem afirmar que a *Estratégia Evolutiva 2* apresenta maiores índices de produtividade para uma velocidade do AGV de  $0.7m/s$ . No entanto, não é possível inferir uma relação quanto à eficácia das estratégias para os restantes valores de velocidade avaliados. No que concerne ao tempo de processamento associado a cada uma das estratégias, e tendo por base a Tabela 6.2, é possível observar melhores resultados relativos à *Estratégia Evolutiva 2*, independentemente dos parâmetros de simulação. Este conjunto de dados permite afirmar que a definição da camada oculta influencia diretamente o tempo de processamento associado. Contudo, apesar de garantir um maior espaço de soluções, um maior número de neurónios pode tornar o espaço de pesquisa demasiado vasto e o tempo de pesquisa associado demasiado longo.

Conjugando este conjunto de resultados, a *Estratégia Evolutiva 2* apresenta-se como uma melhor alternativa. Esta demonstra ser sempre capaz de delinear um comportamento eficaz para o veículo num tempo de processamento menor que a *Estratégia Evolutiva 1*. O baixo custo de processamento associado aos valores de produtividade apresentados permite justificar a viabilidade do método em sistemas de produção que exijam rápidas redefinições no comportamento do veículo de transporte, de maneira a serem maximizados os seus índices de produtividade.

## 6.4 Avaliação da Estratégia Evolutiva 2 num ambiente estocástico

Demonstrada a aplicabilidade da estratégia num ambiente sem variabilidade associada, é importante avaliar o comportamento da mesma num ambiente estocástico. Aproveitando as vantagens conferidas pela simulação de eventos discretos foi avaliada a robustez da *Estratégia Evolutiva 2*. Para isso, foi analisada a produtividade da 3ª melhor solução de cada cenário num ambiente de produção em que o tempo de processamento é variável. A opção pela 3ª melhor solução prende-se por uma avaliação não tendenciosa daquilo que são os resultados da estratégia.

O tempo de processamento segue, agora, uma distribuição triangular, apresentada com maior detalhe na secção 5.4.6, em que o parâmetro  $\alpha$  associado toma o valor 0.4. Os pontos centrais das distribuições relativas ao tempo de processamento de cada peça estão representados na Tabela 3.2.

O comportamento do AGV num ambiente estocástico foi simulado 10 vezes para horizontes temporais de 36h e 52h e velocidades de  $1m/s$  e  $1.5m/s$ . Foram verificados os seguintes resultados:

- Para um horizonte temporal de 52h:

Tabela 6.3: Avaliação da produtividade em ambiente estocástico para um horizonte temporal de 52h

|   | 1m/s   | 1.5m/s       |
|---|--|--------------|
| <b>Produtividade em Ambiente Determinístico</b> | <b>511</b>                                   | <b>511</b>   |
| <b>Índice da Simulação</b>                      | <b>Produtividade em Ambiente Estocástico</b> |              |
| <b>1</b>  | 511  | 508          |
| <b>2</b>  | 510  | 512          |
| <b>3</b>  | 513  | 509          |
| <b>4</b>  | 506  | 507          |
| <b>5</b>  | 510  | 513          |
| <b>6</b>  | 511  | 510          |
| <b>7</b>  | 509  | 512          |
| <b>8</b>  | 513  | 509          |
| <b>9</b>  | 512  | 513          |
| <b>10</b>                                       | 508  | 509          |
| <b>Média</b>                                    | <b>510.5</b>                                 | <b>509.5</b> |
| <b>Desvio Padrão</b>                            | <b>2.213</b>                                 | <b>2.150</b> |
| <b>Coefficiente de Variação</b>                 | <b>0.43%</b>                                 | <b>0.42%</b> |

- Para um horizonte temporal de 36h:

