



Ana Filipa Silva Guarino

Programação Genética aplicada à previsão do consumo de combustível

Relatório de Dissertação

Mestrado em Engenharia Informática

orientada por Professor Doutor Fernando P. Machado e Professor Doutor Nuno Lourenço

e apresentada ao Departamento de Engenharia Informática

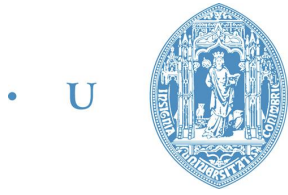
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Julho de 2018



UNIVERSIDADE DE COIMBRA

Esta página foi propositadamente deixada em branco.



• C •

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

ANA FILIPA SILVA GUARINO

Programação Genética aplicada à previsão do consumo de combustível

Mestrado em Engenharia Informática
Dissertação
Relatório Final

Orientadores:

Professor Doutor **Fernando P. Machado**

Professor Doutor **Nuno Lourenço**

Coimbra, 2018

Esta página foi propositadamente deixada em branco.

Agradecimentos

Gostaria de deixar uma palavra de agradecimento aos meus orientadores Professor Doutor Fernando Penousal Machado e Professor Doutor Nuno Lourenço, ao Professor Doutor Pedro Abreu e a todos os meus colegas e amigos do laboratório C.6.5 que sempre me acompanharam e incentivaram durante a realização deste trabalho. Gostaria também de agradecer à empresa Stratio Automotive pela oportunidade de trabalhar neste projeto. Por fim, mas não menos importante, gostaria de agradecer aos meus familiares, em especial aos meus pais e à minha irmã que, ao longo da vida, sempre me apoiaram e que sem eles nada disto teria sido possível.

Esta página foi propositadamente deixada em branco.

Abstract

The technological evolution of sensors and electronics has improved performance, reliability and robustness of transport vehicles for the carriage of goods and passengers, as well as the maintenance activities, particularly the proactive maintenance. These advances along with the expansion of computational learning allow the creation of new fleets' management approaches paving the way for the generation of new automated models that are able to predict the fuel consumption of vehicles.

The goal of this work was to study the usage of symbolic regressions, created through Genetic Programming, as a methodology for the automatic creation of a model that is capable of predicting the fuel consumption of heavy vehicles, using real data collected by sensors. In addition, a study was conducted to understand the influence of the meteorological factors on the prediction of the same values, including features as the atmospheric pressure and the air temperature.

The results obtained were able to prove the feasibility of the technique for this type of problems, as it was able to generate a model using only 25% of the variables from the original expression, created manually by the Stratio Automotive company, and with lower prediction errors. The inclusion of meteorological factors in the prediction models influenced the fuel consumption values, but the improvements in the models generated were not substantial.

Keywords

Evolutionary Algorithms (EA), Genetic Programming (GP), Symbolic Regression, Prediction, Fuel Consumption.

Esta página foi propositadamente deixada em branco.

Resumo

A evolução tecnológica dos sensores e da eletrônica melhorou o desempenho, a confiabilidade e a robustez dos veículos de transporte de mercadorias e de passageiros, assim como as atividades de manutenção, em especial, as de manutenção proativa. Estes avanços, em conjunto com a expansão da aprendizagem computacional possibilitam a criação de novas abordagens de gestão de frotas, abrindo caminho para a geração de novos modelos automatizados que consigam prever o consumo de combustível dos veículos.

O objetivo deste trabalho consistiu no estudo do uso da regressão simbólica com Programação Genética como metodologia para a criação automática de um modelo capaz de prever o consumo de combustível de veículos pesados, utilizando dados reais recolhidos por sensores. Além disso, foi analisada a influência que os fatores meteorológicos têm na previsão desse mesmo valor, entre eles a pressão atmosférica e a temperatura do ar.

Os resultados obtidos permitem-nos comprovar a viabilidade da técnica para este tipo de problemas, tendo sido capaz de gerar um modelo utilizando apenas 25% das variáveis do modelo criado manualmente pela empresa Stratio Automotive e com erros de previsão inferiores. A incorporação de fatores meteorológicos nos modelos de previsão influenciou o valor do consumo de combustível, porém as melhorias obtidas nos modelos gerados não foram substanciais.

Palavras-Chave

Algoritmos Evolucionários (AE), Programação Genética (PG), Regressão Simbólica, Previsão, Consumo de Combustível.

Esta página foi propositadamente deixada em branco.

Conteúdo

1	Introdução	1
1.1	Contextualização	2
1.2	Objetivos	3
1.3	Planeamento	3
1.4	Estrutura do documento	4
2	Fundamentos teóricos da Programação Genética	7
2.1	Visão geral da Programação Genética	7
2.2	Representação dos indivíduos	8
2.3	Inicialização da população	9
2.4	Função de fitness	11
2.5	Reprodução	11
2.6	Seleção	11
2.7	Cruzamento	12
2.8	Mutação	13
2.9	Parâmetros de Programação Genética	13
2.10	Limitações	13
2.10.1	<i>Bloat</i>	14
2.10.2	<i>Overfitting</i>	15
3	Trabalhos relacionados	17
3.1	Trabalhos realizados com Programação Genética	17
3.1.1	Programação Genética em farmacocinética preditiva	17
3.1.2	Melhoramento Genético em software	18
3.1.3	Programação Genética baseada em gramática	27
3.1.4	Geração de software com Programação Genética	31
3.2	Trabalhos realizados sobre previsão e otimização do consumo de combustível	33
3.2.1	Otimização do consumo de combustível com Programação Genética	33
3.2.2	Previsão do consumo de combustível com Redes Neurais Artificiais	35
3.2.3	Previsão e otimização do consumo de combustível e de emissão de gases com Redes Neurais Artificiais e Algoritmos Genéticos	36
3.3	Conclusões	38
4	Caracterização dos dados do estudo	41
4.1	Dados do veículo	41
4.2	Dados meteorológicos	45
4.3	Criação dos conjuntos de dados	48
4.4	Seleção de variáveis	49
4.4.1	Análise correlacional	49
4.4.2	Seleção de variáveis pelos técnicos	50
4.5	Arquitetura das experiências	50

