

**Universidade do Minho**  
Escola de Engenharia

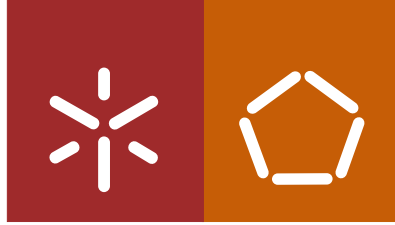
Pedro Miguel Martins de Oliveira

***Benchmarking* sobre técnicas de  
otimização para modelos de apoio  
à decisão na medicina intensiva**

Pedro Miguel Martins de Oliveira ***Benchmarking* sobre técnicas de otimização para modelos de apoio à decisão na medicina intensiva**

UMinho | 2015

outubro de 2015



**Universidade do Minho**  
Escola de Engenharia

Pedro Miguel Martins de Oliveira

***Benchmarking* sobre técnicas de  
otimização para modelos de apoio  
à decisão na medicina intensiva**

Dissertação de Mestrado  
Mestrado integrado em Engenharia e Gestão de  
Sistemas de Informação

Trabalho efetuado sob a orientação do  
**Prof. Doutor Manuel Filipe Santos**  
**Prof. Doutor Carlos Filipe Portela**

## DECLARAÇÃO

Nome Pedro Miguel Martins de Oliveira

Endereço electrónico: hazetheweb@hotmail.com

Telefone: 910970503

Número do Bilhete de Identidade: 13626184

Título dissertação: Benchmarking sobre técnicas de otimização para modelos de apoio à decisão na medicina intensiva

Orientador(es): Manuel Filipe Santos e Carlos Filipe Portela

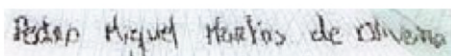
Ano de conclusão: 2015

Designação do Mestrado: Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA DISSERTAÇÃO, APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, \_\_/\_\_/2015

Assinatura:





## Resumo

Os modelos de apoio à decisão na medicina intensiva são desenvolvidos para apoiar as equipas médicas na tomada de decisão sobre os tratamentos a aplicar a um doente. Existem inúmeros sistemas de apoio à decisão (SAD) que foram desenvolvidos nas últimas décadas para os mais variados ambientes. Em muitos desses SADs, o *Machine Learning* é utilizado para dar resposta a um problema específico. No entanto, a otimização desses sistemas é particularmente difícil de aplicar devido à dinâmica, complexidade e naturezas multidisciplinares. Com isso, hoje em dia existe uma constante investigação e desenvolvimento de novos algoritmos capazes de extrair conhecimento tratado de grandes volumes de dados, obtendo assim melhores resultados preditivos do que os atuais algoritmos. Existe e emerge um vasto grupo de técnicas e modelos que melhor se adaptam à natureza e complexidade do problema. É nesse propósito que se insere este trabalho. Esta dissertação teve como principal objetivo identificar essas técnicas de otimização, avaliar, comparar e classificar aquelas que melhor podem responder às particularidades da Medicina Intensiva. Como exemplo foram analisados modelos Evolutionary Crisp Rule Learning, Lazy Learning, Evolutionary Fuzzy Rule Learning, Prototype Generation, Fuzzy Instance Based Learning, Decision Trees, Crisp Rule Learning, Neural Networks e Evolutionary Prototype Selection. De seguida foram efetuados alguns desenvolvimentos / testes de modo a aplicar a melhor técnica a um problema de cuidados intensivos, onde a técnicas Decision Trees Genetic Algorithm, Supervised Classifier System e KNNAdaptive obtiveram a melhor taxa de acuidade, mostrando assim a sua exequibilidade e capacidade de atuar em um ambiente real.

Palavras-Chave: *Benchmarking*; Técnicas de Otimização; Sistemas de apoio à decisão; *Machine Learning*; Medicina Intensiva; Heurísticas

## **Abstract**

The decision support models in intensive care are developed to support medical staff in decision making about treatments to be applied to a patient. There are numerous systems for decision support (DSS) that have been developed in recent decades for a variety of environments. In many of these DSS, the Machine Learning is used to address a specific problem. However, the optimization of these systems is particularly difficult to apply due to the dynamic, complex and multidisciplinary nature. Thus, there is a constant research and development of new algorithms capable of extracting knowledge treated large volumes of data today, able to obtain better predictive results than current algorithms. In fact, emerges a large group of techniques and models that are best suited to the nature and complexity of the problem. This work is incorporated in this context. This dissertation aims to identify these optimization techniques, evaluate, compare and classify them in order to identify what are the best respond to the particularities of Critical Care Medicine. As an example several models were analyzed: Evolutionary Fuzzy Rule Learning, Lazy Learning, Evolutionary Crisp Rule Learning, Prototype Generation, Fuzzy Instance Based Learning, Decision Trees, Crisp Rule Learning, Neural Networks and Evolutionary Prototype Selection. Afterwards some developments / tests were made in order to apply the best technique to a problem of intensive care, where the Decision Trees Genetic Algorithm, Supervised Classifier System and KNNAdaptive obtained the most accurate rate, thus showing their feasibility and ability to work in a real environment.

**Keywords:** Benchmarking; Optimization Techniques; Decision support systems; Machine Learning; Intensive Care Medicine; Heuristics

## Índice

Anexo 3 .....	ii
Resumo .....	iv
Abstract.....	v
Índice de Figuras.....	vii
Índice de Tabelas .....	ix
Índice de Algoritmos .....	xi
Siglas e Acrónimos .....	xii
1. Introdução.....	1
1.1. Enquadramento.....	1
1.2. Objetivos propostos.....	3
1.3. Estrutura do documento.....	4
2. Revisão da Literatura.....	6
2.1. Medicina intensiva .....	7
2.2. Sistemas de apoio à decisão .....	8
2.3. Técnicas e modelos de otimização .....	8
3. Abordagem metodológica .....	11
3.1. Investigação Quantitativa.....	11
3.2. Processo de <i>benchmarking</i> .....	12
4. Planeamento do exercício.....	14
4.1. Etapas do <i>Benchmarking</i> .....	14
4.2. INTCare.....	14
4.2.1. Sistema inteligente de apoio à decisão .....	15
4.2.2. Aquisição de dados.....	17
5. Constituição da avaliação .....	19
5.1. Sistema de aprendizagem automática.....	19
5.1.1. Regressão Lógica.....	19

---

5.1.2. Support Vector Machines .....	20
5.1.3. Redes neuronais artificiais.....	21
5.2. Representação do conhecimento .....	23
5.2.1. Sistema baseado em Ontologias .....	23
5.2.2. Guideline-based .....	24
5.2.3. Fuzzy logic .....	25
5.2.4. Visualização da Informação .....	26
5.3. <i>Text e Data mining</i> .....	26
5.3.1. Informational retrieval.....	27
5.3.2. Processamento de Linguagem Natural .....	28
5.4. <i>Multi-purpose</i> .....	29
5.4.1. Árvores de decisão.....	29
5.4.2. Lógica Bayesian .....	30
6. Recolha de dados.....	31
6.1. Meta-heurísticas .....	31
6.1.1. Single-solution based.....	33
6.1.1.1. Simulated annealing .....	33
A. <i>Microcanonic annealing</i> .....	34
B. Método <i>threshold accepting</i> .....	35
C. Método <i>noising</i> .....	35
6.1.1.2. Pesquisa Tabu .....	36
6.1.1.3. Método GRASP .....	37
6.1.1.4. Variable neighborhood search .....	38
6.1.1.5. Guided local search .....	40
6.1.1.6. Iterated local search .....	40
6.1.2. Population-based .....	41
6.1.2.1. Computação evolucionária .....	41



---

A. Algoritmos genéticos .....	43
B. Estratégia evolutiva .....	44
C. <i>Evolutionary programming</i> .....	44
D. Programação genética.....	45
6.1.2.2. Outros algoritmos evolutivos .....	46
A. <i>Learning Classifier Systems</i> .....	46
B. Algoritmos de estimação de distribuição .....	48
C. Algoritmos de Coevolução .....	49
D. <i>Cultural algorithms</i> .....	50
E. <i>Scatter search and path relinking</i> .....	51
F. <i>Differential evolution</i> .....	51
6.1.2.3. Swarm intelligence .....	52
A. <i>Ant colony optimization</i> .....	53
B. Otimização por enxame de partículas.....	55
C. <i>Bacterial foraging optimization algorithm</i> .....	57
D. <i>Bee colony optimization-based algorithm</i> .....	58
E. <i>Artificial immune systems</i> .....	59
F. <i>Biogeography-based optimization</i> .....	59
G. <i>Bat algorithm</i> .....	60
H. <i>Firefly algorithm</i> .....	62
I. <i>Krill Herd</i> .....	63
J. <i>Cuckoo Search</i> .....	64
K. Comparação de <i>swarm intelligence</i> .....	66
7. Análise dos dados .....	68
7.1. Ferramentas utilizadas .....	68
7.1.1. Visão técnica KEEL .....	69
7.2. Exploração dos dados.....	69

---

7.3. Tratamento dos dados.....	71
7.4. Aplicação.....	73
7.5. Proposta de medidas.....	77
8. Conclusão.....	78
8.1. Conhecimento adquirido e trabalho futuro.....	78
9. Referências Bibliográficas.....	80
Apêndice 1 Experiências efetuadas.....	94
Apêndice 2 Publicação Científica.....	97
Apêndice 3 Publicação Científica.....	98

## Índice de Figuras

Figura 1 – Soluções locais e global ótimas .....	9
Figura 2 – Arquitetura do SIAD .....	16
Figura 3 – Arquitetura da aquisição de dados .....	17
Figura 4 – Classificação linear (ex. RNA) vs. não-linear (ex. SVM).....	21
Figura 5 – Exemplo de uma RNA .....	22
Figura 6 – Exemplos simples dos tipos de rede de transições.....	24
Figura 7 – Representação do output no conjunto de dados fornecidos .....	71
Figura 8 – Interface principal do KEEL .....	72
Figura 9 – Cenário de uma experiência na ferramenta KEEL.....	73



## Índice de Tabelas

Tabela 1 – Fontes de dados do SIAD .....	18
Tabela 2 – Abordagem histórica das meta-heurísticas .....	32
Tabela 3 – Análise de técnicas Swarm Intelligence .....	67
Tabela 4 – Descrição das variáveis.....	70
Tabela 5 – Parâmetros utilizados no estudo experimental das melhores técnicas.....	74
Tabela 6 – Testes de classificação das melhores técnicas .....	75
Tabela 7 – Matriz nível de confiança das melhores técnicas .....	75
Tabela 8 – Matrizes de confusão das melhores técnicas .....	77
Tabela 9 – Demonstração dos algoritmos testados.....	94



## Índice de Algoritmos

Algoritmo 1 – Simulated annealing.....	34
Algoritmo 2 – Pesquisa tabu.....	36
Algoritmo 3 – Greedy Randomized Adaptive Search Procedure.....	37
Algoritmo 4 – Variable Neighborhood Search.....	39
Algoritmo 5 – Computação evolucionária .....	42
Algoritmo 6 – Ant Colony Optimization.....	55
Algoritmo 7 – Otimização por enxame de partículas .....	56
Algoritmo 8 – Artificial Bee Colony .....	58
Algoritmo 9 – Biogeography-based optimization .....	60
Algoritmo 10 – Bat algorithm .....	61
Algoritmo 11 – Firefly Algorithm .....	62
Algoritmo 12 – Krill Herd .....	64
Algoritmo 13 – Cuckoo Search via Lévy Flights .....	66

## **Siglas e Acrónimos**

ABC – *Artificial bee colony*

AC – Algoritmos de coevolução

ACO – *Ant colony optimization*

AD – Árvores de Decisão

AE – Algoritmos evolucionários

AED – Algoritmos de estimação de distribuição

AG – Algoritmos genéticos

AIS – *Artificial immune systems*

BA – *Bat algorithm*

BAM – *Best action map*

BBO – *Biogeography-based optimization*

BCO – *Bee colony optimization*

BCOB – *Bee colony optimization-based*

BFOA – *Bacterial foraging optimization algorithm*

CA – *Cultural algorithm*

CADM – Conjunto de Algoritmos Data Mining

CAM – *Complete action map*

CDM – *Centralized Data Mining*

CE – Computação evolucionária

CHP – Centro Hospitalar do Porto

CMA-ES – *Covariance Matrix Adaptation Evolution Strategy*

CS – *Cuckoo Search*

DE – *Differential evolution*

DDM – *Distributed Data Mining*

DM – *Data Mining*

EC – Eventos Críticos

EE – Estratégia evolutiva

EP – *Evolutionary programming*

FA – *Firefly Algorithm*

FL – *Fuzzy logic*

GLS – *Guided local search*

GRASP – *Greedy Randomized Adaptive Search Procedure*



ILS – *Iterated local search*  
IR – *Information retrieval*  
KH – *Krill Herd*  
LB – Laboratório  
LISP – *List Programming*  
LCS – *Learning classifier systems*  
MA – *Microcanonic annealing*  
MC – Monitor de Cabeceira  
ML – *Machine Learning*  
MN – Método *noising*  
OEP – Otimização por enxame de partículas  
PG – Programação genética  
PLN – Processamento de Linguagem Natural  
PTB – Pesquisa *tabu*  
RC – Representação do Conhecimento  
REE – Registo Eletrónico de Enfermagem  
RES – Registo Eletrónico de Saúde  
RF – *Random Forests*  
RL – Regressão Lógica  
RNA – Redes Neurais Artificiais  
SA – *Simulated annealing*  
SAA – Sistemas de Aprendizagem Automática  
SAD – Sistemas de Apoio à Decisão  
SC – Sistema de Classificação  
SI – *Swarm Intelligence*  
SIAD – Sistema Inteligente de Apoio à Decisão  
SSPR – *Scatter search and path relinking*  
ST – Sistema Terapêutico  
SV – Sinais Vitais  
SVM – *Support Vector Machines*  
TA – *Threshold accepting*  
TIC – Tecnologias de informação e comunicação  
TM – *Text mining*  
UCI – Unidade de Cuidados Intensivas

VBA – *Virtual bee algorithm*

VI – Visualização da Informação

VNS – *Variable Neighborhood Search*

## 1. Introdução

Um sistema de apoio à decisão (SAD) é um sistema orientado para o utilizador, porque ajuda-o a entender e a formalizar os seus objetivos e preferências, a entender a estrutura dos problemas, na procura de soluções e na realização de análises de sensibilidade orientadas para o problema [1]. O seu foco é apoiar e melhorar o processo de tomada de decisão baseado no desenvolvimento e implementação de tecnologias suportadas por dados e informação [2]. As decisões podem ser vistas como uma pesquisa onde o decisor pretende entender que condições têm de ser tomadas em consideração na obtenção de um determinado resultado, mas também pode ser descrita como uma tarefa onde os parâmetros devem ser obtidos através de resultados passados.

Como base nestas decisões estão os sistemas de aprendizagem e conhecimento, que se enriquecem com informação provenientes dos SAD em forma de dados estruturados ou não estruturados para melhor procurar, corresponder e obter previsões e análises para o problema em questão. Mais recentemente o desenvolvimento de SAD implica a utilização de conceitos de aprendizagem automática.

### 1.1. Enquadramento

Os Sistemas de *Machine learning* (ML), em português conhecidos como sistema de aprendizagem automática (SAA), podem ajudar a descobrir padrões, realizar determinadas tarefas através da generalização de casos e na utilização de dados. Este tema levanta questões de ordem filosófica sobre o que constitui “aprendizagem” no geral, tipicamente definido por J. Fulcher [3]:

- Ganhar conhecimento ou habilidades, ao estudar ou experimentar
- Comprometer-se com a memória
- Ser avisado, ser informado
- Tornar-se consciente
- Receber instruções, ser informado
- A modificação de comportamento por meio de interação com o ambiente

Podemos definir a informação como dados (factos) com significado, da mesma forma o conhecimento pode ser considerado como compreensão da informação [3]. A sua

implementação é considerada viável e de baixo custo em relação à programação manual. Com isso, à medida que surgem novos dados, os problemas mais ambiciosos podem ser solucionados recorrendo aos SAA. Como resultado os SAA são amplamente utilizados em ciências da computação entre outros, como por exemplo, pesquisas na web, filtros de *spam*, sistemas de recomendações, colocação de anúncios, atribuição de rankings, detecção de fraudes entre muitas outras [4].

Através de uma abordagem algorítmica [5], os processos de aprendizagem, adaptação e otimização incorporam técnicas dedutivas e indutivas que se motivam em pressupostos para obterem evidências ou certezas. Nesta parte existe uma tentativa de extrair regras/padrões a partir dos dados disponíveis (geralmente recorrendo a técnicas estatísticas ou *Data mining*) [6], onde os resultados são probabilidades em vez de certezas [3].

Com isto, é possível observar que a otimização faz parte de um SAD. A maioria dos problemas de aprendizagem automática reduzem-se a problemas de otimização. Considerando a ação de análise do SAD e resolvendo problemas de um conjunto específico de dados, o decisor formula o problema selecionando famílias de modelos apropriados, transformando os dados para um formato adequado [7]. Este tipo de modelo é tipicamente treinado para resolver problemas nucleares de otimização, que melhoram as variáveis ou parâmetros do modelo a respeito da função utilizada. A área de investigação da matemática computacional interceta com o SAD ao nível dos problemas de otimização nucleares, predispondo teorias e definições que constituem uma solução ótima tendo como base as suas condições ideais. Para uma melhor avaliação do desempenho dos modelos de otimização em um SAD, é necessário desenvolver um instrumento capaz de avaliar as melhores práticas / algoritmos de otimização. O resultado da aplicação desse instrumento deverá ser capaz de trazer vantagens competitivas ao SAD.

Habitualmente, quando algo não está a correr de acordo com o expectável é iniciado um processo de *benchmarking*. O *benchmarking* quando aplicado de forma correta facilita a identificação de falhas e conseqüentemente a identificação de melhorias de desempenho crítico num sistema. O *benchmarking* pode ser classificado em três tipos: interno, competitivo e genérico [8]. Dentro desses tipos existem princípios comuns como a

reciprocidade da informação obtida, analogia na obtenção dos resultados e a medição dos resultados através de indicadores. Por fim deve-se ter em conta que o *benchmarking* é uma comparação de desempenhos e a validação das conclusões é feita através de métodos estatísticos [9].

Para uma aplicação correta de um processo de *benchmarking* é necessário realizar um planeamento, uma recolha de dados, uma análise dos dados e uma adaptação e melhoria da(s) técnica(s) de otimização aplicados aos modelos de apoio à decisão. O planeamento deve definir a missão que se pretende levar a cabo, qual o objeto de *benchmarking* e a recolha de dados das mais variadas fontes. A análise dos dados deve determinar as diferenças de desempenho e a origem dos bons resultados, como a adaptação e melhoria da técnica de otimização identificada, determinando a ação para melhorar o desempenho ou processo da empresa de acordo com as conclusões do estudo [10,11].

Este trabalho, bem como as conclusões do mesmo faz parte de um estudo avaliativo integrado no projeto INTCare. O INTCare é um Sistemas Inteligente de Apoio à Decisão (SIAD) que está em constante desenvolvimento e encontra-se implementado na Unidade de Cuidados Intensivos (UCI) do Centro Hospitalar do Porto (CHP). Este SIAD é baseado em agentes inteligentes e modelos de *data mining* [12] que procuram automatizar tarefas e prever eventos clínicos de modo a ajudar o diagnóstico e tratamento de doentes possibilitando a recuperação para o seu estado anterior de saúde. Estes doentes normalmente são admitidos nas UCI para que seja possível monitorizar e manter as suas funções fisiológicas através de dispositivos de suporte de vida. Estas unidades são vitais para o doente e estes devem ser continuamente controlados bem como o estado de cada sistema orgânico: neurológico, respiratório, hepático, hematológico, cardiovascular e renal [13]. Com vista a assegurar as condições e a vida do doente, estes sistemas orgânicos são monitorizados pelos dispositivos médicos e a sua informação pode ser interpretadas por SAD que ajudam as equipas médicas a aplicar o melhor procedimento. Atualmente a capacidade de otimização dos modelos de previsão presentes no INTCare é diminuta.

## **1.2. Objetivos propostos**

O objetivo principal deste projeto passa por fazer um levantamento exaustivo das principais técnicas de otimização e encontrar uma ou mais heurísticas que sejam capazes de otimizar

modelos de previsão / decisão para Medicina Intensiva. Para a concretização do objetivo principal foram definidos os seguintes objetivos específicos:

- Consciencializar o que é um Sistema de apoio à decisão e de que forma a aprendizagem automática e as técnicas de otimização podem contribuir para a sua construção.
- Pesquisar, recolher e analisar técnicas de otimização;
- Elaborar um processo *benchmarking* às técnicas identificadas através de ferramentas de alto-nível;
- Criar um guia com a descrição das principais técnicas de otimização utilizadas a nível científico;
- Analisar, Avaliar e Classificar as melhores técnicas para o objeto em estudo;
- Demonstrar a exequibilidade das técnicas selecionados num contexto real;

A nível científico este trabalho pretende responder à seguinte questão de investigação:

*Qual a viabilidade da utilização de técnicas de otimização para melhorar o processo de decisão em Medicina Intensiva?*

### **1.3. Estrutura do documento**

Este trabalho de investigação teve como base a utilização do processo de *Benchmarking* e a metodologia de investigação *Quantitative Research*. A realização deste trabalho exigiu uma extensa revisão da literatura e análise de casos de estudo pelo que esta dissertação contém vários documentos científicos referenciados. No início deste documento, o [capítulo 1](#) pretende sublinhar que áreas de estudo estão inerentes a esta dissertação, onde é entendido o ambiente aplicacional da questão de investigação e que objetivos foram definidos para responder a tal questão. O [capítulo 2](#) explica de uma forma geral, o que é a medicina intensiva e que necessidades existem, o que podemos esperar de um sistema de apoio à decisão, e a terminar é efetuada uma breve abordagem à importância e utilidade de técnicas e modelos de otimização. Em qualquer trabalho com este âmbito é necessário um guia para atingir o que pretendemos, e, para isso, no [capítulo 3](#) são apresentadas as metodologias abordadas e os objetivos compreendidos. No [capítulo 4](#) iniciam-se as etapas de um processo *benchmarking*. Neste capítulo o INTCare é abordado de forma sucinta de modo a explicar o que faz, onde se aplica e de que forma a questão de investigação contribuí

para a sua melhoria. Após a recolha das técnicas que melhor se enquadravam neste tipo de sistemas de aprendizagem automática, presente no [capítulo 5](#), foi necessária uma exploração aprofundada das técnicas e modelos de otimização que enriqueceram o seu conhecimento e a sua aplicabilidade em casos reais, explicada no [capítulo 6](#). Para mitigar o trabalho desenvolvido até então, no [capítulo 7](#) foram postas à prova várias técnicas identificadas, através de uma ferramenta de alto-nível *open-source*: *KEEL* que permitiu quantificar as suas capacidades de otimização. Antes da sua classificação, foram explorados e tratados os dados provenientes do CHP. Assim, e utilizando dados reais, é expectável demonstrar que as técnicas seleccionadas contribuam para a construção / melhoria dos sistemas inteligentes na área da saúde e ao mesmo tempo, viabilizem a questão de investigação. Para finalizar, no [capítulo 8](#) explicam-se os fenómenos e as conclusões que se tiram desta dissertação e no [capítulo 9](#) são relatadas algumas dificuldades ou limitações que decorreram ao longo deste período e também enaltece o caminho que fica aberto para possíveis trabalhos de investigação complementares.

## 2. Revisão da Literatura

Para obter informações sobre os sistemas de apoio à decisão existentes, como é que são compreendidos e executados, foram pesquisadas técnicas que otimizassem a resolução dos problemas que esses sistemas procuram solucionar, distinguindo-as pela sua natureza, comportamento, características e aplicações no mundo real. Na parte avaliativa do processo de *benchmarking*, foram utilizadas as técnicas presentes na ferramenta selecionada (por critério de aplicabilidade, interoperabilidade e tempo de implementação) e na parte quantitativa testes estatísticos que permitiram perceber de uma forma não textual as suas acuidades médias na classificação do evento pretendido e as suas taxas de erro. A seleção das melhores técnicas teve como fatores a acuidade média, tempo de execução e interpretabilidade dos resultados. Para tal, esta revisão de literatura teve como estratégia base a pesquisa de conceitos relacionados com o tema dissertação para a qual foram utilizados diversos motores de pesquisa científicos:

- ScienceDirect;
- Web of Knowledge;
- Springer;
- IEEE Xplore;
- Google Scholar;
- B-on;
- Scopus.

As palavras-chave que foram utilizadas para a pesquisa foram:

- *Optimization*
- *Optimization techniques*
- *Optimization techniques benchmarking*
- *Optimization Heuristics*
- *Intensive Medicine Optimization*
- *DSS Optimization*
- *DSS Intensive Medicine*
- *DSS Benchmarking*
- *Benchmarking*
- *Benchmarking Heuristics*



- *Benchmarking Heuristics Intensive Medicine*
- *Machine Learning*

A pesquisa foi efetuada utilizando os mesmos termos em outras linguagens (ex. Português) como também uma combinação de palavras-chave.

A escolha dos artigos consultados seguiu os seguintes critérios:

- Data (preferencialmente artigos posteriores ao ano 2000)
- Relevância (preferencialmente mais de 10 citações por cada ano passado desde a publicação)
- Autor (mais publicações nas palavras-chave abordadas)

Como este trabalho teve como base a revisão de literatura, foi necessário elaborar uma pesquisa exaustiva na identificação das técnicas mais pertinentes no âmbito aplicacional.

## **2.1.Medicina intensiva**

Em qualquer serviço de saúde, a presença das tecnologias de informação e comunicação (TIC) é algo que é comum no seu funcionamento. As TIC são utilizadas para, por exemplo, efetuar o registo eletrónico do diagnóstico atual de um doente, prescrições eletrónicas e armazenar imagens de diagnósticos. A informação registada / armazenada é reutilizada pelas equipas médicas com vários níveis de sucesso [14,15,16].

As pessoas que apresentam doenças graves são normalmente admitidas nas Unidades de Cuidados Intensivos (UCI) para que consigam manter as suas funções fisiológicas vitais através de vários dispositivos de suporte, como medicamentos e dispositivos médicos (ventiladores, máquinas de sinais vitais, outros), até que o doente volte a ter as suas funções autonomamente [17]. No que diz respeito às ações, mais concretamente, os doentes esperam ter assegurado um tratamento correto, seguro e eficaz com uma elevada possibilidade em alcançar o efeito desejado [18].

Contudo, Morris [19] categorizou 236 variáveis diferentes em um registo médico neste tipo de unidades e concluiu que isto excede completamente a capacidade intelectual de um ser humano. O processamento de informação heterogénea em grandes quantidades torna-se crítico, o que faz dos ambientes nas unidades de medicina intensiva extremamente intensos tendo em conta a quantidade de dados a serem considerados pelas equipas médicas [20].

Esta realidade expõe a fragilidade das equipas e de alguns modelos de apoio à decisão existentes atualmente, perante situações onde o tempo de decisão sobre a análise de múltiplas variáveis é crítico e deve ser o melhor possível. As técnicas de otimização têm características interdisciplinares e na área da saúde pretendem otimizar as soluções de modo a, por exemplo, antecipar o processo de decisão sobre um doente, mantendo a monitorização das suas funções vitais, zelando pela sua segurança ao ajudar a perceber qual o melhor método ou procedimento de tratamento a aplicar.

## **2.2.Sistemas de apoio à decisão**

Os Sistemas de Apoio à Decisão (SAD) procuram suportar a resolução de problemas estruturados (situações repetitivas em que são conhecidas todas as regras e limitações), semiestruturados (em que não são conhecidas todas as regras) e não estruturados (onde as regras são desconhecidas, pouco frequentes e assentam na experiência e conhecimento das decisões tomadas) complexos e com um grande número de opções por analisar, que por sua vez impõe sérias penalidades quando uma decisão errada é tomada. Assim, urge a necessidade de existirem ferramentas de apoio à decisão capazes de lidar com decisões urgentes e disponibilizar a informação de acordo com o alinhamento do estilo cognitivo no modelo de decisão [21,22]. Estas ferramentas podem ser baseadas em regras ou ter um conjunto de técnicas de Inteligência Artificial e aprendizagem automáticas.

## **2.3.Técnicas e modelos de otimização**

A expansão de técnicas e modelos de otimização tem levado um ritmo surpreendente nas últimas décadas. Foram desenvolvidas novas técnicas teóricas e algorítmicas, tanto na rápida dispersão por outras disciplinas como no aprofundamento do conhecimento de todos os aspetos no campo da otimização. Ao mesmo tempo, a natureza interdisciplinar é uma das tendências mais marcantes na otimização, tornando-se uma ferramenta básica em todas as áreas de matemática aplicada, engenharia, medicina, economia entre outras ciências [23].

Na medicina, a otimização tornou-se omnipresente. O recurso do poder computacional para aplicações na área da medicina abriu muitas questões e problemas desafiadores inerentes a estas comunidades. As técnicas matemáticas (contínuas e discretas) desempenham um papel cada vez mais fundamental na compreensão de vários problemas críticos na medicina. Naturalmente, a otimização é uma ferramenta de fundamental importância devido à

limitação dos recursos envolvidos e à necessidade de uma melhor tomada de decisão no menor tempo possível.

Começando por uma estimativa inicial sobre o problema em questão, como representa a Figura 1, um algoritmo de otimização é um processo iterativo onde após um certo número de iterações, pode convergir para uma solução que idealmente será a solução ótima para o problema.