Tabela 6.4: Avaliação da produtividade em ambiente estocástico para um horizonte temporal de 36h

|   | 1m/s   | 1.5m/s       |
|---|--|--------------|
| <b>Produtividade em Ambiente Determinístico</b> | <b>346</b>                                   | <b>345</b>   |
| <b>Índice da Simulação</b>                      | <b>Produtividade em Ambiente Estocástico</b> |              |
| <b>1</b>  | 343  | 342          |
| <b>2</b>  | 350  | 347          |
| <b>3</b>  | 343  | 344          |
| <b>4</b>  | 345  | 345          |
| <b>5</b>  | 348  | 344          |
| <b>6</b>  | 345  | 345          |
| <b>7</b>  | 345  | 341          |
| <b>8</b>  | 343  | 349          |
| <b>9</b>  | 346  | 346          |
| <b>10</b>                                       | 345  | 343          |
| <b>Média</b>                                    | <b>345</b>                                   | <b>344.5</b> |
| <b>Desvio Padrão</b>                            | <b>2.263</b>                                 | <b>2.366</b> |
| <b>Coefficiente de Variação</b>                 | <b>0.65%</b>                                 | <b>0.68%</b> |

Avaliando a produtividade de cada um dos cenários é possível concluir que, em média, a atuação do agente de RL num ambiente de simulação estocástico apresenta valores de produtividade idênticos aos verificados num ambiente determinístico. Apesar de ter sido introduzida uma variabilidade de 40% associada ao tempo de processamento de cada peça, o baixo desvio padrão demonstra que os valores de produtividade tendem a estar próximos do esperado.

A robustez da solução face à variabilidade introduzida permite validar a aplicabilidade da *Estratégia Evolutiva 2* num sistema de produção real onde se verificam oscilações temporais ao nível dos processos.

Deste modo, os resultados obtidos indiciam que a metodologia criada não é sensível a alterações dos parâmetros do sistema de produção (velocidade do AGV e horizonte temporal) nem à estocasticidade característica deste tipo de sistemas no mundo real.

## Capítulo 7

# Conclusão e Trabalho Futuro

Neste capítulo, são descritas as conclusões relativas ao trabalho realizado no âmbito do projeto de dissertação. Adicionalmente são descritas as contribuições deste trabalho e avaliada a satisfação dos objetivos inicialmente propostos. São, também, referidas algumas sugestões de trabalho futuro assentes numa premissa de melhoria contínua.

### 7.1 Conclusões

O principal objetivo desta dissertação incidia sobre o desenvolvimento de um método híbrido de *Machine Learning* e Simulação para a definição de um sistema de transporte AGV flexível. Este deveria ser capaz de maximizar índices de produtividade adaptando-se a diferentes parâmetros relativos a um sistema de produção. O desempenho do método e validação da abordagem seriam avaliados em comparação com outros sistemas de transporte baseados em heurísticas.

A abordagem seguida partiu do desenvolvimento de um modelo de otimização de forma a redefinir o *layout* do sistema de produção e programou-se o mesmo em CPLEX. Desta ação verificou-se um aumento dos valores de produtividade ainda antes de se atuar sobre o sistema de transporte. De seguida, foram definidas várias soluções de transporte que serviram como base de comparação para a estratégia híbrida desenvolvida. Neste sentido, numa primeira instância, foram avaliados os desempenhos de diferentes heurísticas e do próprio *Milk Run* clássico. Numa fase seguinte, indo ao encontro dos objetivos propostos, foi desenvolvida uma estratégia híbrida que se baseou na utilização de um algoritmo genético para evoluir redes neuronais, estratégia de RL denominada *Neuroevolution*. O recurso a um algoritmo genético permitiu um ajuste dos pesos da rede no sentido de modelar um comportamento otimizado para o AGV com o objetivo de maximizar os índices de produtividade segundo diferentes restrições. Por fim, foram comparadas as diferentes abordagens sendo que a abordagem híbrida foi capaz de subjugar os melhores resultados apresentados pelas diferentes estratégias não evolucionárias. Este resultado era expectável, tendo em conta que a abordagem criada permite, pelo menos, replicar o comportamento apresentado pelas restantes estratégias, mediante a definição do conjunto certo de pesos para a rede neuronal.