5	Geração de modelo baseado no modelo da empresa	53
5.1	Considerações gerais	54
5.2	1ª Fase	55
5.2.1	Ajuste das configurações	55
5.2.2	Redução das variáveis de entrada	57
5.2.3	Teste da capacidade preditiva	60
5.2.4	Conclusões	62
5.3	2ª Fase	63
5.3.1	Redução adicional das variáveis de entrada	63
5.3.2	Eliminação do <i>bloat</i>	66
5.3.3	Conclusões	70
5.4	3ª Fase	71
5.4.1	Validação dos modelos gerados	71
5.4.2	Conclusões	73
5.5	Conclusões gerais	73
6	Geração de modelo baseado em dados reais	75
6.1	Considerações gerais	76
6.2	1ª Fase	77
6.2.1	Geração de modelos com dados reais	77
6.2.2	Eliminação do <i>bloat</i>	79
6.2.3	Conclusões	81
6.3	2ª Fase	81
6.3.1	Validação dos modelos gerados	82
6.3.2	Conclusões	83
6.4	Conclusões gerais	83
7	Incorporação de variáveis meteorológicas	85
7.1	Considerações gerais	86
7.2	1ª Fase A	86
7.2.1	Geração de modelos baseados no modelo fornecido adicionando fatores meteorológicos	86
7.2.2	Conclusões	88
7.3	2ª Fase A	88
7.3.1	Validação dos modelos gerados	88
7.3.2	Conclusões	89
7.4	1ª Fase B	89
7.4.1	Geração de modelos com os dados reais adicionando fatores meteorológicos	89
7.4.2	Eliminação do <i>bloat</i>	91
7.4.3	Conclusões	93
7.5	2ª Fase B	93
7.5.1	Validação dos modelos gerados	94
7.5.2	Conclusões	94
7.6	Conclusões gerais	95
8	Conclusão e trabalho futuro	97
	Referências	98
	Apêndices	103
A	Análise correlacional	105

B	Resultados das experiências realizadas	109
B.1	Resultados das experiências realizadas com o 1º conjunto de dados baseadas no modelo fornecido	110
B.2	Resultados das experiências realizadas com o 2º conjunto de dados baseadas nos dados reais	113
B.3	Resultados das experiências realizadas com o 2º conjunto de dados baseadas no modelo fornecido	115
C	Expressão resultante da experiência PG_1000_NU Grow 1	117
D	Resultados completos da validação dos modelos	119
D.1	Validação dos modelos das experiências da geração de modelo baseado no modelo da empresa	120
D.2	Validação dos modelos das experiências da geração de modelo baseado em dados reais	124
D.3	Validação dos modelos das experiências da geração de modelo baseado no modelo fornecido adicionando fatores meteorológicos	126
D.4	Validação dos modelos das experiências da geração de modelo baseado nos dados reais adicionando fatores meteorológicos	128

Esta página foi propositadamente deixada em branco.

Acrónimos

- %F** Biodisponibilidade Oral Humana. 15
- %PPB** *Plasma Protein Binding*. 15
- ADN** Ácido Desoxirribonucleico. 22, 24
- AE** Algoritmos Evolucionários. 1, 12
- AG** Algoritmos Genéticos. 34, 35
- API** *Application Programming Interface*. 20, 39
- AST** *Abstract Syntax Tree*. 7, 19, 21
- BNF** *Backus Normal Form*. 21, 22
- CAE** Constantes Aleatórias Efémeras. 16, 53, 74, 84
- CE** Computação Evolucionária. 1, 28
- CUDA** *Compute Unified Device Architecture*. 20, 21
- DL50** Dose Letal Mediana. 15
- EG** Evolução Gramatical. 25
- EQM** Erro Quadrático Médio. 10, 16, 52, 74, 84
- GenProg** *Genetic Program Repair*. 19, 20
- GISMOE** *Genetic Improvement of Software for Multiple Objectives Exploration*. 21, 22, 24
- GPU** *Graphics Processing Unit*. 20–22
- IA** Inteligência Artificial. 1, 31
- JM** Janus Manager. 17, 36, 96
- MG** Melhoramento Genético. 16–19, 22, 31
- ML** *Machine Learning*. 15, 58
- MLP** *Multilayer Perceptron*. 34
- P3G** Programação Genética Guiada por Gramática. 27–29
- PG** Programação Genética. x, 1–4, 6, 7, 12, 13, 15–22, 24–31, 33, 34, 36, 37, 39, 47–49, 51–53, 55, 57, 58, 60, 61, 63, 64, 68–74, 76, 81, 84, 86, 88, 89, 93, 95
- RNA** Redes Neurais Artificiais. 33–37

Esta página foi propositadamente deixada em branco.

Lista de Figuras

1.1	Fluxograma da Computação Evolucionária.	1
1.2	Percentagem de erro entre o consumo real de combustível e o consumo anunciado pelos testes laboratoriais, ao longo dos anos (retirado de [1]). . .	3
1.3	Diagrama de Gantt do plano de trabalho estimado para o 2º semestre. . . .	5
1.4	Diagrama de Gantt do plano de trabalho realizado no 2º semestre.	5
2.1	Fluxograma da Programação Genética.	7
2.2	Árvore de sintaxe abstrata da expressão $\max(a+a, a+b*3)$	9
2.3	Criação de uma árvore completa com profundidade máxima 2, utilizando o método de inicialização <i>Full</i> ($t = \text{tempo}$).	9
2.4	Criação de uma árvore de cinco nós com uma profundidade máxima de 2, utilizando o método de inicialização <i>Grow</i> ($t = \text{tempo}$).	10
2.5	Exemplo de um cruzamento (adaptado de [2]).	12
2.6	Exemplo de uma mutação (adaptado de [2]).	13
3.1	Funcionalidade do <i>Janus Manager</i> dividida em processo diurno e noturno (adaptada de [3]).	19
3.2	Exemplo de uma lista de edição e como ela pode evoluir com uma mutação do tipo <i>Grow</i> , <i>Prune</i> ou <i>Single Edit Change</i> (retirado de [3]).	20
3.3	Código dos ficheiros <code>http.c</code> e <code>cgi.c</code> do <i>webservice nullhttpd</i> (retirado de [4]).	22
3.4	Exemplo da contabilização do número de pares em ordem.	27
4.1	Localização das duas <i>Weather Stations</i> de Coimbra.	46
4.2	Pluviosidade regista em Coimbra de 1 a 31 de Janeiro de 2018.	46
4.3	Pluviosidade média regista em Coimbra de 1 a 31 de Janeiro de 2018. . . .	46
4.4	Temperatura máxima, média e mínima regista em Coimbra de 1 a 31 de Janeiro de 2018.	47
4.5	Pressão atmosférica máxima, média e mínima regista em Coimbra de 1 a 31 de Janeiro de 2018.	47
4.6	Velocidade máxima, média e mínima do vento regista em Coimbra de 1 a 31 de Janeiro de 2018.	48
4.7	Direção média do vento regista em Coimbra de 1 a 31 de Janeiro de 2018. .	48
4.8	Representação do intervalo de valores da Correlação de <i>Pearson</i> [5].	49
4.9	Arquitetura 1 das experiências realizadas com os conjuntos de dados do estudo.	51
4.10	Arquitetura 2 das experiências realizadas com os conjuntos de dados do estudo.	51
5.1	Arquitetura da abordagem da geração de um modelo baseado no modelo fornecido.	53