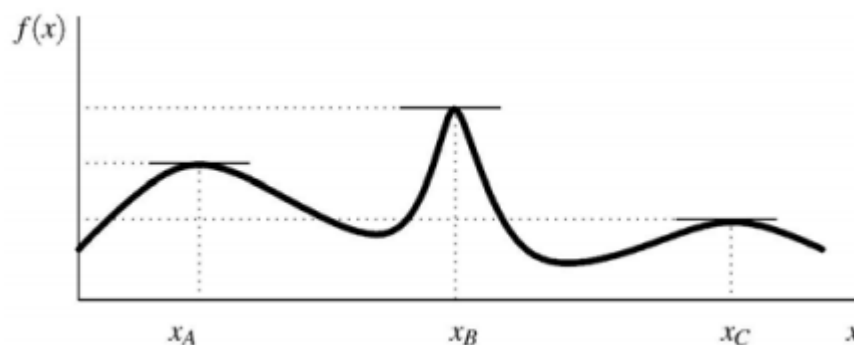


Figura 1 – Soluções locais e global ótimas (retirado de Gili et al. [23])

Durante iterações que decorrem no processo, as soluções que surgem são o estado da evolução que são atraídas de acordo com equações matemáticas ou um conjunto de regras para as soluções convergentes de um sistema auto-organizado. Como resultado, a sua habilidade em representar um sistema auto-organizado mostra-nos algumas características emergentes e a sua capacidade evoluir.

De uma forma sucinta, a seguir são descritas algumas das características mais importantes para este tipo de métodos de otimização [23]:

- Métodos trajetória: A solução atual é parcialmente modificada ao pesquisar a sua vizinhança. Tipicamente utilizados por exemplo, em métodos *threshold* e pesquisa tabu.
- Métodos descontínuos: Onde o espaço da solução atual está completamente disponível para a nova solução. Induzidos pelos operadores genéticos, representam casos de computação evolucionária.
- Métodos *Single-solution based*: É processada uma solução por iteração. Aplicados em métodos *simulated annealing* e *iteration local search*.
- Métodos *Population-based*: São métodos onde a população dos agentes contribui para uma experiência de pesquisa coletiva. Estes métodos são representados por casos de computação evolucionária e *swarm intelligence*.

- Métodos *Guided search*: São métodos caracterizados anteriormente, mas com a particularidade de recorrerem à memória, incorporando regras adicionais ou dicas do que foi concluído no espaço de pesquisa. Por exemplo na pesquisa tabu e *differential evolution*.
- Métodos *Unguided search*: Estes representam métodos que não utilizam memória, como por exemplo, métodos *threshold*.

É indispensável salientar a possibilidade de criar métodos híbridos, onde é possível colmatar as fraquezas de um método com a potencialidade de outro, como por exemplo na necessidade de obter uma melhor troca entre as capacidades de uma heurística ao diversificar o espaço de pesquisa e de intensificar a experiência, sendo distinguidos por ser de baixo-nível e alto-nível e caracterizados por *relay* e hibridização co-evolucionária [23].

Uma hibridização baixo-nível repõe uma componente de um determinado método de otimização com a componente de outro método. Nos casos de hibridizações alto-nível, os métodos são auto-suficientes, incorporando-se. Hibridizações *relay* combinam sequencialmente diferentes métodos, enquanto nos co-evolutivos os métodos cooperam entre si. Por outras palavras, as soluções que estes métodos híbridos geram podem ser caracterizados como [23]:

- Homogéneos: são utilizados os mesmos métodos;
- Heterogéneos: combinam diferentes métodos;
- Parciais: pesquisam o espaço da solução segmentado;
- Globais: exploram o mesmo espaço de solução;
- Especialistas: combinam métodos que resolvem diferentes problemas;
- Generalistas: resolvem o mesmo problema.

De uma forma mais detalhada, na [secção 6.1](#) seguem-se mais exemplos de métodos de otimização e alguns casos práticos.

### **3. Abordagem metodológica**

A abordagem metodológica pretende guiar e clarificar os métodos e objetivos inerentes ao trabalho desenvolvido. Nas subsecções seguintes, são apresentadas as metodologias utilizadas para o cumprimento desta dissertação.

#### **3.1. Investigação Quantitativa**

O objetivo na realização de uma investigação quantitativa é a determinação da relação entre algo (variável independente) com outro objeto (variável dependente ou resultado) dentro de uma população, caracterizar, contar e construir modelos estatísticos que expliquem o que é observado. Este tipo de investigação pode ser descritivo (associações entre variáveis) ou experimental (causalidade entre variáveis) que lida com números e lógica, focando-se em dados estáticos numéricos e detalhados, com raciocínio convergente, que gera espontaneamente uma variedade de ideias sobre um problema de pesquisa [24,25].

Esta investigação teve uma abordagem experimental que se destina a estabelecer a causalidade entre variáveis e será estruturada da seguinte forma:

- Introdução, onde identifique a questão da investigação, a revisão de literatura sobre e uma descrição teórica;
- Metodologia, onde se descreve como cada objetivo do estudo foi alcançado, na amostra e estudo da população, recolha de dados e a sua análise;
- Resultados, devem ser utilizadas análises estatísticas presentes em uma ordem lógica e sequencial sobre elementos não textuais;
- Discussão deve fundir a revisão da literatura com o contexto teórico, interpretar os resultados, descrever as tendências, discutir implicações e limitações;
- Conclusão, onde são resumidos os resultados, feitas recomendações e pesquisas futuras em eventuais lacunas existentes que não foram abordadas.

As vantagens desta abordagem metodológica permitem:

- Um estudo mais amplo, reforçando a generalização de resultados;
- Maior objetividade e precisão nos resultados ao empregar procedimentos prescritos para garantir a validade e confiabilidade;
- A aplicação de normas bem estabelecidas, permitindo a replicação, análise e comparação com estudos semelhantes;

- Resumir muitas fontes de informação e compará-las entre categorias ao longo do tempo;
- Evitar ideias pessoais usando técnicas computacionais acreditadas.

As desvantagens nesta abordagem consistem em:

- Perda de detalhe contextual;
- Tem uma abordagem estática e rígida;
- As questões refletem o ponto de vista do investigador e não a questão da investigação;
- Os resultados têm menos detalhes sobre o seu comportamento, atitudes e motivação;
- O conjunto de dados pode ser insuficiente;
- Os resultados só fornecem descrições numéricas.

O objetivo de realizar uma pesquisa quantitativa experimental foi o de explicar fenómenos através da recolha de dados quantitativos, que foram analisados por métodos matemáticos e que ajudam a determinar a relação entre otimização e *benchmarking* através da sua avaliação de desempenho dos métodos que são utilizados nessa análise [24,25].

### **3.2. Processo de *benchmarking***

Uma ação de *benchmarking* pretende descobrir práticas específicas responsáveis pelo alto desempenho, entender como as melhores práticas funcionam, como são obtidas e adaptadas ao contexto pretendido [26]. Então, esta ação passa por abordar uma possibilidade de *benchmarking* para adotar futuras métricas a técnicas de otimização em sistemas de apoio à decisão.

Os primeiros esforços passaram pelo levantamento de métricas quantitativas e qualitativas para a investigação, com o intuito de examinar de uma forma global a possível aplicabilidade do conceito de *benchmarking* a técnicas de otimização que podem ser utilizados para recolher e analisar dados, mediante a sua performance, melhoria e vantagem competitiva.

Após essa recolha houve a necessidade de efetuar um levantamento das ferramentas que permitiam testar as técnicas identificadas, para poder demonstrar a sua capacidade, resultando no final, uma tabela comparativa dos testes representando as suas forças e fraquezas [26].

Entre as várias possibilidades, o processo de *benchmarking* pode ser composto por cinco fases como descrito por Khurram Bhutta e Faizul Huq [27]. Este processo foi adaptado tendo em conta os requisitos e as características do estudo:

- Planear o exercício: esta fase identifica que técnica de otimização e modelo de apoio à decisão está subjacente ao INTCare, o que fazem e o que atingem;
- Definir e apresentar os modelos de apoio à decisão: esta etapa do processo de *benchmarking*, não são as pessoas que têm de ser identificadas e avaliadas, mas sim as áreas e os modelos que devem ser analisadas tendo por base as principais técnicas de suporte aos SAD. O *benchmarking* a realizar terá por base algoritmos que podem ser incorporados nos modelos definidos neste ponto;
- Recolher os dados: recolha de técnicas de otimização a incorporar nos modelos de apoio à decisão;
- Analisar dados: os modelos de apoio à decisão e técnicas de otimização serão avaliados globalmente e depois serão analisados tendo em conta os que obtêm melhores resultados no contexto da medicina intensiva;
- Tomar medidas: caso se identifiquem técnicas e modelos com melhores capacidades preditivas do que as atuais implementadas, propor uma alteração ao sistema INTCare.

As principais características aplicadas a este trabalho são:

- Os dados podem ser recolhidos utilizando instrumentos de pesquisa estruturados;
- Pode haver replicação ou repetição na pesquisa, devido à alta confidencialidade;
- A pesquisa deve ter definido o propósito da investigação bem definido para os objetivos das respostas pretendidas;
- Todas as questões do estudo são cuidadosamente planeadas para a recolha de dados.

#### Resultado esperado

Em suma o objetivo principal de uma pesquisa quantitativa sobre um processo *benchmarking* é classificar características, contá-las e construir modelos estatísticos na procura de explicar o que é observado [24,25].

## 4. Planeamento do exercício

Este capítulo identifica a primeira etapa do processo de *benchmarking* que faz uma primeira análise do exercício e um enquadramento tendo em conta o contexto em que se insere, ou seja, que previsões e modelos de apoio à decisão que estão subjacentes ao INTCare e determina os fatores a avaliar nas suas análises preditivas.

### 4.1. Etapas do *Benchmarking*

Como diz no início deste capítulo, é iniciado o processo de *benchmarking*, e o planeamento do exercício identifica o sistema INTCare, descrevendo a tecnologia que está subjacente a esse Sistema inteligente de apoio à decisão, mostrando o que faz e como adquire a sua informação.

A constituição da avaliação apresenta os modelos de apoio à decisão, as suas áreas e como devem ser analisadas tendo por base as principais técnicas de suporte aos SAD, tais como Sistemas de aprendizagem automática, Representação do conhecimento, *Text* e *Data mining* e técnicas *Multi-purpose*. O *benchmarking* a realizar terá por base algoritmos que podem ser incorporados nos modelos definidos neste ponto.

Para a recolha dos dados, pretendeu-se enumerar técnicas de otimização *single-solution based* e *population-based* a incorporar nos modelos de apoio à decisão identificados anteriormente.

Na análise dos dados recolhidos, as técnicas de otimização foram avaliadas através da ferramenta KEEL e depois analisadas tendo em conta os que obtêm melhores resultados no contexto da medicina intensiva.

Para proposta de medidas, caso se identifique, sugerem-se a(s) técnica(s) e modelo(s) com melhores capacidades preditivas do que as atuais implementadas, propondo uma alteração ao sistema INTCare.

### 4.2. INTCare

Este trabalho enquadra-se no projeto de investigação INTCare que tem o seu enfoque na medicina intensiva. A previsão da falência de órgãos e condições clínicas do doente foram o primeiro objetivo concluído pelo sistema inteligente [28]. Na criação do projeto em 2009, detetou-se um elevado número de dados no formato de papel ou armazenados manualmente



na base de dados que implicou um conjunto de barreiras que impossibilitou a sua execução. Com isso, a apresentação de uma nova solução para o serviço desencadeou um conjunto de estudos que tornou possível definir quais eram as lacunas do sistema de informação presente na UCI [29,30].

Reformulado o projeto, a solução adotada foi baseada em agentes inteligentes [31]. Passam por executar funcionalidades automaticamente como a aquisição e processamento de dados, sendo ao mesmo tempo, os agentes responsáveis por preparar o sistema a ser implantado no ambiente pretendido [32].

Composto por um conjunto de módulos integrados, o INTCare oferece análises preditivas relacionadas com a falha de órgãos que tenham situações como:

- Problemas cardiovasculares;
- Coagulação sanguínea;
- Complicações respiratórias;
- Doenças hepáticas;
- Doenças renais.

Como resultado, o INTCare tornou-se desde então um SIAD que atua de forma automática e em tempo real, a fim de dar novas informações (conhecimento) para os tomadores de decisão na UCI (médicos e enfermeiros).

#### ***4.2.1. Sistema inteligente de apoio à decisão***

O SIAD envolve um conjunto de sistemas e mudanças no ambiente da UCI. Este sistema caracteriza-se por atender alguns requisitos [32,33]: aprendizagem *online*, aprendizagem em tempo real, adaptabilidade, *Data Mining*, otimização, agentes inteligentes, Precisão de segurança, *Pervasive* / ubiquidade, privacidade, acesso seguro a partir do exterior do hospital, Política de Utilização, integração e interoperabilidade.

A arquitetura deste sistema ilustrada na Figura 2 foi projetada para produzir conhecimento, diferente do que normalmente é usado, principalmente em hospitais e representa uma nova forma de interoperar sistemas, combinando em tempo real componentes de aquisição de dados e análise de dados [34]. Ao serem recebidos, os dados são validados e pré-processados de acordo com o doente em questão para uma correta identificação dos valores reais, ou seja, os dados estão entre o intervalo de valores possíveis definido pela UCI [32]. Depois de os

dados serem armazenados na base de dados são transformados de acordo com o objetivo, tornando cada variável em um atributo de entrada do motor de inferência. Finalmente, o resultado é produzido no motor de inferência, através do uso de *Data Mining* e processamento automático de dados. Os resultados são apresentados no INTCare e no registo eletrónico de enfermagem. Ao mesmo tempo, as equipas médicas podem consultar todo o conhecimento produzido no sistema de inferência em qualquer lugar e a qualquer hora, usando para o efeito um dispositivo (móvel ou fixo) com acesso à internet. Através do acesso remoto, os profissionais da UCI podem consultar os valores do sistema de sinais vitais, sistema de pontuação e eventos críticos e a previsão do estado do doente para diversos problemas. O sistema é reconhecido por ter algumas características particulares de computação *pervasive* [35]: escalabilidade, a heterogeneidade, a integração, a invisibilidade e consciência de contexto. Este trabalho enquadra-se no INTCare pelo facto de que os modelos desenvolvidos até ao momento requererem processo de otimização, pelo que é necessário fazer um levantamento das técnicas e algoritmos existentes a fim de perceber aqueles que melhor se podem enquadrar em Medicina Intensiva.

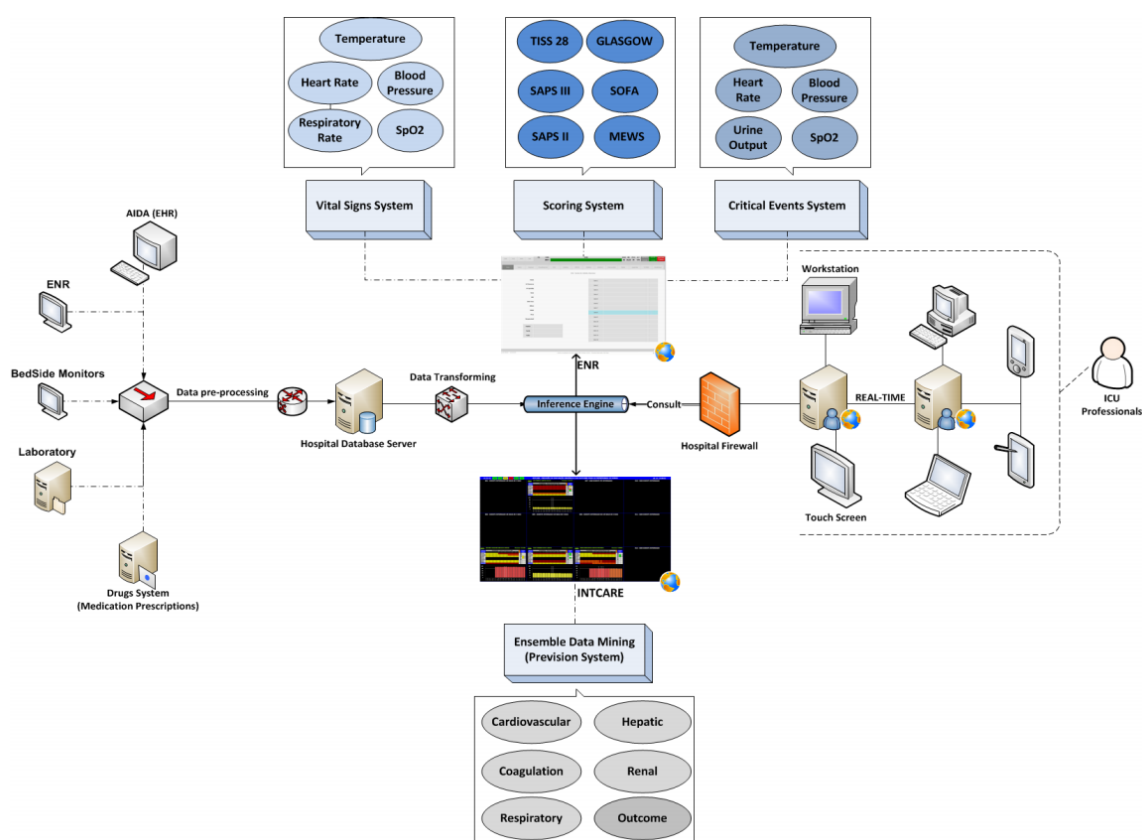


Figura 2 – Arquitetura do SIAD (retirado de Portela et al. [34])

### 4.2.2. Aquisição de dados

O Sistema utiliza dados obtidos por duas formas: manual e automaticamente. Muitos dados foram adquiridos manualmente no início, e registados pelos enfermeiros. Hoje em dia o cenário mudou, os dados são automaticamente recolhidos e guardados na base de dados recorrendo a procedimentos automáticos dotados de agentes inteligentes. Os dados adquiridos para a base de dados estão disponíveis *online* através do registo eletrónico de enfermagem. Por exemplo os sinais vitais e dados de ventilação são adquiridos automaticamente e os resultados estão disponíveis em tempo real através da plataforma como exemplifica a Figura 3.

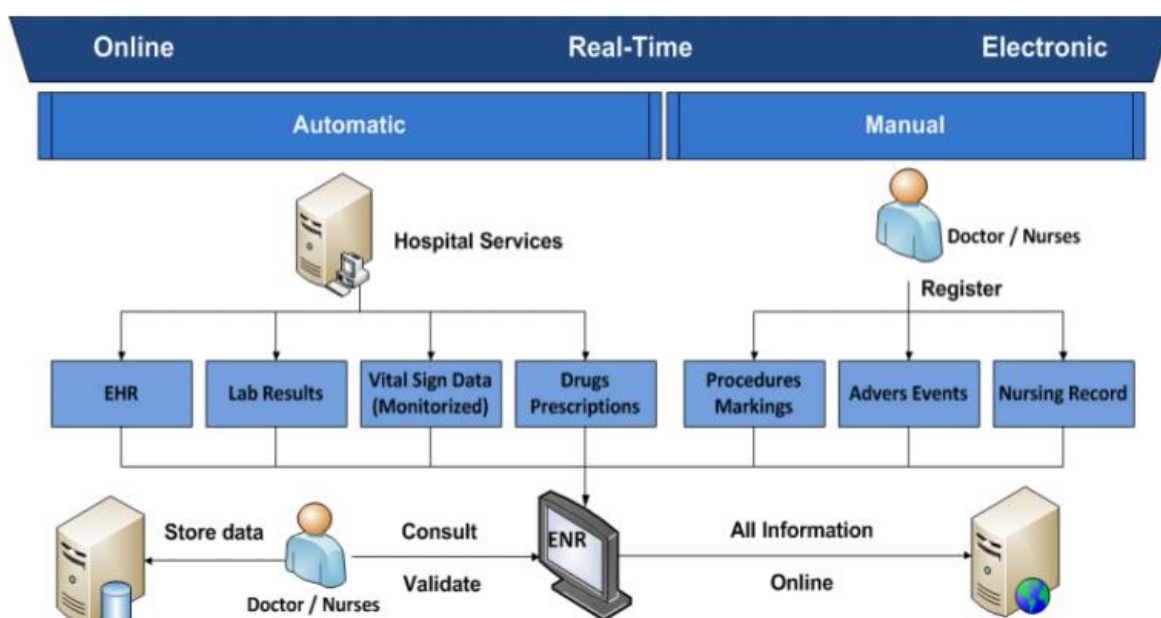


Figura 3 – Arquitetura da aquisição de dados (retirado de Portela et al. [34])

Os dados deste sistema são provenientes de várias fontes:

- Monitores de cabeceira (MC) (ex. sinais vitais e ventiladores);
- Registo eletrónico de enfermagem (REE / ENR);
- Registo eletrónico de saúde (RES / EHR);
- Laboratório (LB);
- Sistema terapêutico (ST).

Os atributos que estas fontes geram, são armazenados na base de dados e utilizados por cada um dos sub-sistemas de decisão:

- Eventos críticos (EC);
- Sinais vitais (SV);
- Sistema de classificação (SC);
- Conjunto de algoritmos *Data mining* (CADM).

A Tabela 1 apresenta os dados que são guardados em tempo real, provenientes das fontes de dados e utilizados pelos sub-sistemas respetivamente.

Tabela 1 – Fontes de dados do SIAD

<b>Variáveis</b>	<b>Fonte de dados</b>	<b>EC</b>	<b>SV</b>	<b>SC</b>	<b>CADM</b>
Pressão arterial e ritmo cardíaco	MC / REE	X	X	X	X
Frequência respiratória	MC / REE		X	X	
Saturação de oxigénio (SpO2)	MC / REE	X	X	X	X
Temperatura	MC / REE	X	X	X	X
Vasopressores	ST			X	X
Idade, dados de admissão e alta	RES			X	X
Doenças crónicas	RES			X	
Eventos críticos e procedimentos	RES			X	
Alerta Voz Dor Não resposta	REE			X	
Glasgow	REE			X	X
Urina /diurese	REE	X		X	X
Albumina e nitrogénio ureico sangue	LB			X	
Bilirrubina /creatinina	LB			X	X
FiO2 e PaO2 / Leucócitos	LB			X	X
Glucose / HCO3 / Leucócitos / PH	LB			X	
Plaquetas / Potássio / Sódio / Ureia	LB			X	

## 5. Constituição da avaliação

A avaliação dos métodos de aprendizagem automática passa por uma abordagem teórica, a forma como o método se comporta, algumas vantagens e/ou desvantagens e casos práticos, tais como modelos de aprendizagem automática, representação de conhecimento, *Text/Data mining* e modelos *multi-purpose*. Após esta avaliação, na secção seguinte encontra-se a recolha de técnicas que otimizam estes modelos de aprendizagem, distinguidas pelo tipo de problemas *single-solution based* ou *population-based* que abordam. Como comparação, tentou-se manter o critério de verificar velocidades de treino, utilização de memória, acuidade preditiva, interpretabilidade, velocidade preditiva e de ajuste, capacidade de lidar com variáveis categóricas, ajuste de parâmetros, se utiliza operadores genéticos, a sua capacidade de intensificar e diversificar no espaço de pesquisa e também casos práticos com a sua aplicação.

### 5.1. Sistema de aprendizagem automática

Um Sistema de Aprendizagem Automática (SAA) está focado no desenvolvimento de sistemas que aprendem através dos dados [4,36]. Envolve uma fase de treino onde o sistema aprende a completar determinadas tarefas (preditivas ou de classificação), utilizando um determinado conjunto de dados contendo informação representativa do problema. Após a fase de treino, o sistema é capaz de analisar novos dados contendo o mesmo grupo de parâmetros e sugerir uma previsão. Infelizmente, não existe um método perfeito que seja capaz de resolver um determinado problema, pois existem vários que oferecem melhores acertos e previsões facilmente [37] estando a sua escolha dependente da área em estudo. Este é um aspeto que deve ser considerado antes de se desenvolver um SAD baseado nestes modelos. Nas seguintes subsecções serão considerados alguns desses métodos.

#### 5.1.1. Regressão Lógica

A Regressão Lógica (RL) [38,39] procura determinar a influência de variáveis independentes ao prever a dependência de uma variável categoricamente (uma variável dependente tem um número limite de valores). Esta técnica é particularmente útil para identificar as variáveis mais discriminativas em um conjunto de dados quando existem muitas variáveis a considerar e o seu *output* só pode ter valores pré-definidos (ex. positivo

ou negativo). Estas técnicas podem ser úteis para identificar variáveis pertinentes e combinar com outras técnicas de SAA. Estes modelos tendem a ser menos robustos que as Redes Neurais Artificiais (RNA) e os *Support Vector Machine* (SVMs) particularmente quando estamos perante conjunto de dados complexos. No entanto, como são usados modelos lineares mais simples para processar decisões rápidas é mais fácil interpretar o *output* e a forma como a decisão foi tomada [37].

Ji et al. [40], adotou a RL no desenvolvimento de um SAD para ferimentos traumáticos. Os peritos nessa área devem considerar muitos aspetos num curto espaço de tempo. Na primeira fase, RL foi utilizada para simplificar o conjunto de dados de treino as variáveis mais preditivas (ex. idade, distúrbios de coagulação sanguínea). Na segunda fase, o conjunto de dados simplificado foi utilizado para treinar SADs recorrendo a outras técnicas de aprendizagem automática. O seu desempenho preditivo baseado em regras foi de 82% com todas as variáveis e 84% quando apenas utilizou as variáveis mais importantes. Já o desempenho do sistema recorrendo às mesmas variáveis foi de 89,7% e de 93,1% na previsão do internamento na UCI para doentes com lesões traumáticas cerebrais.

### **5.1.2. Support Vector Machines**

Os *Support Vector Machine* (SVM), apresentados por Vapnik [41], são instrumentos poderosos e complexos que se ajustam particularmente quando a tarefa de classificação é difícil [42]. Um modelo SVM representa exemplos de um conjunto de dados como pontos no espaço para que categorias distintas fiquem divididas por um espaço o mais amplo possível [37,43]. Os casos analisados são mapeados ficando divididos tendo em conta a sua categoria, espaço e previsão. É um método eficiente na resolução de problema no reconhecimento de padrões e regressões, bem como na análise de documentos escritos à mão, imagens e previsão de séries temporais [41]. O SVM cria vários hiperplanos no espaço na tentativa de encontrar a divisão ótima dos dados.

A Figura 4 – Classificação linear (ex. RNA) vs. não-linear (ex. SVM) (retirado de Fraccaro P, et al.[37]) apresenta exemplos de categorizações no espaço com as diferentes classificações (pontos pretos e brancos). Na Figura 4(a) está representado uma tarefa de classificação simples onde um classificador linear (ex. RNA) consegue fazer uma separação eficaz. A Figura 4(b) representa uma tarefa de classificação complexa onde os métodos lineares não conseguem classificar todas as instâncias e onde os classificadores não-lineares

são necessários. A Figura 4(c) mostra como o SVM consegue projetar as instâncias da Figura 4(b) utilizando o “kernel trick” [42] para mapear não linearmente os dados de entrada para um espaço multidimensional, que torna a classificação uma tarefa mais fácil [37].

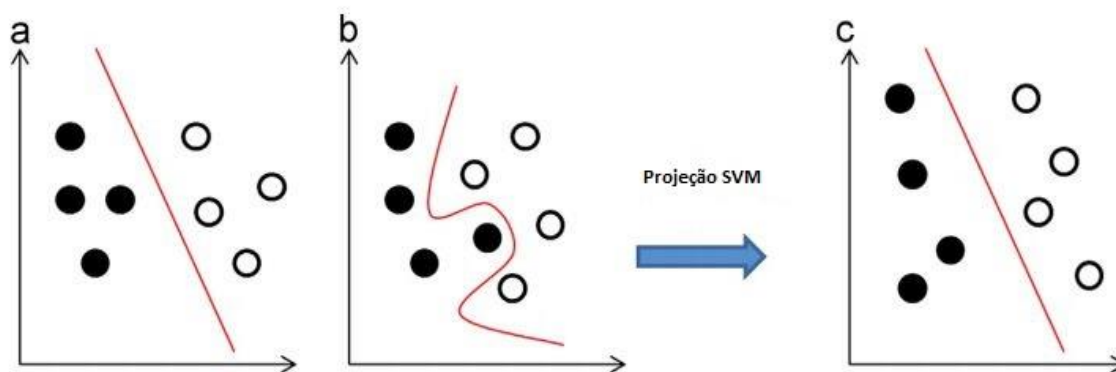


Figura 4 – Classificação linear (ex. RNA) vs. não-linear (ex. SVM) (retirado de Fraccaro P, et al.[37])

Chi et al. [44] utilizou os SVM para criar um SAD assente na avaliação da probabilidade de o hospital maximizar os tratamentos para um determinado doente. Um médico recomenda um tratamento para um doente e o SAD combinando a recomendação com as condições do doente e outras variáveis (instalações, especializações, tipo de admissão, idade, sexo, recursos hospitalares), para avaliar os hospitais apresentando uma maior probabilidade de eficácia no tratamento desse doente.

Cho et al. [45] utilizou os SVM para prever o aparecimento de diabetes em 284 doentes. Usando um conjunto de dados de 184 variáveis, a classificação foi treinada para prever o início dos sintomas 2 a 3 meses antes de o doente ser diagnosticado com essa patologia, desempenhando um grande esforço na previsão perante um conjunto de dados irregular e desequilibrado onde métodos estatísticos (ex. RL) não conseguem operar.

Outros casos de sucesso na utilização de SVMs como no diagnóstico de paralisia cerebral [46], Parkinson [47], e cancro [48,49] realçam a importância destas técnicas.

### 5.1.3. Redes neuronais artificiais

Este modelo matemático, conhecido por redes neuronais artificiais (RNA), conceptualmente parecido com SVMs [37], simula o processo de aprendizagem no cérebro humano utilizando neurónios artificiais interligados numa rede que identifica padrões nos dados [50]. Uma rede neuronal, como representado na Figura 5 normalmente tem alguns *inputs* e produz um ou

mais outputs empregando algoritmos de aprendizagem incremental para processar e modificar a intensidade das ligações entre os inputs, outputs e camadas escondidas da rede, sendo verificados padrões entre os dados [37].

A adoção de modelos neuronais traz algumas vantagens. Implementam-se sem precisar de muito treino estatístico, são dotados de habilidades de implicitamente detetar relações não lineares complexas entre variáveis dependentes e independentes, bem como a habilidade de detetar todas as interações possíveis entre variáveis de previsão [37].

As desvantagens centram-se no seu comportamento racional. A perceção bem como a decisão é executada através das camadas escondidas que é trivial para o utilizador em perceber o que foi decidido e porquê, o que torna não propenso para possíveis ajustes (porque o modelo descreve o erro e o ruído aleatório invés da relação subjacente nos dados) [51]. Porém, têm-se feito esforços na perceção dessa limitação [52].