Independentemente do horizonte temporal definido e da velocidade relativa ao veículo de transporte, a solução híbrida foi capaz de definir um comportamento para o veículo de transporte capaz de apresentar melhores índices de produtividade em comparação com as restantes soluções. Para uma velocidade de  $0.7m/s$  a produtividade aumentou 1% e 2% relativamente à melhor solução para um horizonte temporal de 36 e 52 horas, respetivamente. Para uma velocidade de  $1.5m/s$  a produtividade aumentou 16% e 13% num horizonte temporal de 36 e 52 horas, respetivamente. Se por um lado o desempenho das soluções não evolucionárias é afetado pela alteração dos parâmetros, a abordagem híbrida demonstra ser capaz de reformular um comportamento adequado para o veículo de transporte.

A solução híbrida desenvolvida foi ainda avaliada num ambiente estocástico, tendo-se demonstrado robusta para lidar com toda a variabilidade associada a um sistema de produção.

A flexibilidade apresentada pelo método demonstra que este é generalizável o suficiente para ser adaptado a outros *job shops*, enaltecendo a dinâmica desta abordagem.

Em jeito de conclusão, podem ser dados como cumpridos os objetivos desta dissertação, sendo que os resultados da abordagem podem viabilizar a aplicação desta metodologia por parte do INESC TEC em projetos futuros que envolvam a formulação de sistemas de transporte em ambientes de produção do tipo *job shop*.

## 7.2 Trabalho Futuro

A aposta em métodos inovadores viabiliza soluções consequentemente mais flexíveis, conferindo inúmeras vantagens quando bem exploradas. Neste contexto de melhoria contínua, na ótica da dissertação, diversos pontos devem ser considerados na procura de melhores soluções.

Dum ponto de vista da solução evolucionária desenvolvida, seria interessante explorar a aplicação de diferentes abordagens relativas ao algoritmo genético. A aplicação de novos operadores de *crossover* e técnicas de mutação seria importante para avaliar o impacto que o processo de *tuning* pode ter nas soluções obtidas e no esforço computacional adjacente. Além do mencionado, seria importante tornar a definição da estrutura dinâmica para aumentar o alcance, em termos industriais, deste tipo de metodologia.

Na ótica da inteligência artificial, seria interessante explorar outras abordagens baseadas em *Reinforcement Learning*, nomeadamente *Q-Learning* e *Policy Gradients*, neste contexto de aplicação.

Com o intuito de verificar a aplicabilidade do método, seria importante testá-lo em diferentes sistemas de produção para avaliar o potencial da metodologia na indústria da manufatura como um todo.

# Referências

- [1] A. Azevedo, “Evolução histórica TQM MRP ERP Make-to-order / Make-to-stock,” tech. rep., FEUP, 2013.
- [2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [3] J. Lehman and R. Miikkulainen, “Neuroevolution,” *Scholarpedia*, vol. 8, no. 6, p. 30977, 2013.
- [4] Thomas Simonini, “An introduction to Policy Gradients with Cartpole and Doom,” 2018.
- [5] J. A. Faria, “Part 1: Fundamentals of Quality Management 1. Quality Assurance and Process Control Course: Quality Management,” tech. rep., FEUP, 2017.
- [6] Deloitte, “The future of manufacturing,” <https://www2.deloitte.com/insights/us/en/industry/manufacturing/>, 2019.
- [7] S. S. Heragu and A. Kusiak, “Efficient models for the facility layout problem,” *European Journal of Operational Research*, vol. 53, pp. 1–13, jul 1991.
- [8] E. Klenk, S. Galka, and W. A. Günthner, “Operating Strategies for In-Plant Milk-Run Systems,” vol. 48, no. 3, pp. 1882–1887, 2015.
- [9] J. Banks, B. Nelson, J. Carson, and D. Nicol, *Discrete-event system simulation*. Dorling Kindersley (India), 4th ed., indian subcontinent adaptation. ed., 2007.
- [10] C. Harrell, *Simulation Using ProModel*. McGraw-Hill Publishing, 2011.
- [11] Stephen P. Waring, *Taylorism Transformed: Scientific Management Theory Since 1945*. 1991.
- [12] Taiichi Ohno, *Toyota Production System: Beyond Large-Scale Production*. 1978.
- [13] A. Hall and R. W. Hall, *Zero Inventories*. 1983.
- [14] P. Dennis, *Lean Production simplified: A plain-language guide to the world’s most powerful production system*. Productivity Press, 2007.
- [15] M. Carravilla, “JIT Just-In-Time,” tech. rep., FEUP, 1997.
- [16] R. Filippini and C. Forza, “The Impact of the Just-in-Time Approach on Production System Performance: A Survey of Italian Industry. A Review and Outlook,” in *A Journey through Manufacturing and Supply Chain Strategy Research*, pp. 19–39, 2016.