5.2	Fitness mínimo médio dos conjuntos de configurações <i>PG_500</i> e <i>PG_1000</i> , para os três métodos de inicialização da população, durante 500 e 1000 gerações, respetivamente.	56
5.3	Exemplo de uma divisão 3-fold aplicada a um conjunto de dados.	60
5.4	Arquitetura da utilização de uma divisão 3-fold no algoritmo de Programação Genética.	60
5.5	Fitness mínimo médio do conjunto de experiências <i>PG_1000_VF</i> para os três métodos de inicialização da população, durante 1000 gerações.	64
5.6	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_VF_F</i> para os três métodos de inicialização da população, durante 1000 gerações.	65
5.7	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_VF_F_LD17</i> para os três métodos de inicialização da população, durante 1000 gerações.	68
5.8	Fitness mínimo médio e tamanho médio das árvores dos conjuntos de experiências <i>PG_1000_VF_F_LD17</i> , <i>PG_1000_VF_F_LD30</i> e <i>PG_1000_VF_F_LD35</i> para o método de inicialização da população <i>Grow</i> , durante 1000 gerações.	68
5.9	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_VF_F_LDTA</i> para os três métodos de inicialização da população, durante 1000 gerações.	69
6.1	Arquitetura da abordagem da geração de um modelo baseado em dados reais.	75
6.2	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_DR</i> para os três métodos de inicialização da população, durante 1000 gerações.	78
6.3	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_DR_LD17</i> para os três métodos de inicialização da população, durante 1000 gerações.	80
6.4	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_DR_LDTA</i> para os três métodos de inicialização da população, durante 1000 gerações.	80
7.1	Arquitetura da abordagem da incorporação de variáveis meteorológicas.	85
7.2	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_VF_F_CM</i> para os três métodos de inicialização da população, durante 1000 gerações.	87
7.3	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_DR_CM</i> para os três métodos de inicialização da população, durante 1000 gerações.	90
7.4	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_DR_CM_LD17</i> para os três métodos de inicialização da população, durante 1000 gerações.	92
7.5	Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências <i>PG_1000_DR_CM_LDTA</i> para os três métodos de inicialização da população, durante 1000 gerações.	92

Lista de Tabelas

2.1	Exemplos de possíveis primitivas que podem constituir um conjunto de símbolos terminais em Programação Genética.	8
2.2	Exemplos de possíveis primitivas que podem constituir um conjunto de funções em Programação Genética.	8
2.3	Exemplos de possíveis valores da função de fitness de 6 indivíduos.	12
3.1	Valores dos parâmetros de configuração de Programação Genética quando aplicada ao P3G.	30
5.1	Configurações do algoritmo de Programação Genética transversais a todos os conjuntos de experiências apresentados no capítulo 5.	55
5.2	Resultados dos conjuntos de experiências <i>PG_500</i> e <i>PG_1000</i> realizados durante a etapa de ajuste de configurações.	57
5.3	Resultados dos conjuntos de experiências <i>PG_1000_NU</i> , <i>PG_1000_CORR</i> e <i>PG_1000_NU_CORR</i> realizados durante a etapa de redução das variáveis de entrada.	59
5.4	Resultados do conjunto de experiências <i>PG_1000_E2</i> realizado durante a etapa de teste da capacidade preditiva.	61
5.5	Resultados dos conjuntos de experiências <i>PG_1000_VF</i> e <i>PG_1000_VF_F</i> realizados durante a etapa de redução adicional das variáveis de entrada.	64
5.6	Resultados dos conjuntos de experiências <i>PG_1000_VF_F_LD17</i> , <i>PG_1000_VF_F_LD30</i> e <i>PG_1000_VF_F_LD35</i> realizados durante a etapa de eliminação do <i>bloat</i>	67
5.7	Resultados do conjunto de experiências <i>PG_1000_VF_F_LDTA</i> realizado durante a etapa de eliminação do <i>bloat</i>	69
5.8	Resultados da validação dos modelos gerados pelos conjuntos de experiências: <i>PG_1000_VF_F</i> , <i>PG_1000_VF_F_LD17</i> , <i>PG_1000_VF_F_LD30</i> , <i>PG_1000_VF_F_LD35</i> e <i>PG_1000_VF_F_LDTA</i>	72
6.1	Configurações do algoritmo de Programação Genética transversais a todos os conjuntos de experiências apresentados no capítulo 6.	76
6.2	Resultados do conjunto de experiências <i>PG_1000_DR</i> realizado durante a etapa da geração de modelos com dados reais.	77
6.3	Resultados dos conjuntos de experiências <i>PG_1000_DR_LD17</i> e <i>PG_1000_DR_LDTA</i> realizado durante a etapa da eliminação do <i>bloat</i>	79
6.4	Resultados da validação dos modelos gerados pelos conjuntos de experiências: <i>PG_1000_DR</i> , <i>PG_1000_DR_LD17</i> e <i>PG_1000_DR_LDTA</i>	82
7.1	Resultados do conjunto de experiências <i>PG_1000_VF_F_CM</i> realizado durante a etapa da geração de modelos baseados no modelo fornecido, incorporando fatores meteorológicos.	87