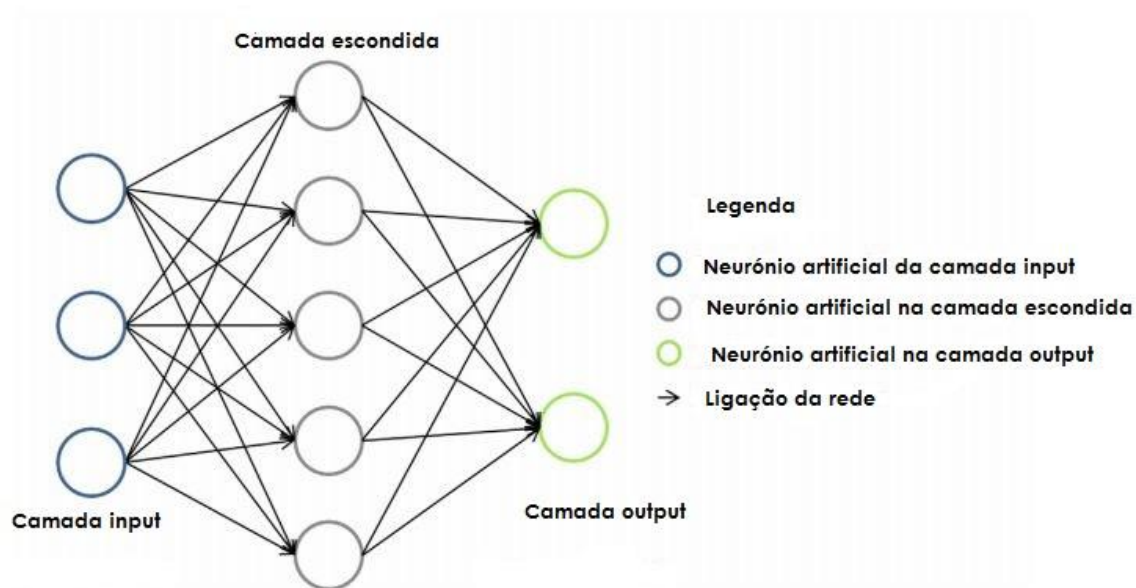


Figura 5 – Exemplo de uma RNA (Fraccaro P, et al.[37])

As RNAs na medicina surgiram no início da década de 90 [53] e rapidamente ganharam popularidade [54]. Aplicações em cardiologia [55-58], ortopedia [59], psiquiatria [60], lombalgia [61], cancro [62-65], gastrite [66], gastrite crónica [67], infeções [68-70], alzheimer [71], doenças pulmonares [72,73] e stress por hipertermia [74].



## 5.2.Representação do conhecimento

A representação do conhecimento (RC) está focada em mostrar o conhecimento e factos de um determinado domínio, a fim de criar uma linguagem de descrição do conhecimento (vocabulário) que é compreensível e explorado em sistemas de informação [75]. O vocabulário pode ser combinado com um sistema de reconhecimento automático e por conseguinte, poder fazer inferências no ambiente. Por exemplo, os dados do doente (vocabulário) podem ser fundamentados automaticamente (regras de práticas clínicas) para inferir sobre o estado de saúde do doente/tratamento ideal [37].

### 5.2.1. Sistema baseado em Ontologias

Ontologia, em ciências da computação, é a representação formal do conhecimento de um domínio em um formato que é compreensível e executável por sistemas computacionais.

Uma ontologia descreve um conjunto de conceções (ex. doenças, terapias) e as suas relações. Uma vez representado o conhecimento nesse formato, estes sistemas conseguem analisar a interação entre as conceções e relações [37]. A aplicação prática dos algoritmos de otimização poderá utilizar uma ontologia orientada aos dados.

Riano et al. [76] desenvolveu um sistema baseado em ontologias para doenças crónicas. Inicialmente foram recolhidas e transcritas os principais conceitos e relações no tratamento de problemas crónicos para um conjunto de regras que selecionam os conteúdos relevantes para o doente. O segundo processo recorria a esta sub-ontologia para automaticamente transformar os planos de intervenção para que descrevem os tratamentos gerais para a especificidade da doença. No caso de comorbidades<sup>1</sup>, este processo inclui uma integração semiautomática de vários planos individuais para um plano personalizado.

Outro exemplo deste tipo de sistemas foi proposto por Farion et al. [77], onde diferentes ontologias foram usadas para representar os componentes essenciais para um SAD na área da medicina, dados do doente, modelos de SAA na resolução de problemas de decisão clínicos e vários tipos de interface (desktop e móvel) na representação das recomendações invocadas pela equipa médica.

---

<sup>1</sup> Comorbidades – Presença de duas ou mais doenças associadas ao mesmo utente

### 5.2.2. Guideline-based

A representação do conhecimento e técnicas de raciocínio são particularmente úteis para desenvolver guias das melhores práticas. Estas representações de conceitos, relações e decisões permitem transformar em formato eletrónico as melhores práticas para o apoio à decisão. A representação pode ser desenhada através de um guia regulado pelas entidades nacionais ou internacionais de saúde, bem como pode ser representado numa rede de transições [78].

Numa breve abordagem, uma rede de transições são grafos orientados na representação de processos sendo composta por dois tipos de nós, lugares (círculos) e transições (quadrados), ligados por arcos direcionados como é possível observar na Figura 6. Isto permite modelar estados, eventos, condições, sincronizações, paralelismos, escolhas e iterações. Com a adição de cores, é possível modelar os dados, a hierarquia permite estruturar grandes modelos utilizando redes e sub-redes e com a adição do tempo, é possível estimar o tempo necessário em cada transição [79]. Seguindo esta aproximação é possível dotar os sistemas através de um conjunto de dados iniciais para sugerir ações, aconselhar próximas decisões e até mesmo alertar quando uma decisão é tomada incorretamente [37].

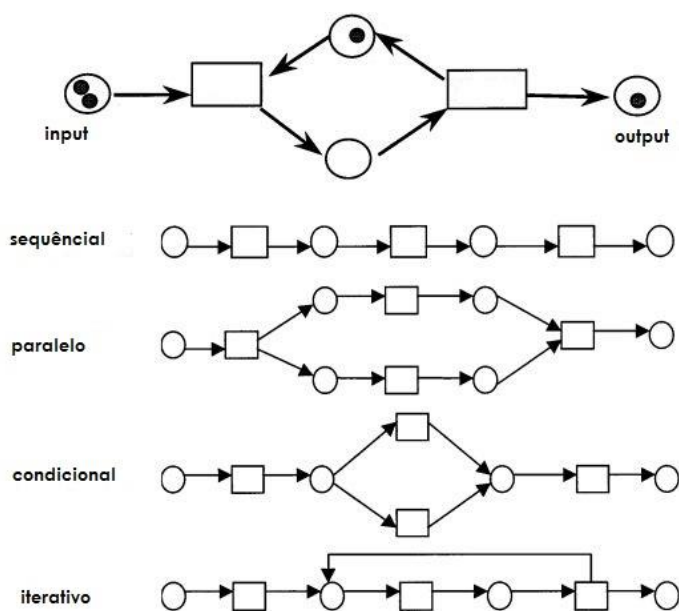


Figura 6 – Exemplos simples dos tipos de rede de transições (Quaglioni et al. [78])

Choi et al. [80] abordou este método de representação de conhecimento para apoiar enfermeiros na triagem e gestão de depressões dos doentes. Os principais conceitos nessa área foram traduzidos para um formato computável que permitiu a execução dos dados do

doente, a nível de extração, categorização e representação dos conceitos relacionados com a depressão.

Por outro lado, Martinez-Garcia et al. [81] recorreram a esta orientação para melhorar a gestão dos doentes com múltiplas patologias, fornecendo informações relevantes a partir de guias clínicos práticos aplicados numa plataforma web, permitindo que todos os profissionais de saúde envolvidos no atendimento de um doente com comorbidades participassem numa discussão de tratamentos e estratégias. Interligado com os dados do doente, permitia a verificação de segurança nas ações de acordo com o guia de práticas identificadas.

Goud et al. [82] relacionando guias de recomendações em tratamentos de reabilitação na terapia de exercícios, educação, relaxamento e mudanças no estilo de vida. No seu estudo foram abordadas equações de correlação nos conjuntos de dados ajustados mediante a idade do doente, sexo, indicação de reabilitação cardíaca e para o tipo e volume de casos de um centro de saúde em específico. Como resultado a partir de 21 centros, incluindo 2787 doentes, o SAD utilizando as decisões recomendadas pelas diretrizes terapêuticas teve aumentos na terapia de exercícios em 7,9%, na educação terapêutica 25,7% e na terapia de relaxamento 25,5%. A aceitação para a terapia de mudança de estilo de vida aumentou 3,2%. Contudo, esta solução reduziu os casos de subtratamento e excesso de zelo.

### ***5.2.3. Fuzzy logic***

*Fuzzy logic* (FL) é um método probabilístico que tenta emular o raciocínio humano onde o raciocínio é aproximado em vez de corrigido [83]. A principal diferença para os métodos binários lógicos (SAA como RNAs e SVMs) onde os outputs são variáveis e têm como valores “*true*” ou “*false*”, é que as variáveis *fuzzy* podem ter diferentes “graus de veracidade”, e um peso diferente que indica a intensidade da ligação. São particularmente ajustáveis para analisar imagens digitais na procura de formas ou anormalidades e na identificação de *clusters* de dados em conjunto de dados [37].

Esposito et al. [84] utilizou este método probabilístico para avaliar o estado de saúde de pessoas com esclerose múltipla. O sistema analisa imagens cerebrais de ressonâncias magnéticas por parte de doentes com esclerose múltipla para identificar os tecidos normais ou aglomerados brancos anormais de potenciais lesões com formas diferentes. Estes autores identificaram diferentes valores possíveis (ex. volume: pequeno, médio, grande; contraste

dos tecidos: pequeno, grande) e desenvolveram regras *if-then* para avaliar a normalidade/anormalidade do tecido cerebral. Em termos de resultados percentuais, o acerto no conjunto de treino foi de 89,1%, no de teste 88,79%, sensibilidade 0,88% e especificidade 0.88%.

Mago et al. [85] usou a FL para descrever a percepção humana perante um determinado sistema. Adotando um tipo particular de FL (Mapeamento cognitivo distorcido) no apoio ao diagnóstico de meningite. Os especialistas na área representavam o seu conhecimento para um modelo gráfico, composto por nós (variáveis, estados, *inputs* e *outputs*) e relações *fuzzy* entre os mesmos. O modelo foi utilizado para analisar os dados do doente na indicação de presença/ausência da doença representando diferentes graus de veracidade.

Na obtenção de resultados a partir de 56 casos, de 40 obteve-se um acerto de 95%, nos restantes 16 foi analisada a sensibilidade e especificidade do sistema, resultando em 83,3% e 80% respetivamente.

#### **5.2.4. Visualização da Informação**

Visualização da Informação (VI) é um conjunto de tecnologias que recorre a computação visual para lidar com informação abstrata [86]. As suas aplicações permitem analisar dados através da exploração e interação permitindo aos utilizadores entenderem sistemas complexos ou conjunto de dados através da observação das consequências e das interações na visualização [37].

Mane et al. [87] adotou um sistema visual para avaliar a eficácia e os riscos inerentes em diferentes terapias na psiquiatria recorrendo a dados do doente com condições similares a um registo clínico eletrónico. Através da utilização de informação médica, qualquer membro de uma equipa consegue compreender claramente as possíveis opções terapêuticas e os seus resultados em situações anteriores semelhantes.

#### **5.3. Text e Data mining**

O *Data mining* (DM) tem-se revelado uma ferramenta bastante útil no apoio ao negócio e a diversas áreas. A grande maioria da utilização de DM é feita para que se possam identificar oportunidades ou ameaças ao negócio ou para tentar solucionar problemas. Desenvolve-se um processo DM através do processamento de dados, previsão, regressão, classificação, *clustering*, associações, visualização e análise de dados exploratória. As tarefas de DM

podem ser divididas tipicamente em duas categorias, as descritivas e as preditivas. As descritivas têm como objetivo caracterizar as propriedades gerais dos dados numa determinada base de dados e as preditivas pretendem inferir sobre os dados para se poder obter previsões através dos mesmos [88]. As tarefas de previsão mais comuns são a classificação, regressão, deteção de anomalias, enquanto que as descritivas mais comuns são o *clustering*, regras de associação e padrões sequenciais [89].

O método *Text mining* (TM) adota técnicas linguísticas, SAA e estatísticas para extrair informação de alta qualidade de texto não-estruturado (ex. repositórios médicos) [90]. Este tipo de sistemas pode ser amplamente definido como um processo de conhecimento intensivo que analisa a interação do utilizador com um conjunto de documentos ao longo do tempo, através de operações algorítmicas, na busca de informação textual (caracteres, palavras, termos, conceitos, padrões). Estas operações de pré-processamento são necessárias para eficientemente explorar conjuntos de dados, como também em dados não-estruturados (documentos e relatórios existentes no negócio). Não é estranho perceber que TM e DM evidenciem similaridades arquiteturais de alto-nível. Ambos os sistemas regem-se em pré-processamentos de rotina, algoritmos de descoberta de padrões e elementos de apresentação em camadas, tal como funcionalidades de visualização e exploração para melhorar a análise do conjunto de dados retornados. No entanto DM explora e controla muitos padrões nas suas operações de pesquisa de conhecimento que fundamentam o TM [91].

Porque assume que os dados já se encontram estruturados, o pré-processamento de DM assume duas tarefas críticas: inspecionar e corrigir dados utilizando dados redundantes (ex. cópias de dados), normalizando-os na busca de reduzir erros e criar inúmeros *joins* em tabelas com o intuito de adquirir conhecimento e conseguir prever através deles. No entanto, para sistemas TM, as operações de pré-processamento centram-se na linguagem natural, explorando técnicas e métodos nas áreas de *Information Retrieval* e Processamento de Linguagem Natural, identificando e extraíndo dados não estruturados em um formato mais explicitamente estruturado, algo que não se mostra relevante para a maioria dos sistemas DM [91].

### ***5.3.1. Informational retrieval***

É o processo através do qual a informação relevante é obtida a partir de dados não-estruturados e disponibilizado ao utilizador de acordo com a informação específica

pretendida [92,93]. No método *Informational Retrieval* (IR), os documentos ou conjunto de dados são representados por uma coleção de palavras não ordenadas sem qualquer controlo gramatical. É necessária uma fase de indexação preliminar onde as palavras com informação relevante são processadas em termos e representadas através de *queries*. Posteriormente, os algoritmos de pesquisa conseguem retornar informações dos recursos pré-processados recorrendo a métodos estatísticos (ex. frequência na utilização de palavras) [37].

O'Sullivan et al. [94] desenvolveu um sistema para apoiar os médicos pediátricos em recuperar informações de casos específicos de tratamento no domínio da asma, retirando um índice de termos que enfatizam o doente em vez de termos mais utilizados, com revisões sistemáticas da Cochrane Library [95]. Estes documentos foram recuperados e ordenados por algoritmos de pesquisa que apresentavam comentários relevantes para os médicos durante a sua intervenção [37] melhorando a sua precisão e recuperação de informação de 81% e 56% para 93% e 87% respetivamente.

### ***5.3.2. Processamento de Linguagem Natural***

Processamento de Linguagem Natural (PLN), procura derivar o significado da linguagem humana ou natural em forma de *input* através de algoritmos [96] em vez de simplesmente representá-las em coleções de palavras. Estas técnicas podem ser utilizadas para automaticamente extrair, sintetizar ou categorizar informação [37]. Reside em modelos formais ou representações de conhecimento da linguagem a nível fonológico, fonético, morfológico, sintáxico, semântico, pragmático e discurso, que se incluem máquinas de estado, modelos probabilísticos e sistemas lógicos e de regras formais na captura desse conhecimento [96].

Matheny et al. [97] para identificar sintomas de infeções através de narrativas clínicas adotou PNL a fim de criar um sistema de vigilância automática de possíveis infeções não identificadas. Criaram regras de deteção que foram baseadas em conceitos e relações nas principais ontologias biomédicas (ex. Nomenclatura sistematizada de termos clínicos médicos [98]), processando textos e conjunto de dados, com vista a identificar sintomas de tuberculose, hepatite aguda e gripe.

O sistema detetou corretamente 84,7% de acertos positivos, 75,1% em negativos e 0,7% de acertos incertos.

## ***5.4. Multi-purpose***

Metodologias *multi-purpose* combinam atributos e características das outras principais categorias.

### ***5.4.1. Árvores de decisão***

As árvores de decisão (AD) podem ser usadas por SAA bem como por RC [99]. Em SAA, são utilizadas na obtenção de previsões e classificações, começando por um *conjunto de dados* de treino, as ADs constroem um modelo que é composto por vários nodos, intitulados por “folhas” da AD, onde as variáveis de entrada são analisadas de acordo com os valores presentes, o processo de decisão flui por diferentes ramos da árvore com o objetivo de realizar a previsão atribuída ou tarefa de classificação. No entanto, a sua robustez fica comprometida quando o número de variáveis de entrada é grande e por vezes são convertidas em *Random Forests* (RFs) que combinam os resultados de vários conjuntos de dados de treino das ADs.

Na RC, a AD pode ser utilizada na produção de gráficos que representam as possíveis decisões associadas às potenciais consequências. São principalmente utilizadas no desenvolvimento de planos de ação para alcançar um objetivo específico. O uso das ADs permite a criação de regras de decisão mais transparentes e compreensíveis pelos utilizadores que procuram seguir o seu raciocínio pelos nós [37].

Ji et al. [100] combinaram ADs no apoio à tomada de decisão em lesões traumáticas. Utilizando as variáveis consideradas mais importantes (ex. idade, *score* do departamento de emergências traumáticas, *score* da gravidade de lesões) como variáveis de *input*, obtendo bons resultados de previsão através de RL. A sua explicação no uso de ADs recaí na transparência e compreensão do raciocínio das regras de decisão criadas através dos seus nós. A média preditiva obtida utilizando todas as variáveis foi de 82%, e de aproximadamente 84% quando apenas foram consideradas as variáveis importantes. A combinação das técnicas recorrendo apenas às variáveis importantes, traduziram-se numa melhoria para 89,7% no seu tratamento e de 93,1% na previsão de estadia na UCI.

Já Bergman et al. [101] utilizou a RC na comparação de papel e instrumentos de diagnóstico escritos, baseado no livro “*Diagnostic and Statistical Manual of Mental Disorders 4th Edition*” [102], para um sistema baseado em ADs. O utilizador foi orientado pelos vários

ramos com base em respostas “sim”, “não” ou “indefinido”. Se estes critérios preenchessem os critérios até um determinado nível (de acordo com o livro) o programa sugeria automaticamente o diagnóstico. Esta comparação mostrou não haver diferenças significativas entre o papel/instrumento de um método sistematizado.

#### **5.4.2. Lógica Bayesian**

A lógica *Bayesian* é um método de inferência baseado na lei de *Bayes*, que é um teorema estatístico com a capacidade de atualizar a probabilidade estimada para um determinado evento através de conhecimento de novas evidências [103]. As redes *bayesian* são modelos probabilísticos gráficos que consideram um conjunto de variáveis aleatórias e as suas dependências condicionais. De uma perspectiva SAA, estas redes são classificadores de “caixas brancas” muito úteis, significando que todos os parâmetros têm uma interpretação clara e a certeza associada à previsão probabilística é intuitivamente compreensível. [37]

Por exemplo, redes *bayesian* podem representar a probabilidade das dependências entre sintomas e doenças, e dada uma lista de sintomas, podem calcular a probabilidade da presença de várias doenças.

Elkin et al. [104] introduziu um sistema intitulado por *DXplain* no fluxo de trabalho de um hospital de ensino onde o sistema apoiava os estudantes no diagnóstico de doenças. Partindo dos sintomas, a lógica *bayesian* foi aplicada para promover aos estudantes um conjunto diferencial de diagnósticos e as suas probabilidades. 1173 casos foram identificados como tal, baixando os custos de 8318\$ para 7328\$, o que representa uma poupança de 10% em encargos e de 12% em custos associados.

Sadeghi et al. [105] aplicou redes *bayesian* no desenvolvimento de um SAD no apoio aos enfermeiros no processo de triagem na unidade de urgência de um hospital. Inicialmente, vários subdomínios (ex. dores torácicas) foram descritas, paralelamente também foram identificadas com precisão informações e sintomas de doenças relevantes. Posteriormente, os autores abordaram a rede para desenvolver um modelo gráfico que representa todas as possíveis interações entre os dados históricos, gerando uma lista de possíveis diagnósticos a partir dos resultados obtidos de um questionário de triagem. O sistema de triagem comparado com o especialista obteve uma sensibilidade de 94% contra 64%, com uma especificidade de 25% contra 48% em doentes com necessidades de internamento hospitalar. Ambos corresponderam a 56% dos casos.



## 6. Recolha de dados

Esta fase tem como objetivo expor a recolha de uma vasta gama de soluções / algoritmos de otimização para modelos de apoio à decisão que tenham por base as técnicas / abordagens apresentadas na [secção 5](#).

### 6.1. Meta-heurísticas

Em qualquer método determinístico é complicado definir problemas de otimização como problemas que não se conseguem resolver otimizando, ou que sejam capazes de garantir algo dentro de um limite razoável de tempo. Estes problemas podem ser divididos em várias categorias, dependendo se são contínuos ou discretos, com ou sem restrições, mono ou multiobjectivo, estático ou dinâmico. No intuito de encontrar soluções satisfatórias para este tipo de problemas, as meta-heurísticas podem ser utilizadas.

São algoritmos concebidos para resolver por aproximação uma ampla gama de problemas de otimização sem a necessidade de adaptação profunda a cada problema.

Oriundo do prefixo grego “meta”, é utilizado para indicar que estes algoritmos são heurísticas de “alto-nível”, em contraste com heurísticas específicas em um determinado problema.

São geralmente aplicadas em problemas para os quais não existe algoritmos com resoluções satisfatórias. Utilizadas para resolver problemas complexos na indústria e em serviços, em áreas que vão da gestão e engenharia até à área financeira.

Quase todas as meta-heurísticas partilham as mesmas características, inspiradas na natureza (segundo princípios físicos, biológicos e etológicos), utilizam componentes estocásticos (que envolve variáveis aleatórias), não utilizam o gradiente ou matriz *Hessiana* na função objetivo, mas sim vários parâmetros que são necessários enquadrar para o problema em questão [106].

Nos últimos trinta anos houve um grande interesse nestas técnicas. Na Tabela 1 é apresentada historicamente a evolução das meta-heurísticas.

Tabela 2 – Abordagem histórica das meta-heurísticas

<b>Autores</b>	<b>Ano</b>	<b>Contribuição</b>	<b>Referência</b>
Kirkpatrick et al.	1982	Métodos de simulação <i>annealing</i>	[107]
Glover	1986	Pesquisa tabu	[108]
Farmer et al.	1986	Sistema imunitário artificial	[109]
Koza	1988- 1992	Programação genética	[110]
Goldberg	1989	Algoritmos genéticos	[111]
Dorigo	1992	<i>Ant colony</i>	[112]
Walker et al.	1993	<i>Bee colony</i>	[113]
Kennedy and Eberhart	1995	<i>Particle swarm</i>	[114]
Hansen and Ostermeier	1995	CMA-ES	[115]
Mühlenbein and Paaß	1996	<i>Estimation of distribution</i>	[116]
Storn and Price	1997	<i>Diferencial evolution</i>	[117]
Passino	2002	<i>Bacterial foraging</i>	[118]
Simon	2008	<i>Biogeography-based</i>	[119]
Yang	2008	<i>Firefly</i>	[120]
Yang	2009	<i>Cuckoo Search</i>	[120]
Yang	2010	<i>Bat</i>	[120]
Gandomi	2012	<i>Krill Herd</i>	[121]

As técnicas meta-heurísticas serão bem-sucedidas em um determinado problema de otimização se conseguir proporcionar um equilíbrio entre a diversificação e a intensificação. A intensificação é necessária na procura de partes no espaço com soluções de alta qualidade, e é importante na procura de algumas áreas promissoras na experiência de investigação acumulada. As principais diferenças entre as meta-heurísticas existentes dizem respeito à sua particular forma de alcançar este equilíbrio [122]. Os critérios de classificação podem

ser utilizados para as meta-heurísticas, em termos das características que seguem na pesquisa, recurso de memória, tipo de exploração vizinha utilizada ou o número de soluções atuais realizadas a partir de uma iteração para a outra.

Para uma classificação mais formal [122,123], foi efetuada uma diferenciação de meta-heurísticas entre *Single-solution based* e *Population-based*. De uma forma geral, as *single-solution based* são mais orientadas para a intensificação, enquanto as *Population-based* são orientadas para a exploração [106]. Os principais algoritmos pertencentes a estas categorizações serão abordados nas subsecções seguintes.

### **6.1.1. Single-solution based**

Nesta secção serão apresentadas meta-heurísticas baseadas em soluções únicas, também conhecidos por métodos de trajetória. Ao contrário das *Population-based*, estas começam com uma solução inicial e afastam-se dela, descrevendo uma trajetória no espaço de pesquisa. Alguns algoritmos podem ser considerados como “extensões inteligentes” de algoritmos de *local search*. Estes métodos englobam principalmente *simulated annealing*, pesquisa tabu, método *GRASP*, *variable neighborhood*, *guided local search*, *iterated local search* e as suas variantes.

#### **6.1.1.1. Simulated annealing**

Oriunda de estatística mecânica (algoritmos Metropolis [124]), proposta por Kirkpatrick et al. [107], *Simulated annealing* (SA) foi inspirada em técnicas metalúrgicas para obter um melhor estado sólido com o menor uso de energia. Esta técnica consiste na formação de um material a altas temperaturas, seguida de uma gradual redução da mesma.

O processo é transposto através desta técnica na obtenção de uma função objetivo do problema, semelhante à energia do material que é minimizada através da introdução de um parâmetro controlável, representando uma temperatura fictícia.

O algoritmo é iniciado através da geração de uma solução inicial (aleatoriamente ou construída através de uma heurística) atribuindo um valor ao parâmetro controlável. Em cada iteração, uma solução é aleatoriamente selecionada na vizinhança da última solução e é aceite dependendo do parâmetro controlável e dos valores da função objetivo [106]. Um exemplo alto-nível é apresentado no algoritmo SA (Algoritmo 1):

Algoritmo 1 – *Simulated annealing*

---

**Algoritmo - SA**

---

```

1 Escolher, aleatoriamente, uma solução inicial  $s$  para a otimização do sistema
2 Iniciar a temperatura  $T$ 
3 while o critério de paragem não é satisfeito do
4     repeat
5         Selecionar aleatoriamente  $s'$  e  $N(s)$ 
6:     if  $f(s') \leq f(s)$  then
7:          $s \leftarrow s'$ 
8:     else
9          $s \leftarrow s'$  com a probabilidade  $p(T, f(s'), f(s))$ 
10    end
11    until o equilíbrio termodinâmico do sistema é alcançado
12    Diminuir  $T$ 
13 end
14 return a melhor solução encontrada

```

---

Podemos verificar que o algoritmo converge para a solução  $s$ , mesmo que uma melhor solução  $s'$  seja encontrada durante o processo de pesquisa. Com isso, uma otimização básica de SA consiste em guardar essa melhor solução em  $s'$ .

Existem aplicações com sucesso em problemas de otimização discretos ou contínuos, embora esta tenha sido incapaz de resolver problemas combinatórios [106], pelo simples facto de não utilizar memória. Adaptações deste método em problemas contínuos de otimização foram estudados [125], e uma vasta bibliografia pode ser consultada em vários documentos científicos [126,127,128,129,130,131].

Por existirem muitas variantes de SA, de seguida serão descritas a *Microcanonic annealing*, *Threshold accepting* e *Noising*.

### A. *Microcanonic annealing*

O princípio de *Microcanonic Annealing (MA)* é muito semelhante ao de SA, diferindo-se na utilização do algoritmo *Creutz* [132], também conhecido por simulação *Monte Carlo*. Este permite alcançar o equilíbrio termodinâmico em um sistema isolado, onde a energia total é o somatório da energia potencial com a energia cinética.

Num problema de otimização, a energia potencial pode ser considerada como a função objetivo a ser minimizada e a energia cinética é usada similarmente como em SA sendo forçada a manter-se positiva. Em cada estado, o equilíbrio é atingido sempre que o rácio entre a energia cinética média observada e o desvio padrão da mesma seja perto de 1.

MA é dotado de várias vantagens quando comparado com SA, como é o caso de não necessitar de funções exponenciais para ser avaliado ou o facto de utiliza a aleatoriedade de números para aceitar ou rejeitar uma solução. No entanto, a sua computação é penalizante em questões de tempo [106].

Uma revisão em trabalhos relativos a aplicações deste método, como no processamento de imagens microscópicas, pode ser consultada na referência [133] de Nakib et al.

### **B. Método *threshold accepting***

Outra variante abordada é o método *Threshold accepting (TA)* [134]. Este método de pesquisa local começa com uma solução aleatória, que em seguida, explora geralmente de forma aleatória o espaço vizinho dessa solução. As diferenças entre SA ficam nos critérios de aceitação nas soluções candidatas. Por um lado, SA aceita novas soluções que causam deterioração na função objetivo apenas com uma determinada probabilidade, por outro lado o TA apenas aceita novas soluções se melhorarem a função objetivo [106]. Isto supera o problema de o algoritmo ficar tendencialmente preso em espaços de pesquisa locais.

Este método é comparado favoravelmente com SA em problemas combinatórios como o *travelling salesman* [135]

### **C. Método *noising***

Proposto por Charon and Hudry [136], os métodos *noising (MN)* recorrem a algoritmos *local search* que começam por desenhar uma solução inicial, melhorando-o até obterem um local ótimo através de várias iterações. Basicamente, em vez de ter em consideração diretamente os dados de um problema de otimização, este algoritmo aplica a função em cada iteração até que a amplitude do *noising* da mesma seja diminuída até zero. A razão deste comportamento está na adição de *noising* ser capaz de encontrar eventuais locais ótimos da função objetivo, sendo um valor tomado por uma variável aleatória dada a distribuição de probabilidade (ex. Lei de Gauss) [106].

Foram propostas várias formas de adicionar *noising* [137] que mostraram poder ser aplicada de forma idêntica em SA ou TA. Estes autores [138] também propuseram recentemente uma forma de este método obter os seus próprios parâmetros.