- [17] N. Azian, A. Rahman, S. Mohd Sharif, and M. Esa, “Lean Manufacturing Case Study with Kanban System Implementation,” *Procedia Economics and Finance*, vol. 7, pp. 174–180, 2013.
- [18] R. Lopes dos Reis, *Manual da gestão de stocks: teoria e prática*. Lisboa: Editorial Presença, 2008.
- [19] G. Zäpfel, R. Braune, and M. Bögl, *Metaheuristic Search Concepts: A Tutorial with Applications to Production and Logistics*. Springer Berlin Heidelberg, 2010.
- [20] J. M. FRAMINAN, “Efficient heuristic approaches to transform job shops into flow shops,” *IIE Transactions*, vol. 37, pp. 441–451, mar 2005.
- [21] R. Chase, N. Aquilano, and F. Jacobs, *Operations Management for Competitive Advantage*. Irwin/McGraw-Hill series in operations and decision sciences, McGraw-Hill Higher Education, 2000.
- [22] D. Bowersox, D. Closs, and M. Cooper, *Supply chain logistics management*. Irwin/McGraw-Hill series in operations and decision sciences, McGraw-Hill, 2002.
- [23] A. Meyer, “Milk Run Design: Definitions, Concepts and Solution Approaches,” 2015.
- [24] M. Baudin, *Lean Logistics: The Nuts and Bolts of Delivering Materials and Goods*. Taylor & Francis, 2005.
- [25] D. A. de Moura and R. C. Botter, *Independent journal of management & production.*, vol. 7, sep 2016.
- [26] X. Sun, M. Garg, Z. Balaporia, K. Bailey, and T. Gifford, “Optimizing Transportation by Inventory Routing and Workload Balancing: Optimizing Daily Dray Operations Across an Intermodal Freight Network,” *Interfaces*, vol. 44, pp. 579–590, dec 2014.
- [27] R. Domingo, R. Alvarez, M. Melodía Peña, and R. Calvo, “Materials flow improvement in a lean assembly line: a case study,” *Assembly Automation*, vol. 27, pp. 141–147, apr 2007.
- [28] I. of Industrial Engineers (1981-), *Industrial engineering terminology: a revision of ANSI Z94.0-1982 : an American national standard, approved July 10, 1989*. ANSI standard, Industrial Engineering and Management Press, 1990.
- [29] S. Akincilar and C. Rad, “Material Handling System Design: A Case-Study in Bosch Rexroth Japan, Master Thesis, Production Development and Management,” tech. rep., 2013.
- [30] H. Ichikawa, “Simulating an applied model to optimize cell production and parts supply (mizusumashi) for laptop assembly,” in *Winter simulation conference*, pp. 2272–2280, Winter Simulation Conference, 2009.
- [31] C. Cho and P. J. Egbelu, “Design of a web-based integrated material handling system for manufacturing applications,” *International Journal of Production Research*, vol. 43, no. 2, pp. 375–403, 2005.
- [32] H.-K. CHU, P. J. EGBELU, and C.-T. WU, “ADVISOR: A computer-aided material handling equipment selection system,” *International Journal of Production Research*, vol. 33, pp. 3311–3329, dec 1995.