7.2	Resultados da validação dos modelos gerados pelo conjunto de experiências <i>PG_1000_VF_F_CM</i>	88
7.3	Resultados do conjunto de experiências <i>PG_1000_DR_CM</i> realizado durante a etapa da geração de modelos com dados reais, incorporando fatores meteorológicos.	89
7.4	Resultados dos conjuntos de experiências <i>PG_1000_DR_CM_LD17</i> e <i>PG_1000_DR_CM_LDTA</i> realizado durante a etapa da eliminação do <i>bloat</i>	91
7.5	Resultados da validação dos modelos gerados pelos conjuntos de experiências: <i>PG_1000_DR_CM</i> , <i>PG_1000_DR_CM_LD17</i> e <i>PG_1000_DR_CM_LDTA</i>	94
A.1	Tabela dos resultados completos da análise correlacional	107
B.1	Resultados das experiências realizadas com o 1ºconjunto de dados baseadas no modelo fornecido.	113
B.2	Resultados das experiências realizadas com o 2ºconjunto de dados baseadas nos dados reais.	115
B.3	Resultados das experiências realizadas com o 2ºconjunto de dados baseadas no modelo fornecido.	116
D.1	Erro absoluto entre os modelos gerados na 2ªfase capítulo 5 e o consumo real de combustível.	124
D.2	Erro absoluto entre os modelos gerados no capítulo 6 e o consumo real de combustível.	126
D.3	Erro absoluto entre os modelos gerados na 1ªfase A capítulo 7 e o consumo real de combustível.	128
D.4	Erro absoluto entre os modelos gerados na 1ªfase B capítulo 7 e o consumo real de combustível.	130

Capítulo 1

Introdução

Os Algoritmos Evolucionários (AE), um subconjunto da Computação Evolucionária (CE), são inspirados na teoria da evolução de Charles Darwin e nos seus mecanismos fundamentais [6, 7, 8]. Segundo Darwin todos os seres vivos são diferentes e por isso uns são mais aptos para determinados ambientes, tendo uma maior probabilidade de sobreviverem e de gerarem descendentes que herdarão as suas características [9]. Estes algoritmos, como mostra a Figura 1.1, encontram-se divididos em quatro técnicas: Estratégias de Evolução (EE) [10], Programação Evolucionária (PE) [11], Algoritmos Genéticos (AG) [12] e Programação Genética (PG) [2].



Figura 1.1: Fluxograma da Computação Evolucionária.

A técnica da Programação Genética (PG) foi popularizada por John Koza em 1992 com o objetivo de solucionar uma ampla gama de problemas. Nesta técnica, a procura de uma solução para um dado problema é feita gerando, de forma automática, programas de computador (e.g. expressões matemáticas, expressões lógicas, código) sem exigir que o utilizador conheça ou especifique, antecipadamente, a forma ou a estrutura da solução [2]. Para tal, é necessário uma declaração de alto nível do objetivo e poder computacional suficiente.

Ao longo dos anos, esta técnica tem-se demonstrado uma área muito promissora de pesquisa em Inteligência Artificial (IA) dada a sua robustez, tendo sido amplamente utilizada na descoberta de relações funcionais entre variáveis pertencentes a um conjunto de dados

(Regressão Simbólica) [13, 14, 15, 16], no agrupamento de dados em categorias (Classificação) [17] e, mais recentemente, no processo de desenvolvimento de software, nomeadamente na geração automática de código [18, 19, 20], melhoramento genético [21, 22] e correção automática de falhas [3, 4, 23].

Esta dissertação está focada na determinação de um modelo capaz de prever o número de litros consumidos pelos veículos pesados durante o seu funcionamento e estudar a influência que os fatores meteorológicos têm na previsão desse mesmo valor. Para tal, será utilizada uma Regressão Simbólica com Programação Genética (PG).

1.1 Contextualização

No mundo real, muitos dos problemas com que nos deparamos podem ser representados como problemas de otimização, onde um conjunto de parâmetros tem de ser ajustado para atingir uma solução ou um conjunto de soluções. Contudo, a maioria desses problemas, entre eles a previsão do consumo de combustível, são realmente complexos o que faz com que, em geral, não possam ser resolvidos de forma eficiente com técnicas determinísticas de otimização [24]. Dentro deste contexto, a empresa Stratio Automotive¹ demorou vários meses a gerar manualmente um modelo, seguindo as leis da Física, capaz de prever o consumo de combustível de veículos pesados de um modelo e marca específicos, baseado em vários parâmetros associados ao motor. No entanto, esse modelo depende de um grande número de variáveis e foi gerado em condições ideais (assumindo um veículo composto apenas por peças de origem e em correto funcionamento), apresentando elevados valores de erro quando testado em contexto real impossibilitando assim, a sua utilização.

No contexto de uma empresa de transportes, a gestão de combustível na sua frota apresenta-se como uma questão vital na sua faturação muito influenciada por fatores como a diminuição dos recursos fósseis e consecutiva escalada dos preços dos combustíveis, condução efetuada pelos seus motoristas, manutenção proativa, entre outros. Ainda que a indústria de fabrico automóvel forneça modelos de consumo específicos para cada veículo, esses valores têm, ao longo do tempo, apresentando uma discrepância cada vez maior face aos valores reais. A existência de uma pequena discrepância acontece sempre, porque a determinação do modelo é feita em condições ideais, não tendo em consideração a maioria dos fatores que têm uma influência direta no consumo real de combustível. No entanto, estudos realizados pela Federação Europeia dos Transportes e Ambiente detetaram que essa discrepância atingiu os 42% em 2015, Figura 1.2, e afirmam que existe uma tendência para aumentar até 2020 [25].

Dadas as incongruências identificadas nos modelos dos fabricantes, as empresas de transportes recorrem aos valores de consumo apresentados nos veículos ou a sistemas de gestão de frotas para calcular o consumo médio de cada viatura e os gastos associados. Ainda que esses valores forneçam as condições necessárias para o cálculo de uma média de consumo, não permitem efetuar previsões com base em fatores internos e externos ao veículo.

¹Para mais informações consultar: www.stratioautomotive.com

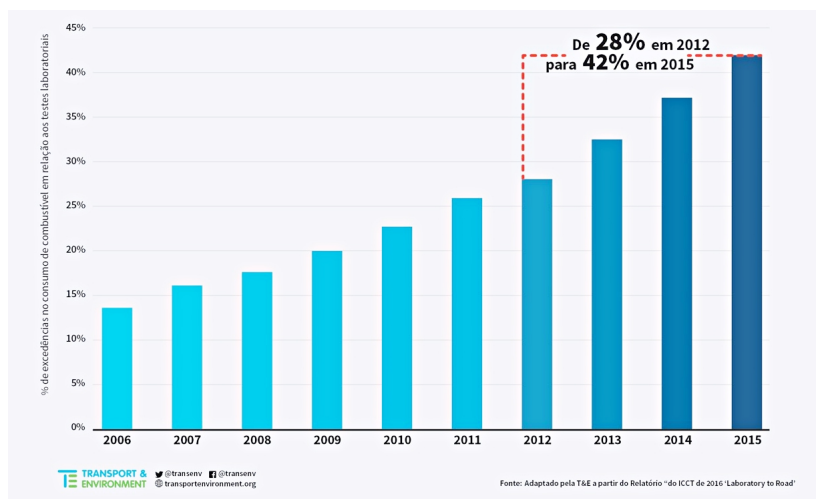


Figura 1.2: Percentagem de erro entre o consumo real de combustível e o consumo anunciado pelos testes laboratoriais, ao longo dos anos (retirado de [1]).