### 6.1.1.2. Pesquisa Tabu

A pesquisa *tabu* (PTB) desenvolvida para gerir algoritmos *local search* incorporados foi apresentada por Glover [108]. Este método utiliza o histórico da pesquisa de forma explícita, tanto para evitar mínimos locais como para implementar uma estratégia exploratória. É uma característica inspirada na memória humana, o que faz dela a sua principal característica, ou seja, é capaz de aprender com o passado.

Na estruturação da memória são utilizadas propriedades específicas do algoritmo na memorização da trajetória através do espaço de pesquisa. Uma lista *tabu* (a partir do qual o nome meta-heurística deriva) grava algumas das últimas soluções de atributos encontradas e nega a consulta de soluções (soluções com algum atributo em comum), tendo em conta que as mesmas já se encontram gravadas. A lista pode ser vista como uma memória volátil / dinâmica (“curta duração”) que regista as soluções visitadas recentemente e nega o acesso recursivo da mesma solução. A sua utilização impede a criação de ciclos infinitos, prevenindo a deterioração da pesquisa. O tamanho da lista também influencia o processo de pesquisa, se a lista for curta, a pesquisa irá concentrar-se em subáreas no espaço de pesquisa. Por outro lado, se a lista for longa, o espaço de pesquisa torna-se em regiões maiores porque este método proíbe visitas a soluções com algum atributo em comum. De seguida o Algoritmo 2 representa o formato geral de um PTB.

Algoritmo 2 – Pesquisa *tabu*

---

*Algoritmo - PTB*

---

```

1 Escolher, aleatoriamente, uma solução inicial  $s$  no espaço de pesquisa
2  $TabuList \leftarrow \emptyset$ 
3 while o critério de paragem não é satisfeito do
4     Selecionar a melhor solução  $s' \in N(s) \setminus TabuList$ 
          $s \leftarrow s'$ 
         Atualizar  $TabuList$ 
13 end
14 return a melhor solução encontrada

```

---

Contudo, a pesquisa pode variar de tamanho durante o seu processo, este facto levou ao desenvolvimento de algoritmos mais robustos como a pesquisa *tabu* reativa [139]. A robustez passou por introduzir estruturas de memória chamadas de *aspiration criteria*, que são um conjunto de regras utilizadas que vão para além das restrições *tabu* (se um movimento é proibido para a lista, então *aspiration criteria*, se satisfeito, permite o movimento), conseguindo guardar a pesquisa de potenciais áreas de interesse (intensificação). Outra das características é a adição de memória de “longa-duração” que incentiva uma exploração mais ampla do espaço de pesquisa. Desta forma, ao efetuar uma pesquisa e o registo de quantas vezes um determinado atributo foi encontrado nas soluções dessas pesquisas, permite evitar encontrar soluções com os atributos mais visitados ou encontrar soluções com atributos que raramente foram visitados (diversificação) [106]. Para conceitos e descrições mais extensa [140] ou possíveis métodos [141,142], deve ser consultado as respetivas referências.

A adoção de PTB na otimização de problemas contínuos por parte Chelouah e Sairry [143] teve resultados satisfatórios, e para funções com menos de 10 variáveis, obtiveram resultados semelhantes ou melhores do que outros métodos.

### 6.1.1.3. Método GRASP

*Greedy Randomized Adaptive Search Procedure* (GRASP) é uma meta-heurística sem recurso a memória para problemas de otimização combinatórios, proposto e especificado por Feo e Resende [144,145,146].

Algoritmo 3 – *Greedy Randomized Adaptive Search Procedure*

---

*Algoritmo - GRASP*

---

```

1 repeat
2     | Construir uma solução viável usando uma heurística greedy aleatória
3     | Aplicar o início da pesquisa local a partir da solução construída
4 until o critério de paragem seja satisfeito
5 return a melhor solução encontrada

```

---

Cada iteração do algoritmo GRASP (Algoritmo 3) consiste na construção e pesquisa local. O primeiro passo é formado por todos os elementos que podem ser incluídos, sem comprometerem a viabilidade da lista. Esta é ordenada em relação à função *greedy* que mede

o benefício da seleção de cada elemento. Em seguida, é escolhido aleatoriamente (por probabilidades) um elemento entre os melhores candidatos chamados de lista de candidatos restrita que pode ser limitada ao número de elementos (cardinalidade) ou pela sua qualidade (valor).

Para outro tipo de abordagem estratégica [147], uma bibliografia mais completa [148] ou propostas na adaptação em problemas de otimização contínua [149], podem ser consultados os artigos científicos apresentados por Resende et al. respectivamente.

#### ***6.1.1.4. Variable neighborhood search***

*Variable Neighborhood Search* (VNS) proposto por Hansen e Mladenovic [150,151] é uma meta-heurística que consiste na exploração e mudança de soluções vizinhas de uma determinada solução. Para iniciar, tem de ser definido um conjunto de vizinhança estruturado com cardinalidade finita, para não comprometer a pesquisa devido a grandes números de soluções encontradas, pois estes vizinhos podem ser escolhidos aleatoriamente [152]. Após isso, a solução é gerada e inicia um ciclo com três etapas: agitar, pesquisa local, e mover. Agitar consiste em selecionar uma solução inicial aleatória de um dado vizinho da solução atual, e de seguida, essa solução inicial passa para a etapa de pesquisa local para gerar uma nova solução. A pesquisa local pode usar qualquer estrutura vizinha da nova solução, que no final da pesquisa, se for melhor do que a da etapa anterior, substitui a solução inicial e o ciclo recomeça. Caso contrário, o VNS (Algoritmo 4) passa para o próximo vizinho e iniciam-se novamente as três etapas [106].



Algoritmo 4 – *Variable Neighborhood Search*

---

**Algoritmo** - VNS

---

```

1 Selecionar um conjunto de vizinhança estruturado  $Nn, n=1, \dots, nmax$ 
2 Escolher, aleatoriamente, uma solução inicial  $s$  no espaço de pesquisa
3 while o critério de paragem não é satisfeito do
4    $n \leftarrow 1$ 
5   while  $n < nmax$  do
6     Agitar: selecionar uma solução aleatória  $s'$  na vizinhança  $Nn(s)$  de  $s$ 
7     Aplicar a pesquisa local em  $s'$  para obter a solução  $s''$ 
8     if  $s''$  é melhor que  $s$  then
9        $s \leftarrow s''$ 
10     $n \leftarrow n+1$ 
11  else
12     $n \leftarrow n+1$ 
13  end
14 end
15 end
16 return a melhor solução encontrada

```

---

A eficiência deste algoritmo passa por parar quando um dado local ótimo é encontrado, e não procurar essa solução em outro local, ou seja, recorrer à vizinhança de forma complementar.

Extensões às matérias abordadas sobre este método podem ser consultadas em [153,154], ou em funcionamento híbrido com outros métodos como GRASP [155,156] ou pesquisa reativa [157].

Aloise et al. [158] utilizou este método para gerir os serviços de manutenção na produção de crude. O apoio à decisão passava por calendarizar o envio de *workovers* para a manutenção baseado em fatores como, o poço em que se encontra, a que distância se localiza e que tipo de serviços são necessários, com vista a minimizar perdas de produção. As conclusões mostraram que, numa projeção de 12 meses, acumularam-se poupanças em perdas de produção na ordem de 2,000,000\$.

Abordagens em aplicações diferentes podem também ser aprofundadas através do artigo científico de Hansen et al [159]

### **6.1.1.5. Guided local search**

*Guided Local Search* (GLS) [160,161] também recorre ao uso de memória, tal como na PTB. Neste caso a memória é intitulada por *augmented objective function*, que muda dinamicamente a função objetivo otimizada por pesquisa local, de acordo com os locais ótimos identificados. Para começar, tem de ser definido um conjunto de recursos finitos. Cada recurso define uma característica de um problema de otimização para resolver, um custo e uma penalidade associada. Como exemplo, um vendedor que necessite de viajar entre a cidade A e B, terá um conjunto de caminhos, com os seus custos associados correspondentes à distância ou tempo de viagem. As penalidades iniciam a 0 e atualizam quando a pesquisa local encontra o ponto ótimo, e penalizando as características mais desfavoráveis desse ponto, ou seja, as que tenham custos mais elevados. Com isto, a pesquisa local consegue desviar-se do pressuposto local ótimo e apresentar outras características mais atraentes [106].

Adaptações ao algoritmo de pesquisa local são constantes [162], e extensões a este método têm sido propostas por Lau [163], Zhang et al [164,165].

Aplicações como transmissões de rádio, calendarização de serviços, logísticas, otimização de funções entre outros, podem ser consultados nos seguintes artigos científicos de Voudouris et al [166,167].

### **6.1.1.6. Iterated local search**

A definição e estrutura do método *Iterated Local Search* (ILS) foram apresentadas por Stützle [168]. É uma meta-heurística baseada na seguinte forma: gera uma solução inicial para a próxima iteração ao perturbar o local ótimo encontrado na iteração atual em vez de aplicar repetidamente o procedimento de pesquisa local para aleatoriamente gerar soluções iniciais. É expectável que o mecanismo de perturbação forneça uma solução base inicial no intuito de atrair um melhor local ótimo. Este mecanismo é fundamental, onde uma perturbação muito fraca pode não ser suficiente para sair da base do local ótimo inicial e uma perturbação forte tornaria o algoritmo em um gerador de soluções iniciais aleatórias. O critério de aceitação quando combinado com o mecanismo de perturbação permite o controlo do equilíbrio entre intensificação e diversificação [106].

Como exemplo, um critério de permissão extrema, em termos de diversificação aceita qualquer solução independentemente da sua qualidade, em termos de intensificação, só aceita soluções melhoradas [169].

Para uma revisão mais aprofundada deste método em extensões e aplicações deve ser consultado o artigo de Lourenço et al. [170].

### ***6.1.2. Population-based***

Metaheurísticas *Population-based* lidam com um conjunto (população) de soluções em vez de uma solução inicial. A maioria dos estudos baseados nestes métodos estão relacionados com a Computação Evolucionária (CE) inspirada na teoria de Darwin, onde a população de indivíduos modifica-se através de recombinações, operadores genéticos e *Swarm Intelligence* (SI), onde a ideia é criar inteligência computacional ao explorar analogias simples de interações sociais, em vez de habilidades cognitivas puramente individuais [106]. Variantes destas temáticas serão abordadas nas subsecções seguintes.

#### ***6.1.2.1. Computação evolucionária***

A Computação Evolucionária (CE), inspirada na capacidade de os seres vivos evoluírem e adaptarem-se ao seu meio ambiente, baseada nos princípios de Darwin, representa o termo geral para vários algoritmos de otimização. Normalmente associado ao termo Algoritmos Evolucionários (AE) existe uma partilha na forma como a simulação da evolução estrutura os seus indivíduos, objetivos, através de processos de seleção, recombinação e reprodução de mutações, com vista a desenvolver melhores soluções [171,172].

Algoritmo 5 – Computação evolucionária

---

*Algoritmo - CE*

---

1	<i>Inicializar</i> a população com indivíduos aleatórios
2	<i>Avaliar</i> cada indivíduo
3	<b><i>repeat</i></b>
4	<i>Selecionar</i> pais
5	<i>Recombinar</i> pares de pais
6	<i>Mutar</i> os descendentes
7	<i>Avaliar</i> novos indivíduos
8	<i>Selecionar</i> indivíduos para a próxima geração
9	<b><i>until</i></b> o critério de paragem seja satisfeito

---

No Algoritmo 5, de uma forma genérica está representado um algoritmo CE que serve como exemplo para os outros métodos que serão discutidos de seguida.

Cada iteração do algoritmo corresponde a uma *geração* de uma população de soluções candidatas dado um problema de otimização. Estas soluções candidatas intituladas por *indivíduos*, são capazes de se reproduzir, estando sujeitas a variações genéticas encadeadas pela sua envolvente, causando uma seleção natural (sobrevivência). Novas soluções são criadas através de recombinações de dois ou mais indivíduos selecionados (pais) para criar um ou mais indivíduos (filhos ou descendentes). A mutação permite diversidade no aparecimento de novos traços nos descendentes. A avaliação do resultado das soluções é adequadamente avaliada através de uma estratégia de seleção onde determina quais são as soluções mantidas para a próxima geração. O critério de paragem passa por um número pré-definido de gerações (avaliações) do processo evolutivo, onde podem ser implementados outros critérios mais complexos [106].

De forma sucinta esta classe de algoritmos [173,174,175], contém:

- Representação (definição dos indivíduos);
- Função avaliativa;
- População;
- Mecanismo de seleção do pai;
- Operadores de variação, recombinação e mutação;
- Mecanismos de sobrevivência (substituição).

De seguida, é efetuada uma abordagem mais detalhada sobre os métodos que partilham estas características e as suas influências.

### **A. Algoritmos genéticos**

Originalmente desenvolvido nos finais de 1980 por Goldberd no estudo de sistemas adaptativos [111], os algoritmos genéticos (AG) têm vários aspetos que podem ser implementados de diferentes formas de acordo com o problema em causa: na representação da solução (cromossomas), estratégia de seleção, tipos de cruzamento (operadores de recombinação) e operadores de mutação. A representação mais comum dos cromossomas é numa *string* binária com tamanho finito. Operações de manipulação em *bits* permitem as operações de cruzamento e mutação. Estes operadores são a parte essencial dos AG, enfatizam o cruzamento como principal operador de variação, que combina normalmente dois (ou mais) indivíduos selecionados em conjunto na troca de algumas das suas partes, suportadas por um parâmetro exógeno (taxa de cruzamento) que indica a probabilidade de submissão no cruzamento por indivíduo (tipicamente entre 0.6 e 1) [176]. Para os indivíduos produzirem descendentes, estes são avaliados na sua aptidão com vista aos critérios de seleção e é usada uma estratégia de seleção (ex. roleta, torneio, classificação). A comparação dos critérios de seleção nos AG são apresentados por Golberg e Keb [177] e por Blickle e Thiele [178].

Após o cruzamento, os indivíduos são submetidos a mutações que introduzem alguma aleatoriedade na pesquisa para que o processo de otimização evite ficar viciado em locais ótimos encontrados. Normalmente consideradas como um operador genético secundário, está encarregue de aplicar ligeiras perturbações para soluções com baixa probabilidade de submissão à mutação, que tipicamente é aplicada com menos de 1% de probabilidade, mas sem nenhum estudo aprofundado na sua adequação. A seleção do sobrevivente com vista à sua aptidão serve para identificar e manter os indivíduos como pais para as gerações futuras e tem a responsabilidade de assegurar a sobrevivência dos indivíduos mais aptos [179]

Neste contexto existem muitas variantes que foram desenvolvidas e aplicadas em vários tipos de otimizações problemáticas. Podem ser encontradas e revisitadas referências com uma visão geral [180,181], formas híbridas [182] e otimizações multiobjectivo [183] com AG.

## B. Estratégia evolutiva

A Estratégia evolutiva (EE) imita os princípios da evolução natural como um método para resolver problemas de otimização, tal como foi apresentado anteriormente nos AG. Introduzida por Rechenberg [184,185] e desenvolvida por Schwefel [186], o primeiro algoritmo de EE foi usado para otimizar parâmetros experimentais. Contudo este baseia-se numa população formada por um único progenitor que por meio de mutação Gaussiana produz um único descendente. O critério de seleção determina a aptidão do indivíduo no intuito de se tornar o progenitor da próxima geração.

Rechenberg propôs a EE *multimembered*, introduzindo o conceito de população, que onde mais do que um progenitor pode em conjunto gerar um único descendente. Com isto, é possível ter operações de recombinação adicionais, quando dois pais escolhidos ao acaso recombina para originar um filho, sujeito a mutações. O processo de seleção agora tem em consideração a extinção do pior, que tanto pode ser um dos pais como o filho, de forma a manter o tamanho da população constante. A mutação é realizada através de números distribuídos com média e desvio padrão zero (determina o tamanho da mutação), sendo fácil entender que os parâmetros da distribuição comprometem o desempenho do algoritmo de pesquisa. A forma mais simples de especificar a mutação é manter o seu mecanismo constante ao longo do tempo.

Existem várias aproximações a este método, como por exemplo, um método introduzido recentemente por Hansen et al. [115] intitulado de *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES), mostrou ser muito eficaz e atualmente é o mais utilizado na gama dos algoritmos evolutivos para otimizações locais [187] como também para otimizações globais [188]. Este algoritmo tem obtido, entre todas as técnicas avaliadas, os melhores resultados [189].

## C. *Evolutionary programming*

*Evolutionary programming* (EP) foi apresentada como uma abordagem evolutiva de *Artificial Intelligence* [190] por Fogel, e reintroduzida mais tarde pelo mesmo para resolver outras tarefas, tal como problemas de previsão, otimização e SAA [191,192].

Normalmente as suas representações são adaptadas ao domínio do problema. A população é escolhida aleatoriamente em relação a uma função de densidade que incrementa com o

respetivo objetivo. Ao contrário do AG, normalmente o EP não depende de qualquer tipo de recombinação, sendo a mutação o único operador para gerar novos descendentes criados na adição aleatória em certas distribuições em relação ao pai (mecanismo de seleção determinístico). Já o critério de substituição é probabilístico (baseado em um torneio de seleções estocásticas).

#### **D. Programação genética**

Com o desenvolvimento por parte de Koza [110], a Programação genética (PG) tornou-se uma técnica de pesquisa popular. É um método automatizado na criação de um programa que trabalha a partir de problemas de alto nível, baseado nos princípios de Darwin de seleção natural e operações com inspirações biológicas. Adota uma estratégia parecida com a de AG mas com uma diferença na representação e no facto de ser dotada de operadores especiais. Aqui, os membros da população não estão distribuídos por *strings* de tamanho fixo (como nos AG), são programas que quando executados, tornam-se as soluções candidatas para um determinado problema. Expressos em árvores sintáticas, em vez de linhas de código, propõem uma forma perceptível de descrever esses programas na linguagem *LISP* (*List processing*), trata-se de uma linguagem dinâmica que permite alterações em tempo real, sem a necessidade de recompilações e principalmente é funcional, no sentido que todo o trabalho pode ser realizado através da composição de funções e recursões [193]. Os terminais (variáveis e constantes) são folhas da árvore, enquanto as operações (funções) são nós internos. A junção dos terminais com as funções forma o alfabeto dos programas (indivíduos) a serem definidos.

A população inicial começa na geração aleatória dos programas compostos por funções e terminais apropriados ao problema. Para tal existem duas formas básicas para o conseguir, através dos métodos *full* e *grow*. O método *full* é responsável por criar árvores para que o comprimento de cada caminho sem retorno entre a raiz e as extremidades seja igual à profundidade máxima especificada (definição de profundidade). O método *grow* encarrega-se de tornar as árvores maiores de uma forma variável (nunca sendo maior do que a profundidade máxima especificada).

Na criação da próxima população de indivíduos, os programas de computador são selecionados através de métodos probabilísticos (tendo em conta o seu valor) dos programas atuais. Com isso, os melhores indivíduos são mais propensos a terem mais filhos que

indivíduos com menor valor (através de seleção torneio), qualquer mecanismo *standard* de seleção pode ser usado [194]. A recombinação é implementada com o cruzamento entre dois pais (os filhos são subárvores dos pais que podem ter tamanhos e posições diferentes). Outra característica relevante é a mutação (mutação de subárvores), que seleciona aleatoriamente um ponto em uma árvore e substitui a subárvore desde a raiz por uma subárvore gerada aleatoriamente. A fase de substituição aplica-se à população dos pais e filhos, através de duas alternativas de implementação: a abordagem de geração, onde a população dos filhos substitui sistematicamente a dos pais, e a abordagem de estado estacionário, onde a população pai é mantida e alguns dos seus filhos são substituídos de acordo com algumas regras. Os avanços nestes métodos apresentados por Koza estão focados na automatização de funções e operações especializadas, como a permutação, edição e encapsulamento [195]. Hoje em dia a PG está presente em tarefas de previsão em SAA e DM, onde pode ser consultada uma análise mais aprofundada efetuada por Shan et al. [196].

Para uma análise de casos práticos onde este método foi aplicado, é pertinente analisar os artigos científicos de Poli et al. [194], William et al. [197] e McKay et al. [198].

### **6.1.2.2. Outros algoritmos evolutivos**

Existem outros modelos de algoritmos evolutivos que têm sido propostos e adaptados ao longo do tempo. Entre muitos, os *Learning classifier systems*, algoritmos de estimação de distribuição, algoritmos de coevolução, *cultural algorithms*, *scatter search and path relinking* e *differential evolution* serão brevemente abordados nas subseções seguintes.

#### **A. Learning Classifier Systems**

Apresentados em 1975 por Holland, os *Learning Classifier Systems* (LCS) [199] têm desenvolvido inúmeras investigações na sua arquitetura e performance. Um LCS aprende um conjunto de regras (com significados) através de interações com o ambiente, que normalmente são executadas por esquemas de *reinforcement learning* [200]. Sempre que um *input* é reconhecido, há uma competição entre as regras para que sejam desencadeadas ações na obtenção de retorno por parte do ambiente. O retorno é utilizado pelo sistema na atualização da qualidade das regras. A componente de exploração, normalmente constituída por AG é eventualmente acionado para melhorar esse conjunto de regras (*best action map*



ou *complete action map*). Com isto foram desenvolvidas duas grandes diferenças na sua arquitetura, sistemas de classificação *strength-based* e *accuracy-based* [201].

*Accuray-based* LCS baseia-se essencialmente na precisão da previsão do que na própria previsão. Isto quer dizer que o AG independentemente de a previsão ser alta ou baixa, a sua pesquisa incidirá em regras que tornem mais precisas as previsões. Na representação do conhecimento, estes sistemas adotam um *complete action map* (CAM), que consiste em um conjunto de regras precisas associadas aos diferentes níveis de recompensa definidos pelo ambiente. Os *Strength-based* LCS baseiam-se na força da interação com o ambiente, isto é, o AG procura regras onde a estimativa de recompensa que recebe a partir do ambiente seja a melhor. Recorrem ao *best action map* (BAM) (mapa que contém apenas regras com altas recompensas) para a representação do seu conhecimento.

Para uma comparação entre qual a melhor representação do conhecimento, vão ser analisados alguns aspetos como a exploração, tamanho da população, complexidade, generalização e mudança no ambiente, onde podem diferir [201].

Na exploração, os CAM necessitam de uma maior exploração do espaço de pesquisa, tanto em regras corretas como incorretas desde que sejam consistentes, enquanto os BAM apenas pesquisam as regras incorretas consistentes. No caso de grandes regiões exploratórias em problemas de classificação os CAM tornam-se mapas não remunerados (não obtêm as recompensas esperadas). No entanto, Kovacs (2000) defendeu a permanência de regras incorretas consistentes no CAM, pois ajudam no processo de exploração pelo facto de, ao codificar e lembrar-se destas regras, o algoritmo evita explorá-las, além disso estas regras ajudam a dizer ao sistema o que não deve fazer.

O tamanho de um CAM é maior do que um BAM, logo consequentemente, necessita de mais recursos computacionais.

Complexidade diz-nos o tamanho ótimo da população. Quando maior a população ótima, maior o número de ciclos necessários para a aprendizagem. Então, a evolução dos CAM pode durar mais que os BAM.

Generalização pode ser equivocadamente associada à fraca evolução dos BAM, pelo facto de alguns CAM conterem BAM e mesmo assim não têm essa aptidão. Esta falha de generalização deve estar relacionada com o sistema e não com a forma de representar o conhecimento. Contudo, os LCS são capazes de generalizar sobre estados onde o retorno seja máximo, numa abordagem BAM [201].

Mais recentemente, Santos et al. [40,159,181,300] aplicaram esta técnica para otimizar sistemas *Distributed Data Mining* (DDM). Fundamentalmente estes trabalhos permitiram estruturar uma *framework* de forma a provar que ambientes distribuídos permitem gerir espaços locais independentes, transmitindo modelos de aprendizagem para o modelo global, criando um conhecimento completo do problema.

Examinaram também a capacidade de modelos DDM analisarem dados homogéneos e heterogéneos, com conjuntos de dados sintéticos e reais, demonstrando premilinarmente, performances competitivas quando comparados com o método convencional *Centralized Data Mining* (CDM).

## **B. Algoritmos de estimação de distribuição**

Os algoritmos de estimação de distribuição (AED) foram introduzidos por Mühlenbein and Paaß [116]. Baseados em modelos probabilísticos, estes procedem a sua execução em duas etapas: estimar a distribuição de probabilidade nas soluções promissoras e gerar novas populações em amostras de distribuição de probabilidades. Estas etapas levam à pesquisa de espaços para soluções promissoras. Estas soluções são incluídas na população inicial, substituindo-a na íntegra ou parcialmente até que o processo cumpra os critérios de paragem. Os modelos probabilísticos e os métodos utilizados para aprendizagem pelos AED podem variar conforme as características do problema de otimização em mãos, tanto na ótica da complexidade como da performance [106].

AED podem ser divididos em três classes (univariada, bivariada, multivariada), dependendo da complexidade dos modelos utilizados para entender as interdependências entre variáveis. Univariadas começam com métodos que assumem independência entre variáveis [202,203], bivariadas têm em consideração alguns pares de interação [204,205,206] e para métodos que conseguem com precisão modelar estruturas complexas (blocos de variáveis sobrepostas) está a classe multivariada [207,208,209,210,211,212,213].

Estes algoritmos têm sido aplicados em vários domínios como na robótica, biomédica e engenharia. Estudos e detalhes das diferentes abordagens nos vários domínios (discretos e contínuos), como vantagens e desvantagens, podem ser consultados em [207,214].

### C. Algoritmos de Coevolução

Há organismos que são ecologicamente íntimos (ex. predadores e presas, insetos e flores) a sua evolução depende um do outro. A coevolução biológica espelha-se em muitos processos naturais que inspiraram a criação de algoritmos de coevolução (AC) nos inícios da década de 90, introduzidos por Hillis [215]. Contrariamente aos métodos convencionais de AE, onde os indivíduos são avaliados independentemente, estes avaliam duas ou mais populações de indivíduos, com adaptações recíprocas, estão constantemente a coevoluir em simultâneo.

Desde a sua introdução, surgiram muitas variantes deste método, que se podem dividir em duas categorias: coevolução competitiva e cooperativa. Na abordagem competitiva, as populações diferentes (soluções) competem para a resolução de problemas globais (o sucesso de uma depende do fracasso da outra) e são recompensadas conforme a sua interação [216]. A coevolução competitiva desde então tem sido aplicada em jogos estratégicos [217,218], na evolução do reconhecimento de padrões [219] e na coevolução do comportamento de agentes complexos [220]. Já na abordagem cooperativa, as populações são convidadas a cooperar na resolução do problema, onde os indivíduos coletivamente são recompensados quando funcionam bem e penalizados quando algo corre mal. É uma inspiração oriunda da simbiose ecológica, introduzida por Potter e De Jong [221] em 1994. Esta contribuição propôs evoluções de soluções em forma de subcomponentes coadaptados (decomposição de problemas complexos em subproblemas). Em cada subcomponente é atribuída uma população que está encarregue de a tornar parte de uma solução maior. A sua evolução ocorre em simultâneo com outros subcomponentes, de uma forma isolada, interagindo para obter ajustes. Este processo pode ser estático, onde os subcomponentes são separados e nunca alterados, ou dinâmico, onde as populações desses subcomponentes podem ser adicionadas ou removidas a qualquer momento [222]. Estas abordagens têm obtido sucesso em relação a outras abordagens concorrentes, tal como a RNA [223], anteriormente abordada. Variantes desta abordagem, como o *co-evolutionary particle swarm optimization* [224] entre outras, têm sido propostas.

Uma combinação entre estas duas abordagens foi interpretada por Goh et al. [225] para resolver problemas em ambiente dinâmicos [106].

### D. *Cultural algorithms*

O termo cultura foi introduzido por Edward B. Taylor no seu livro [226] onde o define como “o complexo como um todo inclui conhecimentos, crenças, arte, moral, costumes, e quaisquer outras capacidades adquiridas pelo homem como um membro da sociedade”. Assim, os *Cultural algorithms* (CA) são apresentados como sendo uma classe de modelos computacionais que derivam da observação do processo e da evolução cultural na natureza [227].

Com este ponto de vista, a evolução é caracterizada pela frequente mudança de genes nas populações ao longo do tempo com o processo de seleção natural, podendo referir que a evolução cultural origina-se com trocas distributivas de atributos culturais das populações. Seguindo esta ideia, Reynolds desenvolveu um modelo em que a evolução cultural é vista como um processo de heranças, operando a nível microevolutivo (na transmissão de material genético entre indivíduos de uma população) e a nível macroevolutivo (conhecimento adquirido através de experiências individuais), onde o fundamental é a noção que o indivíduo tem perante o mundo num mapa cognitivo intitulado por *mappa*, que se baseia em experiências com o exterior, moldando as suas interações. Um *mappa* individual pode ser cortado ou modificado para formar um grupo de *mappa*, com vista a orientas futuras ações do grupo e dos seus indivíduos [106].

Os CA compõem-se em três partes. A exploração do espaço microevolutiva mantém um conjunto de indivíduos para serem evoluídos e mecanismos para avaliá-los, reproduzi-los e modifica-los. A nível macroevolutivo, a exploração do espaço representa o conhecimento adquirido pela população durante a evolução, sendo necessário preservar crenças socialmente aceites (ex. situacionais, normativas, geográficas, históricas e conhecimento no domínio) e descartar as outras. A última parte, protocolo de comunicação, é usada para determinar a interação entre a população e as crenças [106].

Com isto, pode-se concluir que os CA são baseado em sistemas evolutivos híbridos que integram a pesquisa evolucionária e raciocínio simbólico [228], mostrando serem úteis para problemas que requerem um grande conhecimento [229] e em ambientes dinâmicos [230]. O seu desempenho foi avaliado em problemas de otimização [231], e nas mais diversas áreas. Como exemplo este conceito foi aplicado à agricultura [232], calendarização de trabalhos [233], reengenharia [234], problemas de otimização combinatórios [235], multiobjectivo [236] e à modelação baseada em agentes [237].