- [33] J. &. Nomura and S. Takakuwa, “OPTIMIZATION OF A NUMBER OF CONTAINERS FOR ASSEMBLY LINES: THE FIXED-COURSE PICK-UP SYSTEM,” *Int j simul model*, vol. 5, pp. 155–166, 2006.
- [34] Dematic, “Egemin is Now Dematic | DEMATIC Global Website,” 2018.
- [35] H. S. Kilic, M. B. Durmusoglu, and M. Baskak, “Classification and modeling for in-plant milk-run distribution systems,” *The International Journal of Advanced Manufacturing Technology*, vol. 62, pp. 1135–1146, oct 2012.
- [36] H.-S. Hwang, “Heuristic transporter routing model for manufacturing facility design,” *Computers & Industrial Engineering*, vol. 46, pp. 243–251, apr 2004.
- [37] S. I. Satoglu and I. E. Sahin, “Design of a just-in-time periodic material supply system for the assembly lines and an application in electronics industry,” *The International Journal of Advanced Manufacturing Technology*, vol. 65, pp. 319–332, mar 2013.
- [38] U. Teschemacher and G. Reinhart, “Ant Colony Optimization Algorithms to Enable Dynamic Milkrun Logistics,” *Procedia CIRP*, vol. 63, pp. 762–767, jan 2017.
- [39] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2017.
- [40] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement Learning: A Survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, may 1996.
- [41] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, “Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning,” tech. rep., 2018.
- [42] D. Floreano, P. Dürri, and C. Mattiussi, “Neuroevolution: from architectures to learning,” *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, 2008.
- [43] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.
- [44] C. Igel, “Neuroevolution for reinforcement learning using evolution strategies,” in *The 2003 Congress on Evolutionary Computation, 2003. CEC ’03.*, vol. 4, pp. 2588–2595, IEEE.
- [45] R. Koppejan and S. Whiteson, “Neuroevolutionary reinforcement learning for generalized helicopter control,” in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO ’09*, (New York, New York, USA), p. 145, ACM Press, 2009.
- [46] M. Ghavamzadeh and Y. Engel, “Bayesian Policy Gradient,” tech. rep., 2006.
- [47] R. J. Williams, “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning,” tech. rep., 1992.
- [48] N. Kohl and P. Stone, “Policy gradient reinforcement learning for fast quadrupedal locomotion,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, pp. 2619–2624 Vol.3, IEEE, 2004.
- [49] OpenAI, “Openai five.” <https://blog.openai.com/openai-five/>, 2018.

- [50] E. Babulak and M. Wang, “Discrete Event Simulation: State of the Art,” in *Discrete Event Simulations*, Sciyo, aug 2010.
- [51] P. Korytkowski and R. Karkoszka, “Simulation-based efficiency analysis of an in-plant milk-run operator under disturbances,” *The International Journal of Advanced Manufacturing Technology*, vol. 82, no. 5-8, pp. 827–837, 2016.
- [52] J. S. Smith, R. A. Wysk, D. T. Sturrock, S. E. Ramaswamy, G. D. Smith, and S. B. Joshi, “Discrete event simulation for shop floor control,” in *Proceedings of Winter Simulation Conference*, pp. 962–969, Dec 1994.
- [53] S. Overflow, “How to evolve weights of a neural network in Neuroevolution?,” 2015.
- [54] J. Heaton, *Introduction to neural networks with Java*. Heaton Research, Inc., 2008.
- [55] V. Hoekstra, “An overview of neuroevolution techniques,” tech. rep., 2011.