1.2 Objetivos

O objetivo principal desta dissertação é utilizar PG para gerar um modelo capaz de prever o consumo de combustível de veículos pesados de um modelo e marca específicos, utilizando dados das características dos veículos e dados reais recolhidos pelos sensores durante o seu funcionamento. Este objetivo encontra-se dividido em três partes:

- Em primeiro lugar, pretende-se gerar um modelo baseado no modelo já existente, criado manualmente pela empresa Stratio Automotive, reduzindo o número de variáveis de entrada;
- Em segundo lugar, o objetivo é gerar um novo modelo baseado nos dados reais apenas com as variáveis que no ponto anterior se demonstrarem essenciais;
- Por fim, pretende-se adicionar fatores externos tais como a pressão atmosférica e a temperatura do ar e estudar a sua influência na previsão do consumo de combustível.

1.3 Planeamento

Durante o segundo semestre, o foco da dissertação esteve utilização da PG para eliminar algumas das variáveis de entrada do modelo matemático fornecido pela empresa e pela criação automática de um modelo que incluía os fatores internos e externos ao veículo, permitindo uma boa previsão do consumo de combustível, utilizando os conhecimentos adquiridos no primeiro semestre. Para tal, tinha sido definido o plano de trabalho que se encontra na Figura 1.3.

Do plano inicialmente proposto, todas as tarefas definidas foram cumpridas, ainda que tenham ocorrido algumas discrepâncias. Essas discrepâncias ocorreram em grande parte

devido ao tempo necessário para a execução das experiências com o algoritmo de PG, isto porque foram surgindo algumas limitações para as quais foi preciso implementar alterações ao algoritmo. Essas alterações refletem-se em novas tarefas, algumas delas que demoraram mais de um mês na implementação e execução como, por exemplo, a eliminação do *bloat*, tal como mostra o diagrama de Gantt da Figura 1.4. Além disso, foi necessário aguardar pela aprovação dos clientes para o fornecimento dos dados e pela sua recolha por parte da empresa.

Até à data da entrega deste relatório o artigo científico encontra-se em fase de submissão.

1.4 Estrutura do documento

O restante documento encontra-se dividido em 7 capítulos.

Nos capítulos 2 e 3 são descritos detalhadamente os fundamentos teóricos da PG e alguns dos trabalhos recentemente publicados sobre a sua aplicabilidade (secção 3.1) e sobre a previsão e otimização do consumo de combustível (secção 3.2).

No capítulo 4 é feita uma caracterização dos dados e das arquiteturas das experiências realizadas e nos capítulos 5, 6 e 7 são descritos os resultados obtidos e as conclusões retiradas em cada uma das etapas/fases.

Por fim, no capítulo 8 é feito um apanhado das conclusões chave de toda a dissertação.

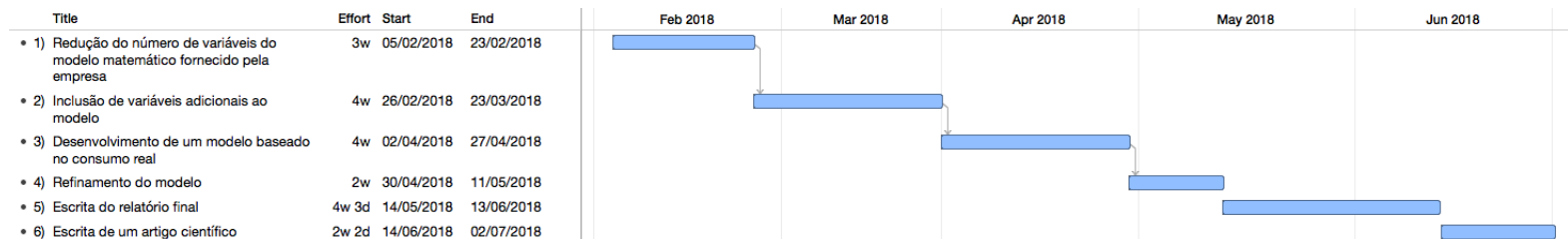


Figura 1.3: Diagrama de Gantt do plano de trabalho estimado para o 2º semestre.