### ***E. Scatter search and path relinking***

Desenvolvido por F. Glover [238] na década de 90, *Scatter Search and path relinking* (SSPR) opera sobre um conjunto de soluções (*RefSet*), tipicamente pequeno, que ao combiná-las cria novas soluções. O algoritmo começa na criação do conjunto de vetores solução que satisfaçam os critérios da diversidade, selecionando subconjuntos dos melhores vetores para serem soluções de referência. Os melhores vetores não estão limitados a qualquer medida determinativa, mas sim na abrangência e na diversidade de soluções, tanto que é possível adicionar uma solução ao conjunto de referência se esta adicionar valor, mesmo que em relação a soluções concorrentes seja de inferior relevância. Com isto surgem soluções teste onde é aplicado um processo de melhoria, onde o conjunto de referência e a população de soluções são atualizadas na ótica da qualidade e diversidade. Este processo de melhoria é iterativo até o conjunto de referência não sofrer qualquer alteração [106].

Os principais passos consultados em [239] deste algoritmo são:

- *SeedGeneration()*;
- *DiversificationGenerator()*;
- *Improvement()*;
- *ReferenceSetUpdate()*;
- *SubsetGeneration()*;
- *SolutionCombination()*.

SSPR tem uma vasta variedade de aplicações, onde Glover et al. [240] apresentam as suas panorâmicas.

### ***F. Differential evolution***

Um dos algoritmos mais populares para problemas de otimização contínuos é o *Differential evolution* (DE). Proposto por Storn e Price [241] com o propósito de resolver um problema de ajuste polinomial, provando ser uma estratégia de otimização muito fiável para outro tipo de tarefas.

Como qualquer AE, uma população de soluções candidata é aleatoriamente selecionada para uma determinada tarefa de otimização. Em cada processo de geração evolutivo, novos indivíduos são criados através da aplicação de operadores (cruzamento e mutação). A aptidão das soluções resultantes são avaliadas por cada indivíduo da população contra um novo indivíduo (mutante), onde este é criado através da recombinação do indivíduo da

população com outro indivíduo criado por mutação, com o intuito de determinar qual deles será mantido para a próxima geração [106].

A principal vantagem dos DE é pelo facto de terem menos parâmetros de controlo (apenas três de entrada), que controlam o processo de pesquisa (tamanho da população, diferenciação e cruzamento). Por consequência estes parâmetros são fixos, tornando-se não trivial definir prioridades nos parâmetros mediante um determinado problema. Com isso, alguns autores desenvolveram estratégias no ajuste de parâmetros de acordo com a experiência na aprendizagem [106].

DE hoje em dia é uma das heurísticas mais populares na resolução de problemas de otimização *single-objective* em espaços de pesquisa contínuos, no qual a sua utilização foi expandida para problemas *multi-objective* [242]. Contudo, existem falhas na lenta convergência e estagnação da população. Para mais variantes, detalhes e aplicações devem consultar os artigos de Chakraborty [243], Neri e Tirronen [244] e de Das e Suganthan [245].

### **6.1.2.3. *Swarm intelligence***

*Swarm Intelligence* (SI) é um paradigma de inteligência distribuída e inovador na resolução de problemas de otimização inspirados em comportamentos coletivos de vários insetos e animais. São tipicamente constituídos por uma população de agentes (capazes de realizar e executar certas operações) que interagem entre eles e com o ambiente envolvente. Não havendo uma estrutura de controlo individual, as interações locais entre esses agentes levam ao aparecimento de comportamentos globais auto-organizados [106], denominando-se por *Stigmergy*. Este pode ser entendido como uma aprendizagem intra-gerações, permutável, simplista e está dependente dos indivíduos da população e da sua comunicação indireta quando focados em um objetivo comum.

Muitos algoritmos de otimização como o *Ant colony optimization*, *Particle Swarm Optimization*, *Bacterial foraging optimization*, *Bee Colony Optimization*, *Artificial Immune Systems* e *Biogeography-Based Optimization* são inspirados nas metáforas deste comportamento [246].

As subseções seguintes analisam de uma forma geral estes algoritmos.

## A. *Ant colony optimization*

*Ant colony optimization* (ACO) é uma meta-heurística inspirada no comportamento das formigas reais na procura de alimento para a resolução de problemas de otimização combinatórios introduzida por M. Dorigo [247,248,249]. Ao procurar comida as formigas começam por analisar a área ao redor do seu ninho, e ao longo da trajetória liberta um rasto com feromonas químicas no solo, com o intuito de marcar um caminho favorável para orientar outras formigas para o alimento encontrado [250]. Após isso, o caminho mais curto entre o ninho é marcado com uma maior concentração de feromonas que conseqüentemente atrai mais formigas. Com isto, espera-se explorar as características das colônias de formigas para construir soluções com a troca de informação sobre a qualidade e o esquema de comunicação para problemas de otimização [251].

Após a inicialização dos parâmetros e caminhos com feromonas o algoritmo passa por três fases:

- *ConstructionAntSolutions*;
- *DaemonActions* (opcional);
- *UpdatePheromones*.

No início do algoritmo *constructionAntSolutions* são ajustados todos os parâmetros e as variáveis das feromonas são iniciadas com um certo valor, que é um parâmetro do algoritmo. Esta fase passa por construir um conjunto artificial de formigas de forma incremental e estocástica na proposição de soluções para o problema em consideração a partir de uma solução inicial vazia. Em cada construção, a solução atual parcial é aumentada pela adição de um componente de solução viável, tendo esta decisão a influência na quantidade de feromonas associadas em cada elemento da solução parcial atual e informação heurística sobre o problema. Esta informação atribui a cada componente da solução viável um valor, que é usado pelas formigas para tomarem decisões em como moverem-se [252].

Na segunda fase, *DaemonActions*, as ações e operações centralizadas que não podem ser tomadas apenas por uma formiga, que consiste na aplicação da pesquisa local nas soluções construídas.

Por último, no *UpdatePheromones* está o processo de atualização das feromonas, que tem como objetivo tornar os componentes das soluções a serem parte de boas soluções, composta por dois mecanismos: evaporação das feromonas que é aplicada enquanto são construídas soluções, responsável por intensificar os melhores caminhos ao longo do tempo e deixar

dissipar os maus caminhos, favorecendo a exploração de áreas que ainda não foram visitadas. Este depósito é aplicado após todas as formigas terem terminado a construção de uma solução e o seu valor está associado à qualidade da componente de solução. O objetivo é tornar estas componentes da solução mais atraentes para as formigas nas iterações seguintes [106].



Algoritmo 6 – *Ant Colony Optimization*

---

**Algoritmo - ACO**

---

```

1 Definir parâmetros
2 Inicializar libertação de feromonas
3 Inicializar matriz heurística
4 Inicializar conjunto Pareto como vazio
5 while critério de paragem não satisfeito do
6     | ConstructAntSolution()
7     | ApplyLocalSearch() (opcional)
8     | UpdateParetoSet()
9     | UpdateGlobalPheromone()
10 return conjunto Pareto

```

---

Foram propostos diferentes algoritmos ACO, mas todos partilhavam as mesmas características. A sua discussão, pesquisa e aplicações podem ser consultadas em [250,251,252,253,254] onde os seus autores relacionam ACO com outras variantes. Já mais recentemente Angus e Woodward [255] defendem que estes algoritmos irão ser uma grande vantagem, e tornar-se-ão comuns no quotidiano, quando forem aplicados de forma sistemática em aplicações do mundo real com dados variáveis no tempo e disponibilidade de informação.

## **B. Otimização por enxame de partículas**

Apresentada em 1995 por Kennedy e Eberhart [114], a Otimização por enxame de partículas (OEP) intitula-se como uma técnica de otimização global que recorre a metáforas do comportamento em grupos das aves quando voam na resolução de problemas de otimização. Existem algumas diferenças entre OEP e otimização evolutiva abordada em [256] são expostas e debatidas essas diferenças.

Algoritmo 7 – Otimização por enxame de partículas

---

*Algoritmo – Otimização por enxame de partículas*

---

```

1 Função objetivo
2 Inicializar localização e velocidade das partículas
3 Encontrar o global best atual
3 while (critério de paragem não satisfeito)
4     for sobre todas as partículas e dimensões
5         Gerar nova velocidade
6         Calcular novas posições
7         Avaliar funções objetivo em todas as novas posições
8         Encontrar local best para cada partícula
10    end for
11    Encontrar a global best atual
15    Atualizar o contador de iterações
16 end while
17 return a localização final e global best

```

---

Neste algoritmo, entidades autónomas (partículas) estão aleatoriamente a gerar eventos no espaço de pesquisa, onde cada entidade é candidata à solução para o problema em questão. Este processo é representado por uma velocidade, localização no espaço e tem memória que apoia a lembrar-se da sua melhor posição [106].

Um enxame é formado por um determinado número de partículas em redor de um certo espaço dimensional de pesquisa, onde há algum tipo de topologia que descreve a interligações entre as partículas. Estas interligações em relação a uma partícula formam os chamados bairros que podem ser toda a população ou um subconjunto da mesma. As topologias mais comuns na identificação de partículas são a *gbest* e a *lbest* [106].

A topologia *gbest* identifica o melhor vizinho (influência) de toda a população para uma determinada partícula e normalmente a sua visualização forma um gráfico totalmente ligado. A *lbest* é uma estrutura simples em anel onde cada partícula está ligada a outras duas adjacentes na população. Kennedy et al. [257] concluiu que esta topologia tendencialmente converge para a probabilidade de ficar preso em locais ótimos, no entanto, esta topologia é mais lenta mas explora mais profundamente e normalmente termina no melhor ponto ótimo. Tem sido implementado um grande esforço na compreensão do funcionamento do algoritmo OEP na análise das trajetórias [258,259,260] e no porquê de falhar sobre certas condições.

A formulação de OEP em espaços discretos e contínuos [261], ambiente dinâmicos [262] e implementação paralela [263] também têm sido consideradas. Já outras adaptações a este tipo de otimização como na diversidade, mutação, cruzamento e inicialização diferentes também foram discutidas por Pant et al. [264].

### C. *Bacterial foraging optimization algorithm*

Introduzido por Passino em 2002 [118], *Bacterial foraging optimization algorithm* (BFOA), é um novo paradigma inspirado no comportamento social da bactéria *Escherichia coli* (*E.coli*) na procura de alimento que estão presentes nos intestinos humanos. Em muitos organismos, a procura de alimento é uma atividade crítica na sobrevivência que envolve estes na criação de grupos, para tentar de maximizar a energia obtida e ao mesmo tempo minimizar o tempo gasto na obtenção de uma fonte de alimento e na exposição a predadores [265]. A procura de alimentos em grupo é um elemento primordial nesta minimização de exposição a predadores, que por sua vez aumenta a probabilidade de encontrar alimento. Este algoritmo funciona baseado em conceitos biológicos (quimiotaxia, *swarming*, reprodução, eliminação-dispersão), que podem ser consultados detalhadamente na seguinte referência [265].

- Quimiotaxia: é o processo onde a bactéria direciona os seus movimentos de acordo com certas substâncias presentes no seu ambiente. Através de proteínas recetoras detetam nutrientes e substâncias nocivas. No entanto pequenas mudanças na concentração de nutrientes podem causar uma mudança significativa no seu comportamento.
- *Swarming*: é o tipo de movimentos espontâneos que são promovidos por flagelos e permite que as bactérias se movam rapidamente em superfícies de ambientes viscosos. Normalmente organizam-se automaticamente em anéis.
- Reprodução: após a quimiotaxia (passos que compreendem o movimento e a determinação do custo da bactéria em cada posição) as bactérias entram na fase da reprodução. Para reproduzirem, as bactérias menos saudáveis morrem (que não conseguiram encontrar nutrientes) e são repostas pelo mesmo número em bactérias saudáveis através da bactéria mais saudável (que se divide assexuadamente em duas bactérias no mesmo local).
- Eliminação-dispersão: o ambiente pode influenciar o comportamento das bactérias. Quando uma mudança ocorre (ex. predadores, variação da temperatura), todas as bactérias em uma região podem dispersar para outra região ou até mesmo morrer.

Qualquer uma destas situações destrói a quimiotaxia existente. Contudo, a dispersão pode ser um bom acontecimento, pode movê-las para uma região rica em nutrientes. Nesta fase há um certo número de eventos onde cada bactéria de uma população é submetida

Contudo, alguns autores [266] discutiram experiências do BFOA em problemas de otimização complexos que revelaram ter pouca performance e fraco crescimento no espaço de pesquisa, em comparação com outros algoritmos como os AG e OEP.

#### D. *Bee colony optimization-based algorithm*

Algoritmos *Bee colony optimization-based* (BCOB) são uma nova geração de algoritmos inspirados no comportamento das colônias de abelhas. Apresentam recursos que podem ser utilizados como modelos para SI e comportamentos coletivos como *waggle dance* (comunicação), procura de alimento, abelha rainha, seleção de tarefa, tomada de decisão coletiva, seleção do local do ninho acasalamento em voo, casamento entre colônias, floração e sistemas de navegação [267]. Com isto, vários algoritmos baseados nestes comportamentos foram propostos de forma a replicar o seu conhecimento, tal como o *Bee Colony optimization metaheuristic* (BCO) [268], *Artificial bee colony* (ABC) [269] e o *Virtual bee algorithm* (VBA) [270]. Dando relevância ao ABC (Algoritmo 8), a seguir é representado os seus principais passos:

Algoritmo 8 – *Artificial Bee Colony*

---

<i>Algoritmo - ABC</i>	
1	<i>Inicializar</i> a população
2	<b><i>repeat</i></b>
3	<i>Colocar</i> as abelhas empregadas nas fontes de alimentos e determinar a sua quantidade
4	<i>Calcular</i> a probabilidade da preferência abelha observadora
5	<i>Colocar</i> as observadoras nas fontes de alimento
6	<i>Parar</i> a fase de exploração das abelhas
7	<i>Enviar</i> guias aleatoriamente para novos locais em busca de novas fontes
7	<i>Memorizar</i> a melhor fonte de alimento até ao momento
8	<b><i>until</i></b> o critério de paragem seja satisfeito

---

Alguns algoritmos foram desenvolvidos inspirados no comportamento cooperativo das abelhas tal como o *Honey bee teamwork strategy* [271], entre outros [272,273].

Uma revisão de literatura em algoritmos inspirados no comportamento das abelhas na natureza e suas aplicações podem ser consultadas no artigo de Karaboga e Akay [274].

### **E. *Artificial immune systems***

O nosso sistema imunológico é uma rede de células, tecidos e órgãos que trabalham em conjunto para defender o organismo de ataques (bactérias e vírus) sem qualquer tipo de estrutura definida. É dotado de poderosas capacidades de aprendizagem e memória que apresentam repostas evolutivas face a elementos perigosos [275]. Os *Artificial immune systems* (AIS), melhor entendido por Dasgupta et al. [276], tentam aplicar este princípio na otimização de problemas de SAA [277].

Nesta tentativa, foram desenvolvidos quatro grandes algoritmos AIS que têm ganho popularidade: *negative selection algorithms*, *artificial immune networks*, *clonal selection algorithms* e *danger theory and dendritic cell algorithms*. Uma análise mais detalhada pode ser encontrada nas referências anteriormente expostas [276,277,278].

### **F. *Biogeography-based optimization***

Desenvolvido por Dan Simon em 2008 [279], o algoritmo *Biogeography-based optimization* (BBO) foi influenciado pelo equilíbrio biogeográfico de ilhas [280], que trata da variação de equilíbrio entre imigração de novas espécies e a emigração de espécies já instaladas. Cada ilha representa um conjunto de soluções candidatas, com um determinado *suitable index variable* (SIV) e outro avaliador intitulado por *habitat suitability index* (HSI) é utilizado para medir a eficiência e eficácia da solução. Neste algoritmo, cada indivíduo tem a sua própria taxa de imigração e emigração, e as boas soluções (ilhas com muitas espécies) tendem a partilhar os seus recursos com soluções fracas (ilhas com poucas espécies). As soluções pobres estão recetivas a novas espécies de boas soluções [280].

Existem outros fatores importantes que influenciam as taxas de migração entre habitats, como a distância para o habitat vizinho, o seu tamanho, clima (precipitação e temperatura), vegetação, diversidade animal e atividade humana que não foram bem considerados. Com isso, Haiping Ma [281] explorou seis tipos de migração diferentes, e testou o seu desempenho com vastas gamas e dimensões através de 23 funções *benchmark*. Os resultados mostraram mudanças positivas significativas no seu desempenho em comparação com modelos lineares na maioria dos *benchmarks*.

Algoritmo 9 – *Biogeography-based optimization*

---

*Algoritmo - BBO*

---

```

1 Inicializar um conjunto de soluções (habitats) para o problema
2 while o critério de paragem não é satisfeito do
3     Avaliar a aptidão (HSI) para cada solução
4     Processar o número de espécies, taxa de imigração e emigração
5     Modificar habitats baseados nas taxas
6     Mutação baseada em probabilidades
7     Implementar elitismo para manter as melhores soluções na
      população para a próxima geração
8 end

```

---

O algoritmo BBO (Algoritmo 9) tem sido aplicado em problemas de otimização reais, como na seleção de sensores, classificação de imagens satélite e otimização de redes elétricas e mostrou desempenhar bom sucesso várias funções *benchmark* [106].

### G. *Bat algorithm*

Os únicos mamíferos com asas existentes no nosso planeta são os morcegos, podem parecer comuns, mas existem pelo menos 1000 espécies diferentes que representam mais de 20% dos mamíferos existentes. *Bat algorithm* (BA), desenvolvido por Xin-She Yang em 2010 [282], representa o comportamento de uma espécie em particular, *Microchiroptera*, que emite impulsos sonoros e escuta o eco do ambiente que se encontra, denominado por ecolocalização. Utilizando pequenos impulsos em ultrassom, consegue aperceber-se da distância e orientação do alvo, tipo de presa e a sua deslocação no escuro. A natureza deste algoritmo traz muitas vantagens, como por exemplo, pode fornecer uma convergência rápida passando de diversificação para a intensificação [120].

Algoritmo 10 – *Bat algorithm*

---

*Algoritmo - BA*

---

```

1 Função objetivo
2 Inicializar a população de morcegos
3 Definir frequência dos impulsos
4 Inicializar ruído e taxa de impulsos
5 while (critério de paragem não satisfeito)
6     Gerar novas soluções ajustando a frequência e atualizando
       velocidades e localizações
7     if (critério de paragem não satisfeito)
8         Selecionar uma solução das melhores soluções
9         Gerar solução local em torno da melhor selecionada
10        end if
11        Gerar uma nova solução voando aleatoriamente
12        if (critério de paragem não satisfeito)
13            Aceitar as novas soluções
14            Aumentar taxas de impulsos e reduzir ruído
15        end if
16        Classificar os morcegos e encontrar o melhor atual
17 end while
18 return visualizações e resultados

```

---

Resumindo as vantagens, há alguns pontos-chave que se salientam, como o ajuste de frequência, *zoom* automático e controlo de parâmetros. Alterando os valores destas características, muitos outros métodos e estratégias têm sido tentadas com o objetivo de aumentar a diversidade da solução e para melhorar o desempenho do algoritmo. Com isso, pelo menos nove variantes para explorar estas diferenças foram criadas. Concluindo esta relevância ao longo dos anos, BA é fácil de implementar e pode resolver uma ampla gama de problemas. Num caso particular, a comparação de resultados obtidos por parte de Khan e Sahai [283] em problemas de classificação e na exploração de um caso *eLearning* mostrou que o BA recorreu a menos avaliações de funções para chegar a uma melhor solução com um erro médio menor comparado quando com outras técnicas como PSO ou GA.

## H. *Firefly algorithm*

Em verões como em Portugal, as pessoas são deslumbradas com as luzes dos pirilampos. Xin-She Yang adaptou esse comportamento para se inspirar no desenvolvimento de uma recente metaheurística intitulada por *Firefly algorithm* (FA). A produção de *flashes* curtos e rítmicos oferecem um padrão único da espécie, que até ao momento do estudo abordado, foram apenas interpretados três comportamentos, na habilidade de comunicação, caça e proteção [284]. Uma idealização simples da estrutura deste algoritmo pode ser destacada em três pontos: os pirilampos são atraídos entre si, independentemente do seu sexo; o brilho da luz é proporcional à sua atratividade procurando no espaço aleatoriamente; a luz é determinada pelo espaço da função objetivo. Com isto, os agentes podem interagir com outros proporcionando mecanismos de intensidade, mas também pode oferecer alguma diversificação baseado no Movimento Browniano que obedece, ou não, à função de Gauss. Contudo, a diversificação que obedeça à função de Gauss mostrou ter mais vantagens que uma abordagem não gaussiana [120].

Algoritmo 11 – *Firefly Algorithm*

---

*Algorithm - FA*

---

```

1 Inicializar a população de pirilampos
2 Computar intensidade da luz para todos os pirilampos
3 Escolher o coeficiente de absorção da luz
4 while (critério de paragem não satisfeito)
5     for todos os pirilampos
6         for todos os pirilampos (ciclo interno)
7             if (critério de paragem não satisfeito)
8                 Mover em direção de pirilampos
9             end if (critério de paragem não satisfeito)
10            Variar atratividade com a distância
11            Avaliar novas soluções e atualizar intensidade da luz
12        end for
13    end for
14 Classificar os pirilampos e encontrar o global best atual
15 end while
16 return visualizações e resultados

```

---



Nos últimos anos, o FA geral mostrou ser eficiente, no entanto, outras variantes ou algumas modificações no algoritmo original foram surgindo rapidamente tornando-se impossível enumerá-las, mas algumas dessas evoluções podem ser encontradas em Yang [120].

A relevância deste algoritmo tem sido amplamente discutida pelas suas características multimodais, capacidade em lidar com problemas eficientemente, com uma rápida taxa de convergência e procura local e global em todos os domínios.

Este método foi aplicado, por exemplo, por Banati et al. [285] com uma versão híbrida relativo a técnicas de pré-processamento em ML. Recorrendo a quatro conjuntos de dados médicos distintos, Banati et al. propôs uma simulação do sistema na atratividade entre pirilampos reais que encontre o melhor procedimento na seleção de recursos. Através disso, conseguiu superar os resultados previamente atingidos em termos de tempo e otimização [286].

## I. *Krill Herd*

Apresentado por Gandomi e Alavi [287], o *Krill Herd* (KH), foi inspirado nos movimentos do *krill* em grupo com o intuito de resolver problemas de otimização. Movimentos esses que são determinados por três ações essenciais ao longo do tempo, dependentes da sua posição, reação na presença de outros indivíduos, procura local e global paralela de alimento e na sua forma de difusão pelo espaço através de operadores evolutivos adaptativos (*mutation* e *crossover*), pois a utilização de procura estocástica aleatória isenta a possibilidade na derivação da informação.

Uma particular e grande vantagem deste algoritmo relacionando com outros algoritmos SI é o facto de apenas ser necessário ajustar o intervalo de tempo para cada problema.

Algoritmo 12 – *Krill Herd*

---

*Algoritmo – KH*

---

- 1 *Definir* os parâmetros e a estrutura dos dados
- 2 *Criar* aleatoriamente a população no espaço de pesquisa
- 3 *Avaliar* individualmente o krill de acordo com a sua posição
- 4 *while* (*critério de paragem não satisfeito*)
  - 5     *Induzir* movimentos pela presença de outros indivíduos
  - 6     Movimentos *Foraging*
  - 7     Difusão física
  - 8     *Implementar* operadores genéticos
  - 9     *Atualizar* a posição do krill individual no espaço de pesquisa
- 10 *end while*
- 11 *return* *melhor solução*

---

Cada agente pode contribuir para o processo de movimentação de acordo com a sua aptidão, o comportamento de atração/repulsão do vizinho, atuando como uma pesquisa local e o *global best* é considerado de acordo com o centro de comida de todos os indivíduos.

Quatro algoritmos KH diferentes foram derivados e criados para comprovar a eficiência do algoritmo proposto: KH sem qualquer operador genético, KH com operador de *crossover*, KH com o operador *mutator* e KH com ambos os operadores. Após resolver problemas de *benchmark*, o KH sem qualquer operador ou com ambos os operadores demonstraram obter melhor resultados do que muitos outros algoritmos [121].

Uma versão híbrida do algoritmo KH proposta por Wang et al. [287], foi testada em oito funções matemáticas de *benchmark* para indicar que esse algoritmo provavelmente seria até à data, o mais poderoso e eficaz dos algoritmos de otimização em problemas *population-based* [287].

## J. *Cuckoo Search*

Um dos mais recentes algoritmos metaheurísticos SI baseado no comportamento da família de aves *Cuculidae*, mais conhecidos por cucos, foi desenvolvido por Xin-She Yang e Suash Deb em 2009 e intitulado por *Cuckoo Search* (CS) [288]. Estas aves extraordinárias têm uma prática social parasitária, ao colocar os seus ovos em ninhos comuns e até mesmo livrar-se de ovos presentes nesses ninhos (até mesmo ovos da sua família de aves) [289].

Esta prática parasitária comporta-se de três formas. A ocupação do ninho verifica-se quando a ave hospedeira luta com o cuco hostil ou quando a ave hospedeira encontra um ovo do cuco e de seguida ela abandona o ninho ou tenta livra-se do ovo. O segundo tipo de comportamento envolve um cuco fêmea que tem uma boa capacidade em imitar cores e padrões de alguns outros ovos específicos, elevando a probabilidade de os seus ovos não serem rejeitados em ninhos de outras aves. O terceiro comportamento descreve o momento em que são colocados os ovos, numa perspetiva de reprodução cooperativa, porque os ovos de cuco normalmente chocam mais rápido. Quando isso acontece, o pequeno cuco começa a empurrar os ovos do hospedeiro para fora do ninho e tenta imitar o método de chamamento da ave hospedeira, com o intuito de aumentar as oportunidades de alimentação [120].

Como é sabido, um cuco é um animal voador, e este tipo de animais têm características típicas de *Lévy flights* [120,288]. Adicionando esta melhoria ao algoritmo CS, mostrou ser uma mais-valia em termos de eficiência do que outros algoritmos [120,288] porque *Lévy flights* têm média e variância infinita, levando a uma melhor exploração do espaço de procura, ou em convergência local e global.

Algoritmo 13 – *Cuckoo Search via Lévy Flights*

---

**Algoritmo** – *CS via Lévy Flights*

---

```

1 Função objetivo
2 Gerar a população inicial de ninhos hospedeiros
3 while (critério de paragem não satisfeito)
4     Obter um cuco aleatoriamente
5     Gerar uma solução através de Lévy flights
6     Avaliar a qualidade da solução ou a solução candidata
7     Escolher um ninho aleatoriamente
8     if (critério de paragem não satisfeito)
9         Trocar o ninho pela nova solução
10    end if
11    Uma fração dos piores ninhos é abandonada
12    Construir novos ninhos/soluções
13    Manter as melhores soluções
14    Classificar as soluções e encontrar a melhor atual
15    Atualizar o contador iterativo
16 end while
17 return Visualizações e resultados

```

---

Um estudo demonstrou recentemente que o CS supera significativamente outros algoritmos nas mais variadas aplicações [290].

### K. Comparação de *swarm intelligence*

Após a revisão de literatura, os algoritmos *swarm intelligence* foram aqueles, apesar de pertencerem a uma nova geração de algoritmos, mostraram ser os que melhor potenciam a sua aplicabilidade ao caso em estudo. No entanto, como a ferramenta utilizada não inclui estas técnicas de otimização nem permite a sua implementação, e como a sua importância é emergente e relevante, foi criada a tabela seguinte que apresenta uma análise às mesmas, representando um conjunto de características medidas e avaliadas consoante a sua funcionalidade: A – Alta; M – Média; B – Baixa; N – Não; S – Sim; C – *Crossover*; Mu – *Mutator*, Se – *Selector*.

Tabela 3 – Análise de técnicas *Swarm Intelligence*

Características	Algoritmos								
	OEP	KH	BBO	GSA	BA	FA	BCOB	ACO	CS
Velocidade treino	B	A	A	A	A	A	M	B	A
Utilização memória	A	B	B	B	B	B	M	M	B
Acerto preditivo	A	A	M	M	M	A	M	M	A
Interpretabilidade	B	M	B	M	M	M	A	A	A
Velocidade preditiva	B	A	M	A	A	A	A	M	M
Velocidade de ajuste	M	A	M	A	A	A	M	B	A
Lida com variáveis categóricas	N	S	N	N	S	S	N	N	N
Ajuste de parâmetros	S	S	S	N	S	S	N	N	S
Operadores genéticos	Se	C;Mu	Mu	C;Mu	Mu;S	Mu;S	N	N	N
Intensificação (local)	A	A	B	B	A	A	B	B	A
Diversificação (global)	B	A	A	A	A	A	A	A	A

Ao verificar a Tabela 3, podemos constatar que o CS, FA e KH são os melhores algoritmos tendo por análise as suas características, nomeadamente algumas semelhanças na alta velocidade de treino e de ajuste, baixa utilização de memória que potencia a rapidez na sua execução explorando o espaço de pesquisa, tanto a nível local como global. O CS, apesar de ter uma velocidade preditiva média, a sua interpretabilidade é alta, o que o torna uma mais-valia para a perceber como se comportaram os indivíduos e que valores estão associados à conclusão do seu comportamento. O KH e o FA, distinguem-se do CS na sua capacidade de lidar com variáveis categóricas, que dependendo do sistema onde se pretende implementar, esta categoria pode ser um fator preponderante.

## 7. Análise dos dados

Relembrando a questão de investigação apresentada na [secção 1.1](#), a revisão da literatura efetuada no [capítulo 2](#), como o INTCare funciona e que técnicas que estão subjacentes, apresentadas na [secção 4.1](#), os modelos de apoio à decisão existentes no [capítulo 5](#) e que técnicas podem otimizar esses modelos no [capítulo 6](#). As subsecções seguintes enquadram o processo de *benchmarking* e a investigação quantitativa nas mesmas fases e objetivos de estudo.

Para analisar a recolha de dados efetuada, foram revistas várias ferramentas que conseguissem lidar com o máximo de técnicas identificadas.

Inicialmente existiu a ponderação da utilização da Linguagem R, onde várias destas técnicas poderiam ser criadas ou utilizadas, mas a sua implementação e execução poderiam comprometer o sucesso desta dissertação no cumprimento do prazo de entrega pelo facto de a experiência nessa linguagem ser um pouco limitada para este nível de complexidade.