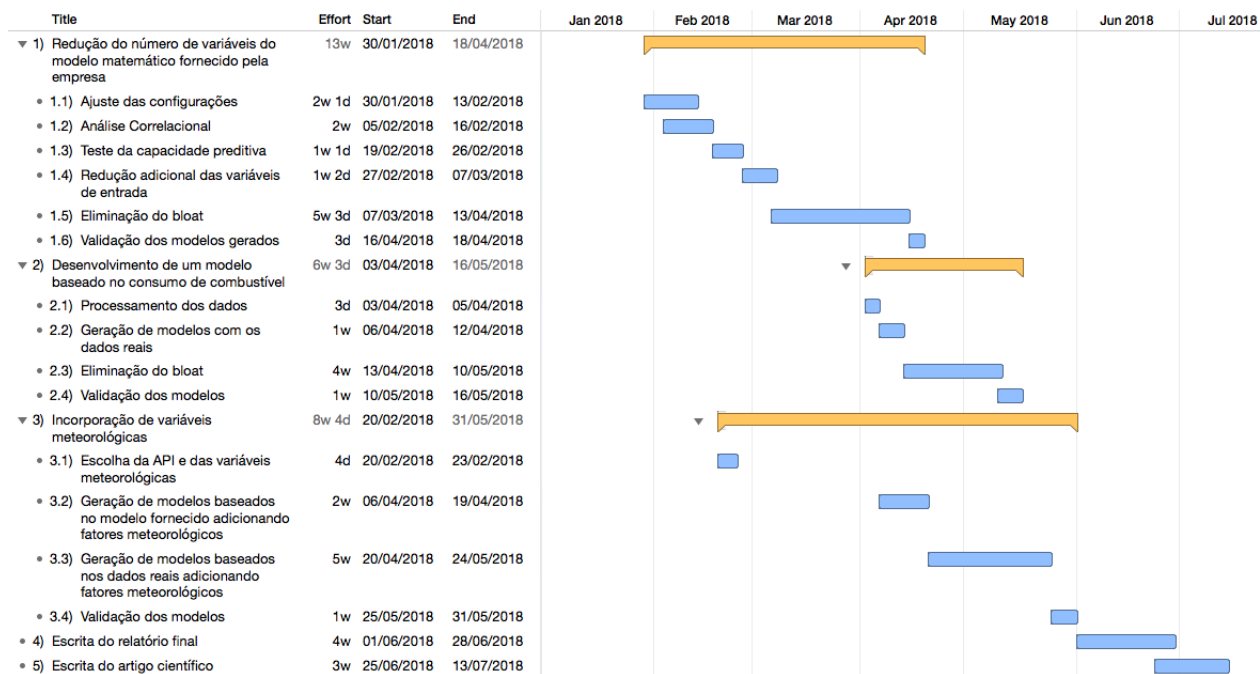


Figura 1.4: Diagrama de Gantt do plano de trabalho realizado no 2º semestre.

Esta página foi propositadamente deixada em branco.

Capítulo 2

Fundamentos teóricos da Programação Genética

Nas secções seguintes, são descritos detalhadamente os conceitos principais e as terminologias da Programação Genética (PG). Essa descrição é feita com base na PG *standard*, definida por Koza [2].

2.1 Visão geral da Programação Genética

A PG é utilizada para evoluir um conjunto de programas de computador, com vista a encontrar aquele que consiga resolver automaticamente um dado problema.

A Figura 2.1 mostra o funcionamento geral da PG.

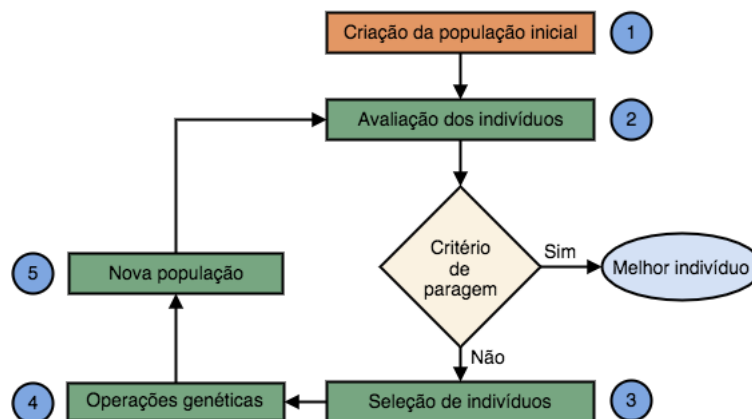


Figura 2.1: Fluxograma da Programação Genética.

Inicialmente, é criada uma população - geração 0 - designada população inicial (Passo 1), composta por vários indivíduos (possíveis soluções do problema) normalmente gerados de forma aleatória. A partir daí, a todos os indivíduos é atribuído um valor, designado valor de fitness, que representa a sua capacidade para resolver o problema em análise (Passo 2). Caso o critério de paragem seja satisfeito, o melhor indivíduo da população é devolvido

como solução ao problema. Caso contrário, os melhores indivíduos são selecionados (Passo 3) e são-lhes aplicados os operadores genéticos (Passo 4), criando uma nova população (Passo 5). Cada execução corresponde a uma nova geração. Os operadores genéticos são aplicados probabilisticamente e são os seguintes: **Reprodução**, **Seleção**, **Cruzamento** e **Mutação** descritos nas secções 2.5-2.8 respetivamente.

2.2 Representação dos indivíduos

Numa abordagem clássica de PG, a representação típica de cada indivíduo é uma *Abstract Syntax Tree* (AST). Cada árvore é constituída por um conjunto de funções, $F = \{f_1, f_2, \dots, f_n\}$, que representam os nós, e por um conjunto de símbolos terminais, $T = \{t_1, t_2, \dots, t_n\}$, que representam as folhas. Desta forma, o espaço de procura é constituído por todas as expressões válidas que podem ser construídas recursivamente com as funções do conjunto F e os símbolos terminais do conjunto T :

- **Conjunto de símbolos terminais (T):** Este conjunto é composto por variáveis, constantes e funções com aridade zero (sem argumentos) definidas no domínio do problema. Alguns exemplos encontram-se na Tabela 2.1.

Símbolos terminais	
Tipo de primitiva	Exemplo(s)
Variáveis	x, y, a, b
Valores constantes	3, 0.45
Funções com aridade zero	rand()

Tabela 2.1: Exemplos de possíveis primitivas que podem constituir um conjunto de símbolos terminais em Programação Genética.

- **Conjunto de funções (F):** Este conjunto é composto por todas as funções de qualquer tipo que aceitem um número variado de argumentos. Uma operação introduzida por Koza foi a divisão protegida, que retorna 1 quando o denominador da divisão é igual a zero, evitando falhas de execução [2]. Alguns exemplos encontram-se na Tabela 2.2.

Funções	
Tipo de primitiva	Exemplo(s)
Aritmética	+, *, /
Matemática	sin, cos, exp
Booleana	AND, OR, NOT
Condicional	IF-THEN-ELSE

Tabela 2.2: Exemplos de possíveis primitivas que podem constituir um conjunto de funções em Programação Genética.

As funções do conjunto F devem obedecer a propriedades específicas, garantindo a consistência do dados e a segurança na avaliação das expressões criadas. Desta forma, é importante, por exemplo, na árvore da Figura 2.2 garantir que os valores das variáveis, a e b , estejam definidos para as operações em que estão envolvidos, $+$ e $*$.

Mesmo que a estrutura da solução não seja conhecida, é importante que os conjuntos F e T sejam representativos, facilitando a procura de uma solução para o problema.

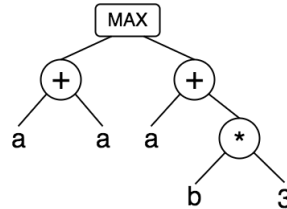


Figura 2.2: Árvore de sintaxe abstrata da expressão $\max(a+a, a+b*3)$.

A Figura 2.2 mostra a representação do programa $\max(a+a, a+b*3)$, onde as variáveis e as constantes (a, b e 3) do programa constituem o conjunto T e as operações aritméticas ($+, *$ e \max) constituem o conjunto F .

2.3 Inicialização da população

A geração da população inicial é normalmente feita de forma aleatória. No entanto, existem vários métodos utilizados para a sua geração, entre eles estão três mais comuns: *Full*, *Grow* e *Ramped half-and-half*.

No método *Full*, as árvores são geradas de forma a que não excedam uma profundidade máxima definida pelo utilizador. A profundidade de um nó corresponde ao número de arestas a serem percorridas para alcançar esse nó a partir do nó raiz (profundidade 0). A raiz da árvore é constituída por uma função do conjunto F escolhida aleatoriamente. Depois disso, outras funções são também escolhidas de forma aleatória para formar os nós da árvore. Os que se encontram na profundidade máxima, designados folhas, são preenchidos apenas com símbolos terminais do conjunto T . Desta forma, é criada uma árvore completa em que todas as folhas se encontram à mesma profundidade. A Figura 2.3 é um exemplo da criação de uma árvore completa com profundidade 2.

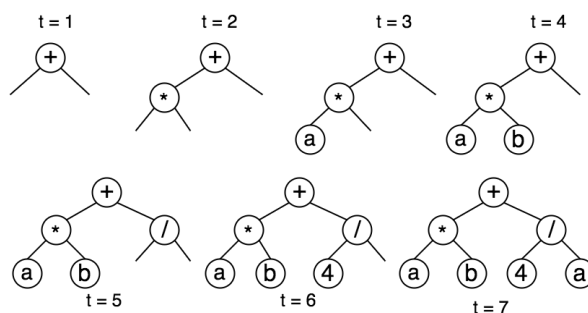


Figura 2.3: Criação de uma árvore completa com profundidade máxima 2, utilizando o método de inicialização *Full* ($t =$ tempo).

No algoritmo 1 está ilustrado o pseudocódigo correspondente a este método quando o parâmetro `metodo` é igual a `full`.

O método *Grow* funciona de forma semelhante ao anterior, mas permite criar árvores que não são completas. Isto é, os nós são criados com elementos escolhidos aleatoriamente dos conjuntos `T` e `F`. Sempre que um símbolo terminal é escolhido, o crescimento da árvore nesse ramo pára mesmo que não se encontre na profundidade máxima. A Figura 2.4 é um exemplo da criação de uma árvore de profundidade 2 com o método *Grow*.

No algoritmo 1 está o pseudocódigo correspondente a este método quando o parâmetro `metodo` é igual a `grow`.

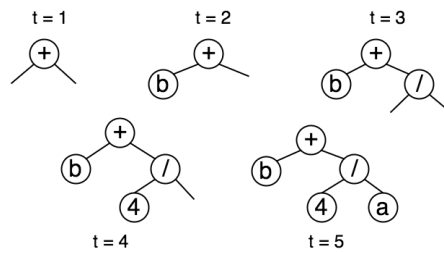


Figura 2.4: Criação de uma árvore de cinco nós com uma profundidade máxima de 2, utilizando o método de inicialização *Grow* ($t = \text{tempo}$).

Algoritmo 1 Pseudocódigo dos métodos *Full* e *Grow*

```

1: procedure GEN_EXP(conj_func, conj_term, max_profund, metodo)
2:   if max_profund = 0 or ( metodo = grow and rand(0,1) =  $\frac{|conj\_term|}{|conj\_term|+|conj\_func|}$  )
3:     then
4:       expr = escolha_aleatoria_elemento(conj_term)
5:     else
6:       expr = escolha_aleatoria_elemento(conj_func)
7:       for i = 1 to arity(func) do
8:         arg_i = gen_rnd_expr(conj_func, conj_term, max_profund - 1, metodo)
9:       end for
10:      expr = (func, arg_1, arg_2, ...)
11:   end if
12:   return expr
13: end procedure

```

Nota: `conj_func` é o conjunto de funções; `conj_term` é o conjunto de símbolos terminais; `max_profund` é a profundidade máxima permitida para a árvore gerada; `metodo` pode ser igual a `grow` ou a `full`; `arity(func)` é o número de parâmetros da função `func`; `expr` é a expressão correspondente à árvore gerada e `rand(0,1)` é uma função que retorna um número uniformemente distribuído entre 0 e 1.

No método *Ramped half-and-half*, uma parte da população é criada usando o método *Full* e a outra parte é criada usando o método *Grow*. Assim sendo, a população será constituída por árvores de diferentes tamanhos (o tamanho de cada árvore é escolhido aleatoriamente entre 1 e a profundidade máxima), garantido uma maior diversidade. No algoritmo 2 está o pseudocódigo correspondente a este método.

Algoritmo 2 Pseudocódigo do método *Ramped half-and-half*

```

1: procedure GEN_EXP_RHH(conj_func, conj_term, max_profund, prob_grow)
2:   profund = rand(1, max_profund)
3:   if rand(0,1) < prob_grow then
4:     return gen_exp(conj_func, conj_term, profund, grow)
5:   else
6:     return gen_exp(conj_func, conj_term, profund, full)
7:   end if
8: end procedure

```

Nota: `conj_func` é o conjunto de funções; `conj_term` é o conjunto de símbolos terminais; `max_profund` é a profundidade máxima permitida para a árvore gerada; `prob_grow` é a probabilidade de criar a árvore com o método `grow` e `rand(1,max_profund)` é uma função que retorna um número uniformemente distribuído entre 1 e `max_profund`.

2.4 Função de fitness

Cada indivíduo da população é criado com base nos conjuntos de símbolos terminais e de funções. No entanto, a sua criação não garante que ele resolva ou parcialmente resolva o problema. Desta forma, é necessário quantificar num valor, por recurso à função de fitness, a capacidade que um dado indivíduo tem para resolver o problema.

A função de fitness permite avaliar um indivíduo para um dado conjunto de casos de aptidão conhecidos.

Usando um problema de regressão simbólica como exemplo, a função de fitness normalmente utilizada é o Erro Quadrático Médio (EQM).

2.5 Reprodução

A reprodução está geralmente associada ao parâmetro elitismo. Esse parâmetro indica o número de indivíduos que são copiados de uma geração para a seguinte sem que sofram qualquer alteração genética.

2.6 Seleção

Dada uma população, em que foi atribuído a cada indivíduo um valor de fitness, é necessário selecionar aqueles que vão sofrer operações genéticas de cruzamento e mutação. Isto leva a que um indivíduo que apresente um melhor valor de fitness tenha uma maior probabilidade de ter mais filhos do que outro que apresente um valor pior.