Após isso, foi tentada uma ferramenta muito utilizada pela comunidade científica, o MATLAB. Com esta ferramenta seria possível implementar na totalidade os algoritmos estudados pois todas as técnicas de otimização (ou variantes) existem nesta linguagem matemática. No entanto após algumas tentativas a utilização desta ferramenta foi abandonada pelo facto de existir carência de conhecimento desta linguagem e a existência de pouco tempo para a aprender. Contudo, qualquer uma das ferramentas seria possível, mas em outro âmbito, como por exemplo, num projeto de investigação.

Entretanto, a ferramenta KEEL surgiu e foi preponderante a sua escolha pela potencialidade em variedade de algoritmos baseados em várias formas de ML, a facilidade de utilização e *benchmark*.

### 7.1.Ferramentas utilizadas

Para uma análise detalhada das técnicas de otimização identificadas (e variantes) foi utilizada uma máquina com características de *hardware* e *software* a seguir descritas:

- **Sistema Operativo:** Windows 8.1 Pro 64-bit
- **Processador:** Intel Mobile Core 2 Duo T9400
- **RAM:** 512MB
- **Armazenamento:**298GB

Na utilização de ferramentas recorreu-se ao uso do *software* MATLAB (para tratamento dos dados (ex. limpeza dos nulos) e o KEEL. A seguir encontra-se o detalhe da principal ferramenta utilizada e a sua visão técnica, na [secção 7.2](#), o que foi realizado em cada uma com maior detalhe.

### **7.1.1. Visão técnica KEEL**

KEEL (*Knowledge Extraction based on Evolutionary Learning*) é uma ferramenta em Java GPLv3 desenvolvida como um pacote de aplicações de ferramentas ML [291].

Projetado para proporcionar um conjunto de soluções para problemas de DM e avaliar algoritmos evolucionários. Contempla-se com uma coleção de bibliotecas para pré-processamento e pós-processamento de técnicas para manipulação de dados, métodos *soft-computing* na extração e aprendizagem de conhecimento como também fornece métodos científicos e de investigação. Desenvolvida em 2004 em parceria com o ministério da educação e ciência espanhol, este *software* tem sido atualizado estando disponível atualmente a versão KEEL 3.0. Utilizando a licença pública GNU, é dotada por ser uma solução *cross-platform* baseada em linguagem *Java* encontrando-se disponível em <http://sci2s.ugr.es/keel/download.php>.

KEEL é uma ferramenta para avaliar algoritmos evolutivos em problemas de DM, mas também é considerada uma ferramenta de ML.

As vantagens na utilização passam por esta incluir regressões, classificações, *Clustering*, descoberta de padrões, entre outras. Contém também uma grande coleção de algoritmos clássicos na extração de conhecimento, técnicas de pré-processamento (por exemplo, por recurso, discretização, métodos de imputação de valores em falta, etc.), algoritmos CE, incluindo aprendizagem por regras evolucionárias com base em diferentes abordagens, modelos híbridos, tais como sistemas *fuzzy* genéticos, redes neuronais evolucionárias, etc.

Em termos de limitações, a sua eficiência é restrita pelo número de algoritmos que suporta, comparando a outras ferramentas [292] e restringe o acesso à totalidade da memória disponível na máquina, ficando limitada a 512Mb.

## **7.2. Exploração dos dados**

Para explorar o conjunto de dados com 14139 entradas que foram utilizados, foi necessário compreendê-los. Após uma breve análise ao que cada variável significava, através de um

algoritmo de Classificação associativa presente o KEEL, foi possível determinar quais eram as variáveis mais preponderantes para uma extubação<sup>2</sup> com sucesso. Após a obtenção dos resultados, foi criado um novo conjunto de dados que a tabela descreve, nomeadamente, o que cada variável de *input* significa, a sua média, desvio padrão e variância.

Tabela 4 – Descrição das variáveis

Variável	Descrição	Média	Desvio padrão	Variância
F12	Velocidade de expiração por minuto	24,00	23,55	554,86
F13	%O2	49,75	7,65	58,56
F15	Pressão nos pulmões em cmH2O	5,09	1,05	1,11
F26	Pressão do suporte respiratório em cmH2O	14,11	3,60	12,98
F35	Volume da capacidade de expiração em litros	0,52	0,16	0,02
F38	Pressão máxima do circuito em cmH2O	20,53	3,86	14,95
F39	Volume da expiração por minuto em litros	10,50	1,84	3,40
F40	Pressão no final da inspiração em cmH2O	19,98	3,85	14,86
F65	Complacência pulmonar durante fluxos de gás	45,75	40,18	1615,23

Como *output* temos a variável Target, que traduz três acontecimentos. Quando o valor é 0, significa que não foi possível extubar o doente, quando o valor é 1, o doente foi extubado, mas voltou a necessitar de suporte respiratório e quando o valor é 2, foi realizada uma extubação com sucesso.

<sup>2</sup> Extubação – Ação de retirar o tubo que suporta a ventilação mecânica artificial



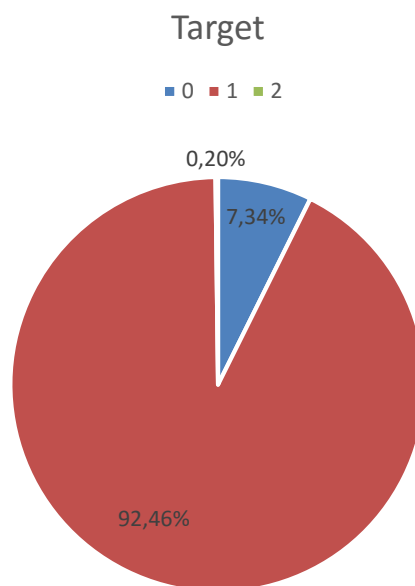


Figura 7 – Representação do *output* no conjunto de dados fornecidos

Como podemos verificar, esta representação demonstra uma limitação na melhor demonstração da exequibilidade destas técnicas de otimização pois os casos de sucesso são praticamente escassos.

### 7.3. Tratamento dos dados

Os dados utilizados para estas experiências são provenientes de outros trabalhos já efetuados, em particular o de Sérgio et al. [293] para a Unidade de Cuidados Intensivos do Centro Hospitalar do Porto (CHP). Esse trabalho teve como objetivo identificar um conjunto de características/variáveis do doente que comprometem a determinação do momento para uma entubação com sucesso. A maioria dos dados facultados já se encontravam caracterizados e transformados pelo sistema INTCare e eram resultantes da aplicação de técnicas DM, mais propriamente na criação de *clusters*, através dos algoritmos *k-means* e *k-medoids* baseadas pelo princípio de partição e diferença de sensibilidade a valores extremos [293]. O resultado obtido por parte desse trabalho mais interessante foi identificado no *cluster 3* com 2.382 ocorrências (representando 18.13% do conjunto de dados), onde obteve a média mais baixa na soma dos valores quadrados de 0,52.

Para tornar os dados utilizáveis para este trabalho, foi necessário eliminar todas as linhas que tinham valores nulos, onde cada linha representa um doente. Esta etapa contém o pequeno

contributo que o *software* MATLAB prestou, ao utilizar a opção *Import* ao ficheiro fornecido pelo INTCare, através da sua ferramenta de extração e filtragem de dados.

De seguida foi abandonada a sua utilização pelas razões apontadas na introdução do [capítulo 7](#). Após isso o foco deste trabalho centrou-se na utilização da ferramenta KEEL. Começando por importar os dados filtrados pelo MATLAB intitulado pelo ficheiro “extratados012varimportantes.xls”.



Figura 8 – Interface principal do KEEL

A Figura 8, apresenta as opções da interface principal do KEEL. No KEEL podemos criar um conjunto de dados que pretendemos para realizar as experiências pretendidas. Para isso, selecionamos “*Data Management*” onde permite a sua criação e tratamento ao selecionar a opção “*Import Data*”. Após a seleção do conjunto de dados, podemos visualizar e definir as variáveis de *input*, *output* e tratá-las da forma que pretendermos, neste caso foi necessário por exemplo alterar o tipo de dados de “*integer*” para “*real*” nos *inputs* e no *output* de “*integer*” para nominal apenas clicando no “*type*” de forma a evitar conflitos na sua utilização. Após isso, o conjunto de dados foi dividido por 5x2 *cross-validation* apresentado

por Dietterich em 1998 [294], formando 5 replicações igualmente dimensionadas de 2 conjuntos de validação, em conjunto de treino e teste.

## 7.4. Aplicação

Para exemplificar a aplicação de um cenário experimental, a Figura 9 mostra um de muitos testes aos algoritmos que o KEEL suporta. Inicialmente selecionam-se os conjuntos de treino e teste respectivamente, representados pelo ícone “data”, que transmite os dados para os algoritmos representados com um cubo azul. Nos casos em que o conjunto de dados não era compatível com o algoritmo que se pretendeu executar, era acionado um alerta acerca do problema de formato. Os parâmetros dos algoritmos podem ser alterados, através de um duplo *click*, mas foram mantidos os pré-definidos. Após isso, deve-se ligar os algoritmos ao extrator de resultados. Com isso a experiência é extraída para um de dois formatos, visualização em lista ou tabulação.

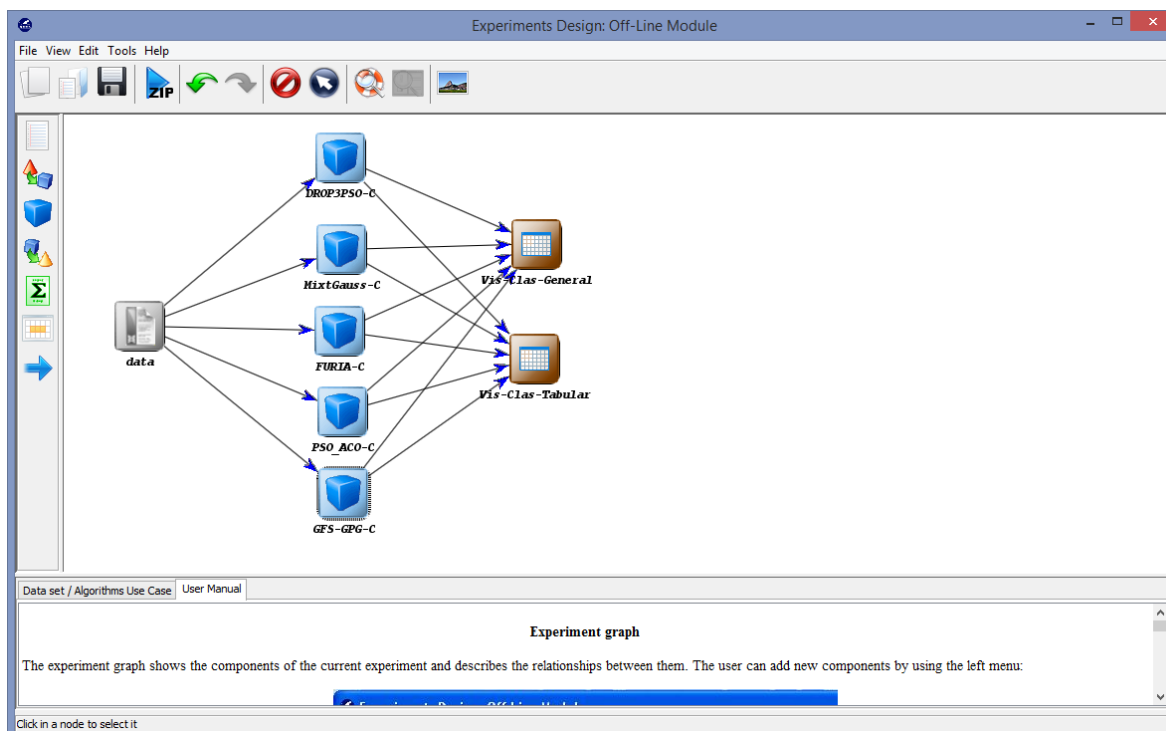


Figura 9 – Cenário de uma experiência na ferramenta KEEL

Existem outros algoritmos passíveis de serem testados, mas como o seu processamento era demorado e como por vezes não gerava resultados, várias das experiências ficaram comprometidas, tornando-as não conclusivas, apesar de terem sido testadas. Ao todo foram efetuados 63 testes a algoritmos propostos posteriormente ao ano 2000, os quais englobam

10 modelos *Evolutionary Fuzzy Rule Learning*, 16 *Evolutionary Crisp Rule Learning*, 4 *Lazy Learning*, 7 *Prototype Generation*, 5 *Fuzzy Instance based Learning*, 7 Árvores de Decisão, 5 *Crisp Rule Learning*, 4 Redes Neurais Artificiais, 5 *Evolutionary Prototype Selection* determinando como critério de seleção a média global da acuidade (do conjunto de dados teste) do caso em estudo. Os resultados dos algoritmos testados encontram-se no [apêndice 1.1](#), onde são discriminados os algoritmos experimentados determinando o seu acrônimo, ano de apresentação, se funcionou e se obteve algum resultado.

Após um exaustivo teste às técnicas presentes no KEEL, foram selecionadas as 3 melhores técnicas, o algoritmo híbrido árvore de decisão com algoritmo genético *Decision Trees Genetic Algorithm* (DT\_GA) [295], o algoritmo híbrido entre regras *fuzzy, learning classifier systems* e *data mining sUpervised Classifier System* (UCS) [296] e o *algoritmo K-Nearest Neighbors Adaptive* (KNNAdaptive) [297] como melhores otimizadores para viabilizar a utilização de técnicas de otimização para melhorar o processo de decisão ao determinar as condições para uma extubação com sucesso na medicina intensiva.

Tabela 5 – Parâmetros utilizados no estudo experimental das melhores técnicas

Algoritmo	Parâmetros
DT_GA	Confiança: 0.25, instâncias por folha: 2, aproximação algorítmica genética: GA-LARGE-SN, limite S para considerar uma pequena disjunção:10, número de gerações para o algoritmo genético: 50, número de cromossomas na população: 200, probabilidade <i>crossover</i> : 0.8, probabilidade <i>mutation</i> : 0.01
UCS	Número de explorações: 100.000, tamanho da população: 6400, delta: 0.1, nu: 10, Acc: 0.99, pX: 0.8, pM: 0.04, theta_ga: 50, theta_del: 50, theta_sub: 500, doGASubsumption: <i>true</i> , tipo de seleção: RWS, tamanho do torneio: 0.4, tipo de <i>mutator</i> : <i>free</i> , tipo de <i>crossover</i> : 2PT, r: 0.6, m: 0.1,
KNNAdaptive	Valor de k: 1, função distância: euclidiana

Para DT\_GA, os parâmetros definidos no KEEL obtiveram uma taxa de acuidade de 93.2%, para UCS de 93.1% e para o algoritmo KNNAdaptive de 92.97%.

A Tabela 6, Tabela 7 e Tabela 8 mostram as médias dos resultados obtidos seguindo os testes *Friedman test* [298], *Multiple test* [298] e *Contrast Estimation* [299], a matriz de confiança para os três níveis descritos e a matriz de confusão de cada algoritmo respetivamente.

Tabela 6 – Testes de classificação das melhores técnicas

<b>Média de erro na classificação</b>			
	<i>Friedman test</i>	<i>Multiple test</i>	<i>Contrast Estimation</i>
DT_GA	0.06	0.06	0.06
UCS	0.07	0.07	0.07
KNNAdaptive	0.07	0.07	0.07

Tabela 7 – Matriz nível de confiança das melhores técnicas

<b>Matriz p-value para níveis de confiança 0.99/0.95/0.9</b>			
	DT_GA	UCS	KNNAdaptive
DT_GA	0.00	0.09	0.25
UCS	0.00	0.00	0.96
KNNAdaptive	0.00	0.00	0.00

De forma a interpretar os resultados passíveis de ser analisados a técnica DT\_GA, definiu algumas árvores de decisão compostas pelas seguintes folhas (regras):

Rule: IF F26 > 11.0 AND F40 > 18.0 AND F65 <= 124.0 AND F39 > 8.6 AND F38 <= 21.0 AND F13 > 40.0 AND F12 > 43.0 AND F15 > 4.0 AND F35 <= 0.44 THEN Class = 2 (3/291)

Rule: IF F26 <= 25.0 AND F40 > 12.0 AND F65 > 33.0 AND F39 > 6.9 AND F38 <= 25.0 AND F13 <= 50.0 AND F12 > 0.0 AND F15 <= 5.0 AND F35 > -0.6799999999999999 THEN Class = 2 (3/1850)

Rule: IF F26 <= 18.0 AND F40 <= 21.0 AND F65 <= 40.0 AND F39 > 8.5 AND F38 > 13.0 AND F13 <= 60.0 AND F12 <= 43.0 AND F15 > 3.0 AND F35 > -0.37 THEN Class = 2 (2/2003)

Rule: IF F26 <= 14.0 AND F40 > 11.0 AND F65 > 12.0 AND F39 <= 16.0 AND F38 > 11.0 AND F13 > 49.0 AND F12 <= 45.0 AND F15 > 3.0 AND F35 <= 0.26 THEN Class = 2 (1/55)

Rule: IF F35 <= 0.39 AND F15 <= 6.5 AND F12 > 0.0 AND F65 <= 30.0 AND F65 > 18.0 AND F65 > 24.0 AND F38 <= 24.0 AND F12 > 46.0 THEN Class = 2 (2/2)

Rule: IF F26 > 10.0 AND F40 <= 20.0 AND F65 <= 27.0 AND F39 <= 13.0 AND F38 <= 21.0 AND F13 <= 45.0 AND F12 <= 68.0 AND F15 <= 6.0 AND F35 > -0.71 THEN Class = 2 (3/253)

Rule: IF F26 <= 13.0 AND F40 <= 17.0 AND F65 > 12.0 AND F39 > 5.4 AND F38 > 9.0 AND F13 > 49.0 AND F12 > 40.0 AND F15 > 4.0 AND F35 > -0.65 THEN Class = 2 (2/125)

Rule: IF F26 <= 15.0 AND F40 > 21.0 AND F65 > 11.0 AND F39 <= 8.8 AND F38 <= 40.0 AND F13 <= 50.0 AND F12 <= 57.0 AND F15 > 4.0 AND F35 <= 0.63 THEN Class = 2 (1/1)

Rule: IF F26 > 9.0 AND F40 <= 17.0 AND F65 > 35.0 AND F39 <= 9.3 AND F38 > 16.0 AND F13 > 49.0 AND F12 > 0.0 AND F15 > 5.0 AND F35 <= 0.57 THEN Class = 2 (1/2)

Fazendo um breve análise dos resultados obtidos é possível verificar que um doente que tenha valores na velocidade de expiração por minuto entre 9 e 25, uma percentagem de O<sub>2</sub> entre 11 e 21, uma pressão nos pulmões em cmH<sub>2</sub>O entre 11 e 40, uma pressão do suporte respiratório em cmH<sub>2</sub>O entre 5.9 e 16, volume da capacidade de expiração entre 9 e 40 litros, uma pressão máxima do circuito em cmH<sub>2</sub>O entre 40 e 60, um volume da expiração por minuto entre 0 e 57 litros, uma pressão no final da inspiração em cmH<sub>2</sub>O entre 3 e 6, e finalmente a complacência pulmonar<sup>3</sup> durante fluxos de gás entre -0.65 e 0.63, estaria perante um cenário onde seria submetido a um processo de extubação e o mesmo seria realizado com sucesso.

Esta informação será apresentada aos intensivistas de uma forma intuitiva de modo a facilitar a leitura e interpretação das regras criadas e contribuir para o suporte ao processo de decisão.

---

<sup>3</sup> Complacência pulmonar – Capacidade de extensão dos pulmões a cada aumento de pressão pulmonar

Tabela 8 – Matrizes de confusão das melhores técnicas

<b>Matriz de confusão DT_GA</b>			
	0	2	1
0	98	0	874
2	6	0	27
1	53	0	13080

<b>Matriz de confusão KNNAdaptive</b>			
	0	2	1
0	315	2	655
2	6	1	26
1	290	14	12829

<b>Matriz de confusão UCS</b>			
	0	2	1
0	264	2	706
2	5	0	28
1	247	4	12882

### 7.5. Proposta de medidas

Ao ter por base os estudos já efetuados no INTCare [12,13,29,30,32,33,34,293] e os resultados obtidos tendo em conta o processo de extubação e a sua acuidade dos modelos e as técnicas identificadas com este trabalho, as mesmas devem ser consideradas em futuras implementações dos modelos para a previsão de situações clínicas futuras, como foi o caso da extubação de um doente na unidade de cuidados intensivos do centro hospitalar do Porto. O algoritmo DT\_GA mostrou ser o mais capaz de responder à questão de investigação, sendo capaz de obter uma taxa de acuidade de 93.2%.

## 8. Conclusão

Como base nos resultados associados ao SIAD implementado no INTCare, enriquecido neste caso pelos monitores de cabeceira (sinais vitais e ventilados) presente em cada cama onde o doente é acompanhado, as técnicas de DM que classificam os dados para obter análises preditivas associadas à extubação de um doente são diminutas, sendo que apenas foi possível aplicar técnicas de *clustering* [293]. Ao longo deste documento esse estudo foi estendido por uma investigação quantitativa sobre um processo de *benchmarking*, onde foram analisados os modelos de apoio à decisão que são passíveis de serem aplicados, que técnicas de otimização existentes podem ser incluídas nesses modelos e que variáveis mostraram ser preponderantes para definir a causalidade entre os *inputs* e o *output* em explicar, com o melhor desempenho, o que é observado para este problema em causa. Após a extensa revisão de literatura e comparação das técnicas mais adequadas foram identificadas as melhores técnicas resultantes de 63 experiências efetuadas, das quais 10 representam modelos de apoio à decisão *Evolutionary Fuzzy Rule Learning*, 16 *Evolutionary Crisp Rule Learning*, 4 *Lazy Learning*, 7 *Prototype Generation*, 5 *Fuzzy Instance based Learning*, 7 Árvores de Decisão, 5 *Crisp Rule Learning*, 4 Redes Neurais Artificiais, 5 *Evolutionary Prototype Selection* implementados no *software* KEEL. Foram selecionados modelos *Evolutionary Crisp Rule Learning*, *Lazy Learning* e Árvores de Decisão de onde resultaram as técnicas DT\_GA, UCS e a KNNAdaptive com sucesso na taxa de acuidade de 93.2%, 93.1% e 92.97% respetivamente. Com isso, é respondido com sucesso a todos os objetivos propostos e à questão de investigação que originou esta dissertação, onde a técnica DT\_GA demonstrou ser viável na melhoria do processo de decisão para uma extubação com sucesso em medicina intensiva. Como trabalho futuro outros modelos do sistema INTCare devem ser experimentados e testados utilizando as técnicas de otimização descritas neste documento.

### 8.1. Conhecimento adquirido e trabalho futuro

Esta dissertação teve vários fatores de aprendizagem que culminou com a obtenção de novo conhecimento útil na área dos algoritmos e modelos de otimização. Para a concretização dos objetivos propostos foi necessário elaborar uma revisão de literatura muito aprofundada para entender este tema, pois um processo de *benchmarking* mais comum é utilizado nas organizações sobre as pessoas, e este caso é sobre os vários modelos sobre técnicas de forma



a otimizá-los. Das técnicas identificadas, este trabalho gerou uma nova percepção na forma como algumas das recentes meta-heurísticas *Population based* interpretam os indivíduos, representam as suas interações no ambiente (decisões) e as suas variáveis (comportamento). Como aspetos positivos de aprendizagem, salienta-se o desafio, a criação de um vasto guia com modelos, técnicas e aplicações em situações reais, a contribuição na aplicação de três técnicas para responder a uma necessidade real e o meu enriquecimento na área dos sistemas de apoio à decisão. Como aspetos negativos, realço a minha dificuldade inicial em perceber realmente o que era pedido e a não implementação de todas as técnicas *swarm intelligence*. A ferramenta utilizada mostrou ser a solução mais viável para o tempo disponível para esta dissertação, devido à falta de conhecimento em por exemplo o MATLAB e a linguagem R, mas ao mesmo tempo requeria bastante tempo para executar alguns algoritmos. Para trabalho futuro, sugiro que estas técnicas sejam investigadas em outras áreas de estudo.

## 9. Referências Bibliográficas

- [1] Nipun Agarwal, “Negotiation Decision Support Systems: Analysing Negotiations Under the Conditions of Risk”, *Journal of Business Systems, Governance and Ethics*, Volume 4, 3ª edição, pp 11-25. (2009)
- [2] Peter Braun , Jakub Brzostowski , Gregory Kersten , Jin Baek Kim , Stefan Strecker e Rustam Vahidov, “e-Negotiation Systems and Software Agents: Methods, Models, and Applications”, *Intelligent decision-making support systems: foundations, applications and challenges / Jatinder N. D. Gupta, Guiseppe A. Forgionne and Manuel Mora T., Parte II, Capítulo 15, pp. 271-300. (2006)*
- [3] Fulcher, J., & Jain C., L. “Computational Intelligence: A Compendium”. (2008)
- [4] Domingos, P. “A few useful things to know about machine learning”. *Communications of the ACM*, 55(10), 78. Retrieved from <http://www.mendeley.com/catalog/few-useful-things-know-about-machine-learning/>, (2012)
- [5] Pardalos, P. M., & Romeijn, H. E. “Handbook of Global Optimization vol.2.”, (2005)
- [6] Shearer C, Caron P (2002) *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, UK.
- [7] P. Bennet, K., & Parrado-Hernández, E. “The Interplay of Optimization and Machine Learning Research”. *Journal of Machine Learning Research*, 7. (2006)
- [8] Harrington, H. James e Harrington, James S.; “The Complete Benchmarking Implementation Guide”, (1996)
- [9] Czarnecki, Mark T., “Managing by Measuring”, AMACOM, (1999)
- [10] K. Rigby, D. “Management Tools an executive’s guide”, (2011)
- [11] Kelessidis, V., “Benchmarking: dissemination of innovation management and knowledge techniques”, (2010)
- [12] Santos, M.F., Portela, F., Vilas-Boas, M., “INTCARE : multi-agent approach for real-time intelligent decision support in intensive medicine”, *SciTePress*, (2011)
- [13] Portela, F., Pinto, F., Santos, M.F., “Data mining predictive models for pervasive intelligent decision support in intensive care medicine”, (2012)
- [14] McGinn, C. A., Grenier, S., Duplantie, J., Shaw, N., Sicotte, C., Mathieu, L., Gagnon, M. P., “Comparison of user groups’ perspectives of barriers and facilitators to implementing electronic health records: a systematic review”, *BMC Medicine*, 9, 46. doi:10.1186/1741-7015-9-46, (2011)
- [15] Schade, C. P., Sullivan, F. M., de Lusignan, S., & Madeley, J., “e-Prescribing, efficiency, quality: lessons from the computerization of UK family practice”, *Journal of the American Medical Informatics Association: JAMIA*, 13(5), 470–5. doi:10.1197/jamia.M2041, (2006)
- [16] Grieger, D. L., Cohen, S. H., & Krusch, D. A., “A pilot study to document the return on investment for implementing an ambulatory electronic health record at an academic medical center”, *Journal of the American College of Surgeons*, 205(1), 89–96. doi:10.1016/j.jamcollsurg.2007.02.074, (2007)
- [17] Ramon, J., Fierens, D., Güiza, F., Meyfroidt, G., Blockeel, H., Bruynooghe, M., & Van Den Berghe, G., “Mining data from intensive care patients”, *Advanced Engineering Informatics*, 21(3), 243–256. doi:10.1016/j.aei.2006.12.002, (2007)
- [18] Kohn, L. T., Corrigan, J. M., & Donaldson, M. S., “To Err is Human: Building a Safer Health System”, *Proceedings of the First Workshop on* (p. 312). Retrieved from <http://roc.cs.berkeley.edu/talks/pdf/easy01.pdf>, (2000)
- [19] A.H. Morris, R.M. Gardner, in: J. Hall, G. Schmidt, L. Wood (Eds.), “Principles of Critical Care”, *Mc Graw-Hill, NewYork*, pp. 500–514,(Chapter on Computer applications), (1992)
- [20] De Turck, F., Decruyenaere, J., Thysebaert, P., Van Hoecke, S., Volckaert, B., Danneels, C., De Moor, G., “Design of a flexible platform for execution of medical decision support agents in the intensive care unit”, *Computers in Biology and Medicine*, 37, 97–112. doi:10.1016/j.combiomed.2005.10.004, (2007)
- [21] Da Silva Oliveira, F. R., & De Lima Neto, F. B., “Incorporating user cognitive profile information in intelligent decision support systems”, *Conference Proceedings IEEE International Conference on Systems*,

- Man and Cybernetics, (October), 4391–4396. doi:10.1109/ICSMC.2009.5346933, (2009)
- [22] E. Turban, J. E. Aronson, TP Liang and R. Sharda, "Decision support and business intelligence systems", 8th. Edition, Prentice Hall, New York, USA, (2006)
- [23] Gilli, M., & Winker, P., "A review of heuristic optimization methods in econometrics", *Heuristic Optimization Methods in Econometrics*. <http://doi.org/10.1002/9780470748916.ch3>, (2008)
- [24] Babbie, Earl R., "The Practice of Social Research", 12th ed. Belmont, CA: Wadsworth Cengage, (2010)
- [25] Muijs, D., "Doing Quantitative Research in Education with SPSS", 2nd edition. London: SAGE Publications, (2010)
- [26] Kozak, M., "Destination Benchmarking Concepts, Practices and Operations", Retrieved from <http://pt.scribd.com/doc/133336551/Metin-Kozak-Destination-Benchmarking-Concepts>, (2013)
- [27] Khurram S. Bhutta, Faizul Huq, "Benchmarking – best practices: an integrated approach", *Benchmarking: An International Journal*, Vol. 6 Iss: 3, pp.254 – 268, (1999)
- [28] Gago, P., Santos, M.F., Silva, Á., Cortez, P., Neves, J., Gomes, L., "INTCare: a knowledge discovery based intelligent decision support system for intensive care medicine", *Journal of Decision Systems*, (2006)
- [29] Portela, F., Santos, M., Vilas-Boas, M., Rua, F., Silva, Á., Neves, J., "Real-time Intelligent decision support in intensive medicine", *KMIS 2010-International Conference on Knowledge Management and Information Sharing*, Valência, Espanha, p. 7, (2010)
- [30] Santos, M., Portela, F., Vilas-Boas, M., Machado, J., Abelha, A., Neves, J., Silva, Á., Rua, F., Salazar, M., Quintas, C., Cabral, A., "Intelligent Decision Support in Intensive Care Units - Nursing Information Modelling", (2009)
- [31] Wooldridge, M., "Intelligent agents. In: *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*", pp. 27–77. MIT Press (1999)
- [32] Portela, C.F., Santos, M.F., Silva, Á., Machado, J., Abelha, A., "Enabling a Pervasive Approach for Intelligent Decision Support in Critical Health Care", In: Cruz-Cunha, M.M., Varajão, J., Powell, P., Martinho, R. (eds.) *CENTERIS 2011, Part III. CCIS*, vol. 221, pp. 233–243. Springer, Heidelberg (2011)
- [33] Portela, F., Santos, M.F., Vilas-Boas, M., "A Pervasive Approach to a Real-Time Intelligent Decision Support System in Intensive Medicine", In: Fred, A., Dietz, J.L.G., Liu, K., Filipe, J. (eds.) *IC3K 2010. CCIS*, vol. 272, pp. 368–381. Springer, Heidelberg (2013)
- [34] Portela, F., Santos, M. F., Machado, J., Abelha, A., Silva, Á., & Rua, F., "Pervasive and intelligent decision support in intensive medicine-the complete picture", *Information Technology in Bio-and Medical Informatics*, 87–102. [http://doi.org/10.1007/978-3-319-10265-8\\_9](http://doi.org/10.1007/978-3-319-10265-8_9), (2014)
- [35] Saha, D., Mukherjee, A., "Pervasive computing: a paradigm for the 21st century", *IEEE Computer* 36, 25–31 (2003)
- [36] Rogers S, Girolami M., "A first course in machine learning", New York, NY, USA: CRC Press Inc.; (<http://books.google.co.uk/books?id=rdQ1daD8BH8C>), (2011)
- [37] Fraccaro, P, et al., "Behind the screens: Clinical decision support methodologies – A review", *Health Policy and Technology*, doi:10.1016/j.hlpt.2014.10.001, (2014)
- [38] Steyerberg E. W., "Clinical prediction models: a practical approach to development, validation, and updating", New York, NY, USA, Springer, (2009)
- [39] Ji S-Y, Smith R, Huynh T, Najarian K., "A comparative analysis of multi-level computer-assisted decision making systems for traumatic injuries", *BMC Med Inform Decis Mak*, 9(1):2. January (<http://www.biomedcentral.com/1472-6947/9/2>), (2009)
- [40] M. F. Santos, W. Mathew, T. Kovacs, and H. Santos., "A grid data mining architecture for learning classifier systems". *W. Trans. on Comp.* 8, 5, 820-830, (2009)
- [41] Vapnik, V., "Statistical Learning Theory", Wiley Press, New York, 493–520, (1998)
- [42] Cristianini N, Shawe-Taylor J., "An introduction to support vector machines and other Kernel-based learning methods", New York, NY, USA, Cambridge University Press, Available from: (<http://books.google.it/books?id=B-Y88GdO1yYC>), (2000)
- [43] Waghlikar, K. B., Sundararajan, V., Deshpande, A. W., "Modeling paradigms for medical diagnostic

decision support: A survey and future directions”, *Journal of Medical Systems*, 36, 3029–3049. doi:10.1007/s10916-011-9780-4, (2012)

[44] Chi C.L., Street WN, Ward MM., “Building a hospital referral expert system with a prediction and optimization-based decision support system algorithm”, *J Biomed Inform*, 41(2):371–86. Available from: <http://www.sciencedirect.com/science/article/pii/S1532046407001086>, (2008)

[45] Cho BH, Yu H, Kim K-W, Kim TH, Kim IY, Kim SI., “Application of irregular and unbalanced data to predict diabetic nephropathy using visualization and feature selection methods”, *Artif IntellMed. Elsevier*, 42(1):37–53. Available from: [http://www.aiimjournal.com/article/S0933-3657\(07\)00115-7/abstract](http://www.aiimjournal.com/article/S0933-3657(07)00115-7/abstract), (2008)

[46] Kamruzzaman, J., Begg R.K., “Support vector machines and other pattern recognition approaches to the diagnosis of cerebral palsy gait”, *IEEE Transactions on BioMedical Engineering*, 53(12 Pt 1):2479–2490, (2006)

[47] Brewer, B. R., Pradhan, S., Carvell, G., Delitto, A., “Feature selection for classification based on fine motor signs of parkinson’s disease”, *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009:214–217, (2009)

[48] C. Lu, T. Van Gestel, J. A. Suykens, S. Van Huffel, I. Vergote, and D. Timmerman, “Preoperative prediction of malignancy of ovarian tumors using least squares support vector machines”, *Artificial Intelligence in Medicine*, 28(3):281–306, (2003)

[49] Qianfei Yuan, Congzhong Cai, Hanguang Xiao, Xinghua Liu, and Yufeng Wen, “Svm-aided cancer diagnosis based on the concentration of the macroelement and microelement in human blood”, *Journal of Biomedical Engineering*, 24(3):513–518, (2007)

[50] Picton P., “Neural networks”, New York, NY, USA: Palgrave Macmillan; <http://books.google.it/books?id=mBk6qAAACAAJ>. (2000)

[51] Tu JV., “Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes”, *J Clin Epidemiol. Elsevier*;49(11):1225–31. November. Available from: [http://www.jclinepi.com/article/S0895-4356\(96\)00002-9/abstract](http://www.jclinepi.com/article/S0895-4356(96)00002-9/abstract), (1996)

[52] Rudy Setiono, Wee K. Leow, and James Y. L. Thong, “Opening the neural network black box: an algorithm for extracting rules from function approximating artificial neural networks”, In *ICIS '00: Proceedings of the twenty first international conference on Information systems*, pp. 176–186, Atlanta, GA, USA. Association for Information Systems, (2000)

[53] W. G. Baxt, “Use of an artificial neural network for the diagnosis of myocardial infarction”, *Annals of Internal Medicine*, 115(11):843–848, (1991)

[54] P. J. Lisboa, “A review of evidence of health benefit from artificial neural networks in medical intervention”, *Neural Networks: the Official Journal of the International Neural Network Society*, 15(1):11–39, (2002)

[55] R. L. Kennedy, R. F. Harrison, A. M. Burton, H. S. Fraser, W. G. Hamer, D. MacArthur, R. McAllum, and D. J. Steedman, “An artificial neural network system for diagnosis of acute myocardial infarction (ami) in the accident and emergency department: evaluation and comparison with serum myoglobin measurements”, *Computer Methods and Programs in Biomedicine*, 52(2):93–103, (1997)

[56] M. L. Astion, M. H. Wener, R. G. Thomas, G. G. Hunder, and D. A. Bloch, “Application of neural networks to the classification of giant cell arteritis”, *Arthritis and Rheumatism*, 37(5):760–770, (1994)

[57] J. Li, Y. Mu, and L. Zhang., “Study of the pulmonary heart disease computer-aided diagnosis system based on combining neural network”, *Journal of Biomedical Engineering*, 18(4):573–576, (2001)

[58] Resul Das, Ibrahim Turkoglu, and Abdulkadir Sengur, “Diagnosis of valvular heart disease through neural networks ensembles”, *Computer Methods and Programs in Biomedicine*, 93(2):185–191, (2009)

[59] Lutz Leistritz, Mirosław Galicki, Eberhard Kochs, Ernst Bernhard B. Zwick, Clemens Fitzek, Jürgen R. Reichenbach, and Herbert Witte, “Application of generalized dynamic neural networks to biomedical data”, *IEEE Transactions on Bio-Medical Engineering*, 53(11):2289–2299, (2006)

[60] I. G. Vlachonikolis, D. A. Karras, M. J. Hatzakis, and N. Paritsis, “Improved statistical classification methods in computerized psychiatric diagnosis”, *Medical Decision Making: an International Journal of*

- the Society for Medical Decision Making, 20(1):95–103, (2000)
- [61] N. H. Mann and M. D. Brown, “Artificial intelligence in the diagnosis of low back pain”, *The Orthopedic Clinics of North America*, 22(2):303–314, (1991)
- [62] Malek Adjouadi, Melvin Ayala, Mercedes Cabrerizo, Nuannuan Zong, Gabriel Lizarraga, and Mark Rossman, “Classification of leukemia blood samples using neural networks”, *Annals of Biomedical Engineering*, 38(4):1473–1482, (2010)
- [63] Nuannuan Zong, Malek Adjouadi, and Melvin Ayala, ”Optimizing the classification of acute lymphoblastic leukemia and acute myeloid leukemia samples using artificial neural networks”, *Biomedical Sciences Instrumentation*, 42:261–266, (2006)
- [64] Jonathan L. Jesneck, Loren W. Nolte, Jay A. Baker, Carey E. Floyd, and Joseph Y. Lo, “Optimized approach to decision fusion of heterogeneous data for breast cancer diagnosis”, *Medical Physics*, 33(8):2945–2954, (2006)
- [65] Rohit Dua, Daryl G. Beetner, William V. Stoecker, and Donald C. Wunsch, “Detection of basal cell carcinoma using electrical impedance and neural networks”, *IEEE Transactions on Bio-Medical Engineering*, 51(1):66–71, (2004)
- [66] Angelo Andriulli, Enzo Grossi, Massimo Buscema, Alberto Pilotto, Virginia Festa, and Francesco Perri, “Artificial neural networks can classify uninvestigated patients with dyspepsia”, *European Journal of Gastroenterology & Hepatology*, 19(12):1055–1058, (2007)
- [67] Bruno Annibale and Edith Lahner, “Assessing the severity of atrophic gastritis”, *European Journal of Gastroenterology & Hepatology*, 19(12):1059–1063, (2007)
- [68] J. S. Shang, Y. S. Lin, and A. M. Goetz, “Diagnosis of mrsa with neural networks and logistic regression approach”, *Health Care Management Science*, 3(4):287–297, (2000)
- [69] E. Pesonen, “Is neural network better than statistical methods in diagnosis of acute appendicitis?”, *Studies in Health Technology and Informatics*, 43 Pt A:377–381, (1997)
- [70] G. Rovetta, G. Bianchi, P. Monteforte, L. Buffrini, and G. Ghirardo, “Automated diagnosis and characterization of lyme disease using neural network analysis”, *The Journal of Rheumatology*, 22(3):571–572, (1995)
- [71] Enzo Grossi, Massimo P. Buscema, David Snowdon, and Piero Antuono, “Neuropathological findings processed by artificial neural networks (anns) can perfectly distinguish alzheimer’s patients from controls in the nun study”, *BMC Neurology*, 7:15+, (2007)
- [72] Orhan Er, Cengiz Sertkaya, Feyzullah Temurtas, and A. Cetin Tanrikulu, “A comparative study on chronic obstructive pulmonary and pneumonia diseases diagnosis using neural networks and artificial immune system”, *Journal of Medical Systems*, 33(6):485–492, (2009)
- [73] C. Lagor, D. Aronsky, M. Fiszman, and P. J. Haug, “Automatic identification of patients eligible for a pneumonia guideline: comparing the diagnostic accuracy of two decision support models”, *Studies in Health Technology and Informatics*, 84(Pt 1):493–497, (2001)
- [74] Yogender Aggarwal, Bhuwan Mohan M. Karan, Barda Nand N. Das, and Rakesh Kumar K. Sinha, “An unsupervised neural network to predict the level of heat stress”, *Journal of Clinical Monitoring and Computing*, 22(6):425–430, (2008)
- [75] Brachman RJ, Levesque HJ, “Knowledge representation and reasoning”, San Francisco, CA, USA: Elsevier/Morgan Kaufmann Publishers, Available from: (<http://www.webcitation.org/6NYodAVnC>). (2004)
- [76] Riaño D, Real F, López-Vallverdú JA, Campana F, Ercolani S, Mecocci P, et al, “An ontology-based personalization of healthcare knowledge to support clinical decisions for chronically ill patients”, *J Biomed Inform*, 45(3):429–46. Available from: <http://dx.doi.org/10.1016/j.jbi.2011.12.008>. (2012)
- [77] Farion K, Michalowski W, Wilk S, O’Sullivan D, Rubin S, Weiss D., “Clinical decision support system for point of care use ontology-driven design and software implementation”, *Methods Inf Med*, 48(4):381–90. Available from: (<http://www.ncbi.nlm.nih.gov/pubmed/19448882>). (2009)
- [78] Quaglini, S., Stefanelli, M., Cavallini, a., Micieli, G., Fassino, C., & Mossa, C., “Guideline-based careflow systems”, *Artificial Intelligence in Medicine*, 20, 5–22. doi:10.1016/S0933-3657(00)00050-6, (2000)

- [79] Jensen K., “Coloured Petri nets. Basic concepts, analysis methods and practical use”, In: EATCS Monographs on Theoretical Computer Science. Berlin: Springer – Verlag, (1992)
- [80] Choi J, Currie LM, Wang D, Bakken S., “Encoding a clinical practice guideline using guideline interchange format: a case study of a depression screening and management guideline”, *Int J Med Inform*;76 Suppl 2(null):S302–07. Available from: 10.1016/j.ijmedinf.2007.05.011, (2007)
- [81] Martínez-García A, Moreno-Conde A, Jódar-Sánchez F, Leal, S, Parra C., “Sharing clinical decisions for multimorbidity case management using social network and open-source tools”, *J Biomed Inform*;46(6):977–84. Available from: (<http://www.webcitation.org/6NX2irOgx>), (2013)
- [82] Goud, R., de Keizer, N. F., ter Riet, G., Wyatt, J. C., Hasman, A., Hellems, I. M., & Peek, N., “Effect of guideline based computerised decision support on decision making of multidisciplinary teams: cluster randomised trial in cardiac rehabilitation”, *BMJ (Clinical Research Ed.)*, 338, b1440. doi:10.1136/bmj.b1440, (2009)
- [83] Nguyễn HT, Walker EA., “A first course in fuzzy logic”, Boca Raton, FL, USA: Chapman & Hall/CRC. ([http://books.google.it/books?id=\\_IuuzgOJ0MgC](http://books.google.it/books?id=_IuuzgOJ0MgC)), (2006)
- [84] Esposito M, De Falco I, De Pietro G., “An evolutionary-fuzzy DSS for assessing health status in multiple sclerosis disease”, *Int J Med Inform*; 80(12):e245–54. Available from: 10.1016/j.ijmedinf.2011.09.003, (2011)
- [85] Mago VK, Mehta R, Woolrych R, Papageorgiou EI., “Supporting meningitis diagnosis amongst infants and children through the use of fuzzy cognitive mapping”, *BMC Med Inform Decis Mak*;12(1):98. Available from: (<http://www.biomedcentral.com/1472-6947/12/98>), (2012)
- [86] Card S., “The human–computer interaction handbook: fundamentals, evolving technologies, and emerging applications”, Boca Raton, FL, USA: New York, NY, USA: London, UK: Lawrence Erlbaum Assoc. Inc; Available from: (<http://www.webcitation.org/6NYny57Bu>), (2007)
- [87] Mane KK, Bizon C, Schmitt C, Owen P, Burchett B, Pietrobon R, et al., “VisualDecisionLinc: a visual analytics approach for comparative effectiveness-based clinical decision support in psychiatry”, *J Biomed Inform*;45(1):101–6. Available from: 10.1016/j.jbi.2011.09.003, (2012)
- [88] Han, J., Kamber, M., & Pei, J., “Data Mining: Concepts and Techniques (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, (2011)
- [89] Gorunescu, F., “Data mining concepts, models and techniques”, Berlin: Springer. Retrieved from <http://site.ebrary.com/id/10454853>, (2011)
- [90] Aggarwal CC, Zhai C, editors., “Mining text data”, Boston, MA: Springer US; (<http://www.springerlink.com/index/10.1007/978-1-4614-3223-4>), (2012)
- [91] Feldman, R., & Sanger, J., “The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Imagine (Vol. 34, p. 410). Retrieved from <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0521836573>, (2007)
- [92] Manning CD, Raghavan P, Schütze H., “Introduction to information retrieval”, Cambridge, UK: Cambridge University Press, (<http://books.google.it/books?id=t1PoSh4uwVcC>), (2008)
- [93] Jurafsky D, Martin JH., “Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition”, Upper Saddle River, New Jersey, USA: Pearson Prentice Hall/Pearson Education International, (2009)
- [94] O'Sullivan DM, Wilk SA, Michalowski WJ, Farion KJ., “Automatic indexing and retrieval of encounter-specific evidence for point-of-care support”, *J Biomed Inform. Elsevier Inc*;43(4):62331. Available from: (<http://www.ncbi.nlm.nih.gov/pubmed/20230908>), (2010)
- [95] Home – The Cochrane Library. Available from: (<http://www.thecochranelibrary.com/view/0/index.html>)
- [96] Bungum, L., & Gambäck, B., “Evolutionary Algorithms in Natural Language Processing. Norwegian Artificial Intelligence Symposium, (November), (2010)
- [97] Matheny ME, Fitzhenry F, Speroff T, Green JK, Griffith ML, Vasilevskis EE, et al., “Detection of infectious symptoms from VA emergency department and primary care clinical documentation”, *Int J Med Inform*, 81(3):143–56. Available from: <http://dx.doi.org/10.1016/j.ijmedinf.2011.11.005>, (2012)

- [98] SNOMED Clinical Terms (SNOMED CTs) [Internet]. U.S.National Library of Medicine; Available from: (<http://www.webcitation.org/6NYo9VKom>).
- [99] Rokach L., “Data mining with decision trees: theory and applications”, Hackensack, New Jersey, USA: London, UK: World Scientific Publishing Company, Incorporated, (<http://books.google.it/books?id=GIKIIR78OxkC>), (2007)
- [100] Ji S-Y, Smith R, Huynh T, Najarian K., “A comparative analysis of multi-level computer-assisted decision making systems for traumatic injuries”, *BMC Med Inform Decis Mak*, 9(1):2. Available from: (<http://www.biomedcentral.com/1472-6947/9/2>), (2009)
- [101] Bergman LG, Fors UGH, Decision support in psychiatry – a comparison between the diagnostic outcomes using a computerized decision support system versus manual diagnosis. *BMC Med Inform Decis Mak*, 8(1):9. Available from: (<http://www.biomedcentral.com/147-6947/8/9>), (2008)
- [102] American Psychiatric Association., “Diagnostic and statistical manual of mental disorders”, (text revision). 4th ed. (2000)
- [103] Sivia DS, Skilling J., “Data analysis: a Bayesian tutorial”, Oxford, UK: Oxford University Press, (<http://books.google.it/books?id=6O8ZAQAIAAJ>), (2006)
- [104] Elkin PL, Liebow M, Bauer BA, Chaliki S, Wahner-Roedler D, Bundrick J, et al., “The introduction of a diagnostic decision support system (DXplain™) into the workflow of a teaching hospital service can decrease the cost of service for diagnostically challenging Diagnostic Related Groups (DRGs)”, *Int J Med Inform*, 79(11):772–7. Available from: (<http://www.sciencedirect.com/science/article/pii/S1386505610001620>), (2010)
- [105] Sadeghi S, Barzi A, Sadeghi N, King B., “A Bayesian model for triage decision support”, *Int J Med Inform*. 75(5):403–411. Available from: (<http://www.ncbi.nlm.nih.gov/pubmed/16140572>), (2006)
- [106] Boussaïd, I., Lepagnot, J., & Siarry, P., “A survey on optimization metaheuristics”, *Information Sciences*, 237, 82–117. doi:10.1016/j.ins.2013.02.041, (2013)
- [107] S. Kirkpatrick, C. Gelatt, M. Vecchi, “Optimization by simulated annealing”, *Science* 220 671–680, (1983)
- [108] F. Glover, “Future paths for integer programming and links to artificial intelligence”, *Computers and Operations Research* 13, 533–549, (1986)
- [109] J.D. Farmer, N.H. Packard, A.S. Perelson, “The immune system, adaptation, and machine learning”, *Physica D* 2, 187–204, (1986)
- [110] J.R. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)”, first ed., The MIT Press, (1992)
- [111] D.E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine learning”, first ed., *Studies in Computational Intelligence*, Addison-Wesley Longman Publishing Co., Inc., (1989)
- [112] M. Dorigo, “Optimization, Learning and Natural Algorithms”, Ph.D. Thesis, Politecnico di Milano, Italy, (1992)
- [113] R.G. Reynolds, T.A. Kohler, Z. Kobti, “The effects of generalized reciprocal exchange on the resilience of social networks: an example from the prehispanic mesa verde region”, *Computational and Mathematical Organization Theory* 9, 227–254, (2003)
- [114] J. Kennedy, R. Eberhart, “Particle swarm optimization”, *IEEE International Conference on Neural Networks* 4, 1942–1948, (1995)
- [115] N. Hansen, A. Ostermeier, A. Gawelczyk, “On the adaptation of arbitrary normal mutation distributions in evolution strategies: the generating set adaptation”, in: *Proceedings of the 6th International Conference on Genetic Algorithms*, Morgan Kaufman Publishers Inc., San Francisco, CA, USA, pp. 57–64, (1995)
- [116] H. Mühlenbein, G. Paaß, “From recombination of genes to the estimation of distributions I. Binary parameters”, in: *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, PPSN IV*, Springer-Verlag, London, UK, pp. 178–187, (1996)
- [117] R.M. Storn, K.V. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization* 11, 341–359, (1997)

- [118] K.M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control”, *IEEE Control Systems Magazine* 22, 52–67, (2002)
- [119] D. Simon, “Biogeography-based optimization”, *IEEE Transactions on Evolutionary Computation* 12, 702–713, (2008)
- [120] Yang, X., “Nature-Inspired Optimization Algorithms”, 1st Edition, Elsevier, (2014)
- [121] Gandomi AH, Alavi AH., “Krill herd: a new bio-inspired optimization algorithm”, *Commun Nonlinear Sci Numer Simul*, 17(12), (2012)
- [122] M. Birattari, L. Paquete, T. Stützle, K. Varrentrapp, “Classification of Metaheuristics and Design of Experiments for the Analysis of Components”, Technical Report AIDA-01-05, FG Intellektik, FB Informatik, Technische Universität Darmstadt, Darmstadt, Germany, (2001)
- [123] E.G. Talbi, “Metaheuristics: From Design to Implementation”, first ed., Wiley-Blackwell, (2009)
- [124] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, “Equation of state calculations by fast computing machines”, *Journal of Chemical Physics* 21, 1087–1090, (1953)
- [125] J. Courat, G. Raynaud, I. Mrad, P. Siarry, “Electronic component model minimization based on log simulated annealing”, *IEEE Transactions on Circuits and Systems: Part I* 41, 790–795, (1994)
- [126] N.E. Collins, R.W. Eglese, B. Golden, “Simulated annealing – an annotated bibliography”, *American Journal of Mathematical and Management Sciences*, 209–307, (1988)
- [127] M. Fleischer, “Simulated annealing: past, present and future”, in: *Proceedings of the 27th Conference on Winter Simulation*, Arlington, VA, USA, pp. 155–161, (1995)
- [128] C. Koulamas, S.R. Antony, R. Jaen, “A survey of simulated annealing applications to operations research problems”, *Omega* 22, 41–56, (1994)
- [129] P.V. Laarhoven, E. Aarts, “Simulated Annealing: Theory and Applications”, first ed., D. Reidel Publishing Company, 1987.
- [130] P. Larrañaga, J.A. Lozano (Eds.), “Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation”, Kluwer Academic Publishers, Boston, MA, (2002)
- [131] B. Suman, P. Kumar, “A survey of simulated annealing as a tool for single and multiobjective optimization”, *Journal of the Operational Research Society* 57, 1143–1160, (2006)
- [132] M. Creutz, “Microcanonical Monte Carlo simulation”, *Physical Review Letters* 50, 1411–1414, (1983)
- [133] A. Nakib, H. Oulhadj, P. Siarry, “Microscopic image segmentation with two-dimensional exponential entropy based on hybrid microcanonical annealing”, in: *IAPR Conference on Machine Vision Applications*, Tokyo, Japan, pp. 420–423, (2007)
- [134] G. Dueck, T. Scheuer, “Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing”, *Journal of Computational Physics* 90, 161–175, (1990)
- [135] J. Dréo, A. Pétrowski, P. Siarry, E. Taillard, “Metaheuristics for Hard Optimization: Methods and Case Studies”, Springer, (2006)
- [136] I. Charon, O. Hudry, “The noising method: a new method for combinatorial optimization”, *Operations Research Letters* 14, 133–137, (1993)
- [137] I. Charon, O. Hudry, “The noising methods: a generalization of some metaheuristics”, *European Journal of Operational Research* 135, 86–101, (2001)
- [138] I. Charon, O. Hudry, “Self-tuning of the noising methods”, *Optimization* 58, 1–21, (2009)
- [139] R. Battiti, G. Tecchiolli, “The reactive tabu search”, *ORSA Journal on Computing* 6, 126–140, (1994)
- [140] F. Glover, M. Laguna, “Tabu Search”, Kluwer Academic Publishers, (1997)
- [141] M. Gendreau, J.Y. Potvin, “Chapter 6: Tabu search”, in: E.K. Burke, G. Kendall (Eds.), *Search Methodologies*, Springer, pp. 165–186, (2006)
- [142] M. Gendreau, “Chapter 2: an introduction to tabu search”, in: F. Glover, G.A. Kochenberger (Eds.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, pp. 37–54, (2003)
- [143] R. Chelouah, P. Siarry, “Tabu search applied to global optimization”, *European Journal of Operational Research* 123, 256–270, (2000)
- [144] T.A. Feo, M.G.C. Resende, “A probabilistic heuristic for a computationally difficult set covering problem”, *Operations Research Letters* 8, 67–71, (1989)



- [145] T.A. Feo, M.G.C. Resende, “Greedy randomized adaptive search procedures”, *Journal of Global Optimization* 6, 109–133, (1995)
- [146] P. Festa, M. Resende, “An annotated bibliography of GRASP, Part I: algorithms”, *International Transactions in Operational Research* 16, 1–24, (2009)
- [147] P. Festa, M.G.C. Resende, “Hybridizations of GRASP with Path-Relinking”, Technical Report HYB-GPR-2011, AT&T Labs Research, (2011)
- [148] M.G.C. Resende, C.C. Ribeiro, “GRASP: Greedy Randomized Adaptive Search Procedures”, Technical Report SGRASP2010, AT&T Labs Research, (2010)
- [149] M.J. Hirsch, C.N. Meneses, P.M. Pardalos, M.G.C. Resende, “Global optimization by continuous GRASP”, *Optimization Letters* 1, 201–212, (2007)
- [150] N. Mladenovic, “A variable neighborhood algorithm – a new metaheuristic for combinatorial optimization”, in: *Abstracts of Papers Presented at Optimization Days, Montréal, Canada*, (1995)
- [151] N. Mladenovic, P. Hansen, “Variable neighborhood search”, *Computers and Operations Research* 24, (1997)
- [152] C. Blum, A. Roli, “Metaheuristics in combinatorial optimization: overview and conceptual comparison”, *ACM Computing Surveys* 35, 268–308, (2003)
- [153] P. Hansen, N. Mladenovic, J.A.M. Pérez, “Variable neighbourhood search: methods and applications”, *4OR* 6, 319–360, (2008)
- [154] P. Hansen, N. Mladenovic, J.A.M. Pérez, “Variable neighbourhood search: algorithms and applications”, *Annals of Operations Research* 175, 367–407, (2010)
- [155] J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, N. Velasco, “GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots”, *Engineering Applications of Artificial Intelligence* 23, 780–794, (2010)
- [156] A. Salehipour, K. Sörensen, P. Goos, O. Bräysy, “Efficient GRASP + VND and GRASP + VNS metaheuristics for the traveling repairman problem”, *4OR* 9, 189–209, (2011)
- [157] R. Battiti, M. Brunato, F. Mascia, “Reactive Search and Intelligent Optimization”, Springer, (2008)
- [158] Aloise DJ, Aloise D, Rocha CTM, Ribeiro CC, Ribeiro JC, Moura LSS, “Scheduling workover rigs for onshore oil production”. *Discrete Appl Math* 154(5):695–702, (2006)
- [159] M. F. Santos, W. Mathew, T. Kovacs, H. Santos, “Supervised learning classifier systems for grid data mining”, In *Proceedings of the international conference on Computational and information science (CIS'09)*, Vasilis Zafiris, Maria Benavides, Kun Gao, Shohreh Hashemi, Katarina Jegdic, Guennadi A. Kouzaev, Plamen Simeonov, Luige Vladareanu, and Carol Vobach (Eds.)(WSEAS), Stevens Point, Wisconsin, USA, 416–424, (2009)
- [160] C. Voudouris, “Guided Local Search for Combinatorial Optimization Problems”, Ph.D Thesis, University of Essex, (1997)
- [161] C. Voudouris, E. Tsang, “Guided local search”, *European Journal of Operational Research* 113, 469–499, (1999)
- [162] C. Voudouris, “Guided local search: an illustrative example in function optimization”, *BT Technology Journal* 16, 46–50, (1998)
- [163] L.T. Lau, “Guided Genetic Algorithm”, Ph.D thesis, University of Essex, (1999)
- [164] Q. Zhang, J. Sun, E. Tsang, J. Ford, “Combination of guided local search and estimation of distribution algorithm for quadratic assignment problem”, in: *Proceedings of the Genetic and Evolutionary Computation Conference, Chicago, IL, USA*, pp. 42–48, (2003)
- [165] N. Tairan, Q. Zhang, “Population-based guided local search: some preliminary experimental results”, in: *Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain*, pp. 1–5, (2010)
- [166] C. Voudouris, E.P.K. Tsang, A. Alsheddy, “Guided local search”, in: J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, J.C. Smith (Eds.), *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley & Sons, (2010)
- [167] C. Voudouris, E.P.K. Tsang, A. Alsheddy, “Effective application of guided local search”, in: J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, J.C. Smith (Eds.), *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley & Sons, (2010)

- [168] T. Stützle, “Local Search Algorithms for Combinatorial Problems: Analysis, Improvements, and New Applications”, Ph.D. Thesis, Darmstadt University of Technology, (1998)
- [169] E.G. Talbi, “Metaheuristics: From Design to Implementation”, first ed., Wiley-Blackwell, (2009)
- [170] H.R. Lourenço, O. Martin, T. Stützle, “Chapter 12: iterated local search: framework and applications”, in: M. Gendreau, Y. Potvin (Eds.), *Handbook of Metaheuristics*, second ed., Springer, pp. 363–397, (2010)
- [171] T. Bäck, “Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms”, Oxford University Press, Oxford, UK, (1996)
- [172] T. Bäck, H.P. Schwefel, “An overview of evolutionary algorithms for parameter optimization”, *Evolutionary Computation* 1, 1–23, (1993)
- [173] L. Bianchi, M. Dorigo, L.M. Gambardella, W.J. Gutjahr, “A survey on metaheuristics for stochastic combinatorial optimization”, *Natural Computing* 8, 239–287, (2009)
- [174] C. Blum, A. Roli, “Metaheuristics in combinatorial optimization: overview and conceptual comparison”, *ACM Computing Surveys* 35, 268–308, (2003)
- [175] C. Coello, G.B. Lamont, D.A. Van Veldhuizen, “Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)”, Springer Verlag New York, Inc., Secaucus, NJ, USA, (2006)
- [176] T. Bäck, H.P. Schwefel, “An overview of evolutionary algorithms for parameter optimization”, *Evolutionary Computation* 1, 1–23, (1993)
- [177] D.E. Goldberg, K. Deb, “A comparative analysis of selection schemes used in genetic algorithms”, in: *Foundations of Genetic Algorithms*, Morgan Kaufman, pp. 69–93, (1991)
- [178] T. Blickle, L. Thiele, “A comparison of selection schemes used in genetic algorithms”, *Evolutionary Computation* 4, 311–347, (1995)
- [179] D.E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine learning”, first ed., *Studies in Computational Intelligence*, Addison-Wesley Longman Publishing Co., Inc., (1989)
- [180] D. Beasley, D. Bull, R.R. Martin, “An overview of genetic algorithms. Part i: fundamentals”, *University Computing* 15, 58–69, (1993)
- [181] H. Santos, M. F. Santos, W. Mathew, “Supervised Learning Classifier System for Grid Data Mining”, *New Fundamental Technologies in Data Mining*, Prof. Kimito Funatsu (Ed.), ISBN: 978-953-307-547-1, InTech, DOI: 10.5772/14199, (2011)
- [182] A. Sinha, D. Glodberg, “A Survey of Hybrid Genetic and Evolutionary algorithms”, Technical Report 2003004, Illinois Genetic Algorithms Laboratory (IlligAL), (2003)
- [183] A. Konak, D. Coit, A. Smith, “Multi-objective optimization using genetic algorithms: a tutorial”, *Reliability Engineering & System Safety in Special Issue –Genetic Algorithms and Reliability* 92, 992–1007, (2006)
- [184] I. Rechenberg, “Cybernetic Solution Path of an Experimental Problem”, Technical Report, Royal Air Force Establishment, (1965)
- [185] Rechenberg, “Evolutions strategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution”, Frommann-Holzboog, Stuttgart, (1973)
- [186] Beyer, H.-G., Beyer, H.-G., Schwefel, H.-P., & Schwefel, H.-P., “Evolution strategies – A comprehensive introduction”, *Natural Computing*, 1, 3 – 52. doi:10.1023/A:1015059928466, (2002)
- [187] N. Hansen, A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies”, *Evolutionary Computation* 9, 159–195, (2001)
- [188] N. Hansen, “The CMA evolution strategy: a comparing review”, in: J. Lozano, P. Larranaga, I. Inza, E. Bengoetxea (Eds.), *Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms*, Springer, pp. 75–102, (2006)
- [189] A. Auger, N. Hansen, “A restart CMA evolution strategy with increasing population size”, in: B. McKay, et al. (Eds.), *IEEE International Congress on Evolutionary Computation (CEC’05)*, vol. 2, pp. 1769–1776, (2005)
- [190] L.J. Fogel, A.J. Owens, M.J. Walsh, “Artificial Intelligence through Simulated Evolution”, John Wiley, New York, USA, (1966)

- [191] D.B. Fogel, “System Identification through Simulated Evolution: A Machine Learning Approach to Modeling”, Ginn Press, (1991)
- [192] D.B. Fogel, “Evolutionary Computation: Toward a New Philosophy of Machine Intelligence”, IEEE Press, Piscataway, NJ, USA, (1995)
- [193] McCarthy, J., Abrahams, P. W., Edwards, D. J., Hart, T. P., & Levin, M. I., “LISP 1.5 Programmer’s Manual”, 1–116. Retrieved from [http://www.softwarepreservation.org/projects/LISP/book/LISP\\_1.5\\_Programmers\\_Manual.pdf](http://www.softwarepreservation.org/projects/LISP/book/LISP_1.5_Programmers_Manual.pdf) [npapers2://publication/uuid/84B30846-C821-4CCE-B664-23AAE0EEE135, (2005)
- [194] R. Poli, W.B. Langdon, N.F. McPhee, “A Field Guide to Genetic Programming”, Lulu Enterprises, UK Ltd., (2008)
- [195] J.R. Koza, “Introduction to genetic programming”, in: K.E. Kinnear Jr. (Ed.), *Advances in Genetic Programming*, MIT Press, Cambridge, MA, USA, (1994)
- [196] Y. Shan, R.I. McKay, D. Essam, H.A. Abbass, “A survey of probabilistic model building genetic programming”, in: *Scalable Optimization via Probabilistic Modeling*, *Studies in Computational Intelligence*, vol. 33, Springer, pp. 121–160, (2006)
- [197] B.L. William, P. Riccardo, F.M. Nicholas, R.K. John, “Genetic programming: an introduction and tutorial, with a survey of techniques and applications”, in: J. Fulcher, L.C. Jain (Eds.), *Computational Intelligence: A Compendium*, *Studies in Computational Intelligence (SCI)*, vol. 115, Springer-Verlag, pp. 927–1028, (2008)
- [198] R.I. McKay, N.X. Hoai, P.A. Whigham, Y. Shan, M. O’Neill, “Grammar-based genetic programming: a survey”, *Genetic Programming and Evolvable Machines* 11, 365–396, (2010)
- [199] J.H. Holland, “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, MI, USA, (1975)
- [200] Sutton, R. and Barto, A., “Reinforcement Learning: an introduction”, Cambridge, MA: The MIT Press/Bradford Books, (1998)
- [201] Bernad, E., “Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks”, 11(3), 209–238, (2003)
- [202] S. Baluja, “Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning”, Technical Report, Carnegie Mellon University, Pittsburgh, PA, USA, (1994)
- [203] G. Harik, F. Lobo, D. Goldberg, “The compact genetic algorithm”, in: *Proceedings of the IEEE Conference on Evolutionary Computation*, (1998)
- [204] J. De Bonet, C. Isbell, P. Viola, “Mimic: finding optima by estimating probability densities”, in: *NIPS*, (1996)
- [205] S. Baluja, S. Davies, “Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space”, in: D.H. Fisher (Ed.), *14th International Conference on Machine Learning*, Morgan Kaufmann, (1997)
- [206] M. Pelikan, H. Mühlenbein, “The bivariate marginal distribution algorithm”, in: R. Roy, T. Furuhashi, P.K. Chawdhry (Eds.), *Advances in Soft Computing – Engineering Design and Manufacturing*, Springer-Verlag, London, (1999)
- [207] P. Larrañaga, J.A. Lozano (Eds.), “Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation”, Kluwer Academic Publishers, Boston, MA, (2002)
- [208] M. Pelikan, D.E. Goldberg, E. Cant-Paz, “BOA: the Bayesian optimization algorithm”, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99 (Orlando, FL)*, vol. I, Morgan-Kaufmann Publishers, San Fransisco, CA, (1999)
- [209] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, “Linkage problem, distribution estimation, and bayesian networks”, *Evolutionary Computation* 8, (2000)
- [210] R. Etxeberria, P. Larrañaga, “Global Optimization Using Bayesian Networks”, in: A.A.O. Rodriguez, M.R.S. Ortiz, R.S. Hermida (Eds.), *CIMAF 99, Second Symposium on Artificial Intelligence, Adaptive Systems*, La Habana, (1999)
- [211] H. Mühlenbein, T. Mahnig, “The factorized distribution algorithm for additively decomposed

- functions”, in: A. Ochoa, M.R. Soto, R. Santana (Eds.), *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, Habana, Cuba, (1999)
- [212] G. Harik, “Linkage Learning via Probabilistic Modeling in the EcGA”, IlliGAL Report 99010, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, (1999)
- [213] M. Soto, A. Ochoa, S. Acid, L.M. de Campos, “Introducing the Polytrees Approximation of Distribution Algorithm”, in: R.S.E.A. Ochoa, M. Soto (Ed.), *Proceedings of the Second International Symposium on Artificial Intelligence, Adaptive Systems (International Conference CIMAF’99)*, (1999)
- [214] M. Hauschild, M. Pelikan, “An introduction and survey of estimation of distribution algorithms”, *Swarm and Evolutionary Computation* 1, 111–128, (2011)
- [215] W.D. Hillis, “Co-evolving parasites improve simulated evolution as an optimization procedure”, *Physica D* 42, (1990)
- [216] K.O. Stanley, R. Miikkulainen, “Competitive coevolution through evolutionary complexification”, *Journal of Artificial Intelligence Research* 21, 63–100, (2004)
- [217] C.D. Rosin, R.K. Belew, “New methods for competitive coevolution”, *Evolution Computing* 5, 1–29, (1997)
- [218] J.B. Pollack, A.D. Blair, “Co-evolution in the successful learning of backgammon strategy”, *Machine Learning* 32, 225–240, (1998)
- [219] T. Kowaliw, N.N. Kharm, C. Jensen, H. Moghnieh, J. Yao, “Using competitive co-evolution to evolve better pattern recognisers”, *International Journal of Computational Intelligence and Applications* 5, 305–320, (2005)
- [220] K. Sims, “Evolving 3D morphology and behavior by competition”, *Artificial Life* 1, (1994)
- [221] M.A. Potter, K.A.D. Jong, “A cooperative coevolutionary approach to function optimization”, in: *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, PPSN III*, Springer-Verlag, London, UK, (1994)
- [222] R.P. Wiegand, “An Analysis of Cooperative Coevolutionary Algorithms”, Ph.D. Thesis, George Mason University, Fairfax, VA, USA, (2004)
- [223] M.A. Potter, K.A. De Jong, “Cooperative coevolution: an architecture for evolving coadapted subcomponents”, *Evolutionary Computation* 8, (2000)
- [224] Q. He, L. Wang, “An effective co-evolutionary particle swarm optimization for constrained engineering design problems”, *Engineering Applications of Artificial Intelligence* 20, 89–99, (2007)
- [225] C.K. Goh, K.C. Tan, “A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization”, *Transactions on Evolutionary Computation* 13, 103–127, (2009)
- [226] E.B. Tylor, “*Primitive Culture*”, vol. 2, seventh ed., New York: Brentano’s, (1924)
- [227] R.G. Reynolds, “An introduction to cultural algorithms”, in: A.V. Sebalk, L.J. Fogel (Eds.), *Proceedings of the Third Annual conference on Evolutionary Programming*, World Scientific Publishing, River Edge, NJ, (1994)
- [228] E.G. Talbi, “*Metaheuristics: From Design to Implementation*”, first ed., Wiley-Blackwell, (2009)
- [229] C.A. Coello Coello, R.L. Becerra, “Adding knowledge and efficient data structures to evolutionary programming: a cultural algorithm for constrained optimization”, in: *GECCO*, pp. 201–209, (2002)
- [230] S.M. Saleem, “*Knowledge-Based Solution to Dynamic Optimization Problems using Cultural Algorithms*”, Ph.D. Thesis, Wayne State University, Detroit, MI, USA, (2001)
- [231] R.G. Reynolds, “*Cultural Algorithms: Theory and Applications*, in: *New Ideas in Optimization*”, McGraw-Hill Ltd., UK, Maidenhead, UK, England, pp. 367–378, (1999)
- [232] R.G. Reynolds, “An adaptive computer model of plant collection and early agriculture in the eastern valley of Oaxaca”, *Guila Naquitz: Archaic Foraging and Early Agriculture in Oaxaca, Mexico*, (1986)
- [233] D.C. Rivera, R.L. Becerra, A. Coello Coello, Carlos, “Cultural algorithms, an alternative heuristic to solve the job shop scheduling problem”, *Engineering Optimization* 39, 69–85, (2007)
- [234] N. Rychtyckyj, R.G. Reynolds, “Using cultural algorithms to re-engineer large-scale semantic networks”, *International Journal of Software Engineering and Knowledge Engineering* 15, 665–694, (2005)
- [235] A. Ochoa, J. Ponce, A. Hernández, L. Li, “Resolution of a combinatorial problem using cultural

- algorithms”, *JCP* 4, 738–741, (2009)
- [236] R.G. Reynolds, D. Liu, “Multi-objective cultural algorithms”, in: *IEEE Congress on Evolutionary Computation*, pp. 1233–1241, (2011)
- [237] R.G. Reynolds, B. Peng, M.Z. Ali, “The role of culture in the emergence of decision-making roles: an example using cultural algorithms”, *Complexity* 13, 27–42, (2008)
- [238] F. Glover, “A template for scatter search and path relinking”, *Lecture Notes on Computer Science* 1363, (1997)
- [239] M. Laguna, R. Marti, “Scatter Search: Methodology and Implementations in C”, Kluwer Academic Publishers, Norwell, MA, USA, (2002)
- [240] F. Glover, M. Laguna, R. Marti, “Scatter search and path relinking: advances and applications”, in: F. Glover, G. Kochenberger (Eds.), *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, vol. 57, Springer, New York, pp. 1–35, (2003)
- [241] R.M. Storn, K.V. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization* 11, (1997)
- [242] E. Mezura-Montes, M. Reyes-Sierra, C. Coello Coello, “Multi-objective optimization using differential evolution: a survey of the state-of-the-art”, in: U. Chakraborty (Ed.), *Advances in Differential Evolution*, Studies in Computational Intelligence, vol. 143, Springer, Berlin, pp. 173–196, (2008)
- [243] U. Chakraborty, “*Advances in Differential Evolution*”, first ed., Springer Publishing Company, (2008)
- [244] F. Neri, V. Tirronen, “Recent advances in differential evolution: a survey and experimental analysis”, *Artificial Intelligence Review* 33, 61–106, (2010)
- [245] S. Das, P.N. Suganthan, “Differential evolution: a survey of the state-of-the-art”, *IEEE Transactions on Evolutionary Computation* 15, (2011)
- [246] A. Engelbrecht, “*Fundamentals of Computational Swarm Intelligence*”, Wiley, (2006)
- [247] M. Dorigo, V. Maniezzo, A. Coloni, “Positive Feedback as a Search Strategy”, Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, (1991)
- [248] M. Dorigo, “*Optimization, Learning and Natural Algorithms*”, Ph.D. Thesis, Politecnico di Milano, Italy, (1992)
- [249] M. Dorigo, V. Maniezzo, A. Coloni, “The ant system: optimization by a colony of cooperating agents”, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 26, (1996)
- [250] M. Dorigo, C. Blum, “Ant colony optimization theory: a survey”, *Theoretical Computer Science* 344, 243–278, (2005)
- [251] M. Dorigo, M. Birattari, T. Stützle, U. Libre, D. Bruxelles, A.F.D. Roosevelt, “Ant colony optimization – artificial ants as a computational intelligence technique”, *IEEE Computational Intelligence Magazine* 1, 28–39, (2006)
- [252] M. Dorigo, S. Thomas, “Ant colony optimization: overview and recent advances”, in: M. Gendreau, J.Y. Potvin (Eds.), *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Springer, US, pp. 227–263, (2010)
- [253] M. Dorigo, T. Stützle, “The ant colony optimization metaheuristic: algorithms, applications, and advances”, in: F. Glover, G. Kochenberger (Eds.), *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, vol. 57, Springer, New York, pp. 250–285, (2003)
- [254] C. Blum, “Ant colony optimization: Introduction and recent trends”, *Physics of Life Reviews* 2, 353–373, (2005)
- [255] D. Angus, C. Woodward, “Multiple objective ant colony optimization”, *Swarm Intelligence* 3, 69–85, (2009)
- [256] P. Angeline, “Evolutionary optimization versus particle swarm optimization: philosophy and performance differences”, in: V. Porto, N. Saravanan, D. Waagen, A. Eiben (Eds.), *Evolutionary Programming VII*, Lecture Notes in Computer Science, vol. 1447, Springer, Berlin, Heidelberg, (1998)
- [257] J. Kennedy, R. Eberhart, Y. Shi, “*Swarm Intelligence*”, Morgan Kaufman, San Francisco, (2001)
- [258] E. Ozcan, C.K. Mohan, “Particle swarm optimization: surfing the waves”, in: *Proceedings of the IEEE Congress on Evolutionary Computation – CEC*, (1999)

- [259] M. Clerc, J. Kennedy, “The particle swarm – explosion, stability, and convergence in a multidimensional complex space”, *IEEE Transactions on Evolutionary Computation* 6, 58–73, (2002)
- [260] F. Vandenbergh, A. Engelbrecht, “A study of particle swarm optimization particle trajectories”, *Information Sciences* 176, 937–971, (2006)
- [261] D.S. Johnson, L.A. McGeoch, “The travelling salesman problem: a case study in local optimization”, in: E.H.L. Aarts, J.K. Lenstra (Eds.), *Local Search in Combinatorial Optimization*, John Wiley & Sons, (1997)
- [262] T. Blackwell, “Particle swarm optimization in dynamic environments”, in: S. Yang, Y.S. Ong, Y. Jin (Eds.), *Evolutionary Computation in Dynamic and Uncertain Environments*, *Studies in Computational Intelligence*, vol. 51, Springer, Berlin, Heidelberg, pp. 29–49, (2007)
- [263] A. Banks, J. Vincent, C. Anyakoha, “A review of particle swarm optimization. Part i: background and development”, *Natural Computing* 6, 467–484, doi:10.1007/s11047-007-9049-5, (2007)
- [264] M. Pant, R. Thangaraj, A. Abraham, “Particle swarm optimization: performance tuning and empirical analysis”, in: A. Abraham, A. Hassanien, P. Siarry, A. Engelbrecht (Eds.), *Foundations of Computational Intelligence*, vol. 3, Springer-Verlag, Berlin, Heidelberg, New York, NY, USA, pp. 101–128, (2009)
- [265] K.M. Passino, “Bacterial foraging optimization”, *International Journal of Swarm Intelligence Research* 1, 1–16, (2010)
- [266] S. Das, A. Biswas, S. Dasgupta, A. Abraham, “Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications”, in: A. Abraham, A.E. Hassanien, P. Siarry, A. Engelbrecht (Eds.), *Foundations of Computational Intelligence*, *Studies in Computational Intelligence*, vol. 3, Springer, Berlin, Heidelberg, pp. 23–55, (2009)
- [267] D. Karaboga, B. Akay, “A survey: algorithms simulating bee swarm intelligence”, *Artificial Intelligence Review* 31, 61–85, (2009)
- [268] P. Lucic, D. Teodorovic, “Computing with bees: attacking complex transportation engineering problems”, *International Journal on Artificial Intelligence Tools* 12, 375–394, (2003)
- [269] D. Karaboga, “An Idea Based on Honey Bee Swarm for Numerical Optimization”, Technical Report TR06, Erciyes University, (2005)
- [270] X.S. Yang, “Engineering optimizations via nature-inspired virtual bee algorithms”, in: *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach: First International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2005*, volume 3562 of *Lecture Notes in Computer Science*, Springer, pp. 317–323, (2005)
- [271] S. Sadik, A. Ali, H.F. Ahmad, H. Suguri, “Using honey bee teamwork strategy in software agents”, in: *CSCWD’06: 10th International Conference on Computer Supported Cooperative Work in Design*, pp. 1–6, (2006)
- [272] A. Gupta, N. Koul, “Swan: a swarm intelligence based framework for network management of ip networks”, *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, Vol. 01, IEEE Computer Society, Washington, DC, USA, pp. 114–118, (2007)
- [273] S. Nakrani, C. Tovey, “On honey bees and dynamic server allocation in internet hosting centers”, *Adaptive Behavior – Animals, Animats, Software Agents, Robots, Adaptive Systems* 12, 223–240, (2004)
- [274] D. Karaboga, B. Akay, “A survey: algorithms simulating bee swarm intelligence”, *Artificial Intelligence Review* 31, 61–85, (2009)
- [275] J.D. Farmer, N.H. Packard, A.S. Perelson, “The immune system, adaptation, and machine learning”, *Physica D* 2, (1986)
- [276] D. Dasgupta, S. Yu, F. Nino, “Recent advances in artificial immune systems: models and applications”, *Applied Soft Computing* 11, 1574–1587, (2011)
- [277] J. Timmis, P. Andrews, N. Owens, E. Clark, “An interdisciplinary perspective on artificial immune systems”, *Evolutionary Intelligence* 1, 5–26, (2008)
- [278] L.N. de Castro, “*Artificial Immune Systems: A New Computational Intelligence Approach*”, Springer-Verlag, London, (2002)

- [279] D. Simon, "Biogeography-based optimization", *IEEE Transactions on Evolutionary Computation* 12, 702–713, (2008)
- [280] R. MacArthur, E. Wilson, "The Theory of Biogeography", Princeton University Press, Princeton, NJ, (1967)
- [281] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization", *Information Sciences* 180, 3444–3464, (2010)
- [282] Yang XS. "A new metaheuristic bat-inspired algorithm". In: Cruz C, González JR, Pelta DA, Terrazas G, editors. *Nature inspired cooperative strategies for optimization (NISCO 2010)*. Studies in computational intelligence. Berlin, Germany: Springer, (2010)
- [283] Khan, K., & Sahai, A., "A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context", *International Journal of Intelligent Systems and Applications*, 4(7), 23–29. <http://doi.org/10.5815/ijisa.2012.07.03>, (2012)
- [284] Lewis SM, Cratsley CK. "Flash signal evolution, mate choice and predation in fireflies". *Ann Rev Entomol*, 53(2):293–321, (2008)
- [285] H. Banati, M. Bajaj, "Firey based feature selection approach", *IJCSI International Journal of Computer Science Issues* 8 (4), 473-480, (2011)
- [286] Fister, I., Yang, X. S., & Brest, J., "A comprehensive review of firefly algorithms", *Swarm and Evolutionary Computation*, 13, 34–46. doi:10.1016/j.swevo.2013.06.001, (2013)
- [287] Wang, G., Guo, L., Wang, H., Duan, H., Liu, L., & Li, J., "Incorporating mutation scheme into krill herd algorithm for global numerical optimization". *Neural Computing and Applications*, 24, 853–871. doi:10.1007/s00521-012-1304-8, (2014)
- [288] Yang, X. S., & Deb, S., "Cuckoo search via Lévy flights", 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings, 210–214. <http://doi.org/10.1109/NABIC.2009.5393690>, (2009)
- [289] Payne RB, Sorenson MD, Klitz K., "The cuckoos". Oxford, UK: Oxford University Press, (2005)
- [290] Gandomi AH, Yang XS, Alavi AH., "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems". *Eng Comput*, 29(1):17–35, (2013)
- [291] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera., "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework". *Journal of Multiple-Valued Logic and Soft Computing* 17:2-3, 255-287, (2011)
- [292] Rangra, K., "Comparative Study of Data Mining Tools", 4(6), 216–223, (2014)
- [293] Oliveira, S., Portela, F., Santos, M. F., Neves, J., Silva, Á., Rua, F., "Feature selection for detecting patients with weaning failures in Intensive Medicine", *Mathematics and Computers in Sciences and Industry*. Volume 50, pp 195-200. ISBN: 978-1-61804-327-6. CPS. (2015)
- [294] T.G. Dietterich., "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms", *Neural Computation* 10:7, (1998)
- [295] D.R. Carvalho, A.A. Freitas., "A hybrid decision tree/genetic algorithm method for data mining", *Information Sciences* 163:1, 13-35, (2004)
- [296] E. Bernadó-Mansilla, J.M. Garrell., "Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks", *Evolutionary Computation* 11:3, 209-238, (2003)
- [297] J. Wang, P. Neskovic, L.N. Cooper., "Improving nearest neighbor rule with a simple adaptative distance measure", *Pattern Recognition Letters* 28, 207-213, (2007)
- [298] D. Sheskin., "Handbook of parametric and nonparametric statistical procedures". Chapman and Hall/CRC, (2003)
- [299] K. Doksum., "Robust procedures for some linear models with one observation per cell", *Annals of Mathematical Statistics* 38, (1967)
- [300] M. F. Santos, W. Mathew, T. Kovacs, H. Santos, "Grid data mining by means of learning classifier systems and distributed model induction", In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (GECCO '11)*, Natalio Krasnogor (Ed.). ACM, New York, NY, USA, 391-398. DOI=<http://dx.doi.org/10.1145/2001858.2002024>, (2011)

## Apêndice 1 Experiências efetuadas

Tabela 9 – Demonstração dos algoritmos testados

<b>Evolutionary Fuzzy Rule Learning</b>			
<b>Acrónimo do algoritmo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
NSLV	2009	X	92.89
SGERD	2008	X	92.87
SLAVE2	2001	X	92.85
GFS-LogitBoost	2006	X	92.77
GFS-MaxLogitBoost	2007	X	92.89
GFS-Adaboost	2004	X	92.06
GFS-GP	2001	X	92.87
GFS-GPG	2001	X	92.88
IVTURS	2013	X	S/R
GFS-SP	2001	X	S/R
<b>Evolutionary Crisp Rule Learning</b>			
<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
UCS	2003	X	93.10
LDWPSO	2004	X	86.73
PSO_ACO	2004	X	S/R
OIGA	2004	X	S/R
REPSO	2004	X	92.89
Advanced_Ant_Miner	2002	N	
Advanced_Ant_Miner_Plus	2002	N	
Ant_Miner	2002	N	
Ant_Miner_Plus	2002	N	
BioHel	2009	X	92.93
Bojarczuk	2004	X	31.54
CORE	2006	X	92.87
DMEL	2003	N	
Falco_GP	2002	X	92.83
GAssist-ADI	2003	X	92.55



PSO_ACO	2004	X	S/R
<b>Lazy Learning</b>			
<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
KNNAdaptive	2007	X	92.97
CamNN	2006	X	92.46
CenterNN	2007	X	91.02
PW	2006	X	91.33
<b>Prototype Generation</b>			
<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
SGP	2007	X	57.47
PSCSA	2008	X	92.66
POC-NN	2005	X	56.35
MixtGauss	2006	X	92.46
LVQPRU	2005	X	91.10
DROP3PSO	2011	X	91.71
AMPSO	2007	X	S/R
<b>Fuzzy Instance Based Learning</b>			
<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
CFKNN	2011	X	45.56
FRNN_FRS	2011	X	89.90
FRNN_VQRS	2011	X	89.87
FRNN	2007	X	92.89
PFKNN	2010	X	S/R
<b>Crisp Rule Learning</b>			
<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
ART	2004	N	
DataSqueezer	2006	N	
Riona	2002	X	S/R
Rules6	2005	N	
SRI	2006	N	

**Árvores de Decisão**

<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
Target	2008	X	92.89
AdaBoost.NC	2012	X	1.09
C45_Binarization	2011	X	8.96
DT_GA	2004	X	93.20
DT_Oblique	2003	X	90.56
FunctionalTrees	2004	N	
PUBLIC	2000	X	93.19

**Redes Neurais Artificiais**

<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
Ensemble	2007	N	
EvRBFN	2004	X	92.89
IRProp+	2003	X	92.89
LVQ	2001	X	86.71

**Evolurionary Prototype Selection**

<b>Acrónimo</b>	<b>Ano</b>	<b>Funcionou</b>	<b>% de acerto</b>
CoCoIS	2010	X	S/R
GA_MSE_CC_FSM	2007	X	S/R
EFS_RPS	2013	X	S/R
EIS_RFS	2012	X	91.5
IFS_COOCO	2010	X	S/R

## Apêndice 2 Publicação Científica

### Machine-Learning - An overview of optimization techniques.

Autores: Pedro Oliveira, Filipe Portela, Manuel Filipe Santos, José Machado, António Abelha.

Conferência. Computers - 19th International Conference on Circuits, Systems, Communications and Computers - Intelligent Systems and Applications Special Sessions.

Abstract— In an intelligent system the tasks roles are an essential play between learning and optimization. The Machine Learning is used to address a specific problem. However, the optimization of these systems are particularly difficult to apply due to the dynamic, complex and multidisciplinary nature. Nowadays we notice a constant research and development of new algorithms capable of extracting knowledge treated large volumes of data, thus obtaining better predictive results than current algorithms. There emerges and a large group of techniques and models that are best suited to the nature and complexity of the problem. It is in this regard that incorporates this work. The aim of this work is to presente an overview of the most recent and most used optimization techniques in machine learning.

Keywords—Machine Learning, Optimization techniques, Literature review

## Apêndice 3 Publicação Científica

### Optimization techniques to detect early ventilation extubation in Intensive Care Units

Autores: Pedro Oliveira, Filipe Portela, Manuel Filipe Santos, José Machado, António Abelha, Álvaro Silva e Fernando Rua Martins

Conferência: 4th World Conference on Information Systems and Technologies

Abstract— The decision support models in intensive care units are developed to support medical staff in their decision making process. However, the optimization of these models is particularly difficult to apply due to the dynamic, complex and multidisciplinary nature. Thus, there is a constant research and development of new algorithms capable of extracting knowledge from large volumes of data, in order to obtain better predictive results than the current algorithms. To test the optimization techniques a case study with real data provided by INTCare project was explored. The data provided concerns extubation cases only. In this dataset, several models like Evolutionary Fuzzy Rule Learning, Lazy Learning, Decision Trees and many others were analysed in order to detect early extubation. The hybrids Decision Trees Genetic Algorithm, Supervised Classifier System and KNNAdaptive obtained the most accurate rate 93.2%, 93.1%, 92.97% respectively, thus showing their feasibility to work in a real environment.

Keywords—Optimization techniques; Decision Support Systems; Machine Learning; Heuristics; Intensive Care Units Extubation