Existem diversas formas de seleção, mas a mais conhecida é a seleção por torneio [2]. Nesta, não é feita uma comparação entre todos os elementos da população, apenas n indivíduos são selecionados de forma aleatória usando uma distribuição uniforme, sem qualquer ligação com o valor do fitness. Esses n indivíduos selecionados são comparados e aquele que apresentar melhor valor de fitness é escolhido. Quando é feita uma operação de

cruzamento é necessário selecionar dois indivíduos e por isso são realizados dois torneios. Este método de seleção impede que, caso exista um indivíduo com um bom valor de fitness, a geração seguinte seja constituída apenas por filhos desse indivíduo, o que levaria a uma rápida perda de diversidade.

Supondo que uma população é composta por seis indivíduos, com os valores de fitness presentes na Tabela 2.3. Aplicando uma seleção por torneio de tamanho $n = 3$, poderiam ser escolhidos os indivíduos 1, 4 e 5.

Entre os indivíduos escolhidos pelo torneio, aquele que apresenta melhor valor de fitness é o 4 e, conseqüentemente, é o selecionado.

Indivíduo	Fitness
1	0.2
2	0.1
3	0.4
4	0.5
5	0.3
6	0.8

Tabela 2.3: Exemplos de possíveis valores da função de fitness de 6 indivíduos.

2.7 Cruzamento

O cruzamento permite introduzir diversidade na população ao longo das várias gerações. Este operador gera novos indivíduos (filhos) com base noutros indivíduos (pais) que foram selecionados (descrito na secção 2.6).

Este operador começa por selecionar o nó de cruzamento em cada cópia das árvores pais (representado a cinzento na Figura 2.5). Em seguida é criado um filho, substituindo a subárvore do ponto de cruzamento do pai 1 pela subárvore do pai 2, como mostra a Figura 2.5.

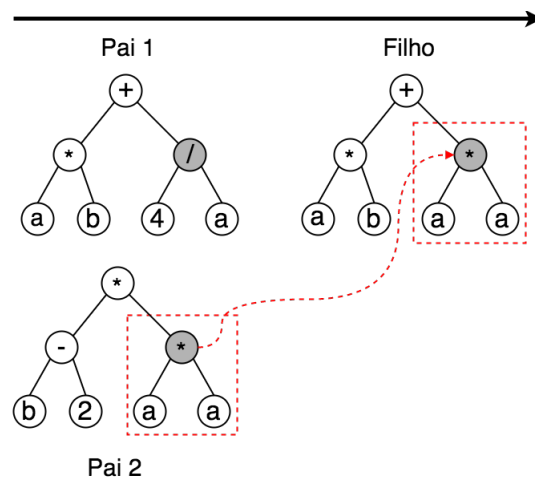


Figura 2.5: Exemplo de um cruzamento (adaptado de [2]).

Os pontos de cruzamento não são escolhidos uniformemente de entre os nós existentes, uma vez que isso levaria a cruzamentos de pequenas subárvores como, por exemplo, a troca de folhas. Desta forma, Koza [2] sugere que 90% das vezes devem ser escolhidos nós compostos por funções e nos restantes 10% sejam escolhidas folhas.

2.8 Mutação

A mutação é um operador genético que deve geralmente ocorrer depois do cruzamento. Este operador é aplicado com uma probabilidade baixa e atua apenas sobre um indivíduo, introduzindo diversidade à população.

Nesta operação genética é escolhido um ponto de mutação na cópia da árvore pai e essa subárvore é substituída por outra gerada aleatoriamente, como mostra a Figura 2.6.

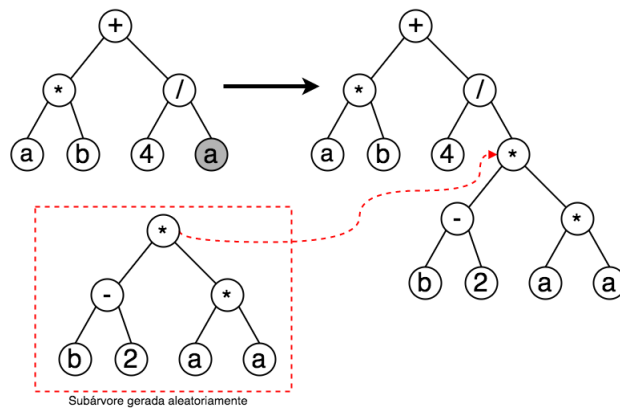


Figura 2.6: Exemplo de uma mutação (adaptado de [2]).

2.9 Parâmetros de Programação Genética

Antes de executar um algoritmo de PG é necessário definir um conjunto de parâmetros importantes para o desempenho dos Algoritmos Evolucionários (AE). Alguns desses parâmetros são: o tamanho da população, o critério de paragem, a probabilidade de cruzamento e de mutação, a profundidade máxima das árvores, os métodos de seleção e os conjuntos de funções e símbolos terminais.

2.10 Limitações

Para que o algoritmo de PG consiga obter bons resultados é importante uma correta definição dos parâmetros enunciados anteriormente. Contudo, algumas das vezes, mesmo quando é feita uma correta definição desses parâmetros existem limitações que permanecem e que têm sido alvo de vários estudos. Essas limitações são *bloat* e *overfitting*.

2.10.1 *Bloat*

O *bloat* é caracterizado por um crescimento excessivo do tamanho dos indivíduos durante a fase de procura, sem que se verifiquem melhorias proporcionais nos valores de fitness. Ainda que esse crescimento seja normal e resultante da aplicação dos operadores genéticos, muitas das vezes corresponde ao aparecimento de redundância em segmentos do mesmo indivíduo.

Para combater o *bloat* foram propostas técnicas que permitem controlar o crescimento descontrolado do tamanho das árvores, resultantes da aplicação dos operadores genéticos. Por exemplo, a abordagem mais simples e tradicional é a utilização de um limite fixo [2]. No entanto, estudos mais recentes [26] mostraram que é possível tornar esse limite dinâmico em que o seu valor se ajusta ao longo das gerações.

Limite estático

A PG *standard* utiliza limites estáticos para o tamanho e/ou profundidade das árvores geradas pelos operadores genéticos.

Sempre que é aplicado um operador genético é verificado se a sua descendência excede o limite definido. Se não exceder, o indivíduo é incluído na população, caso contrário, é eliminado e um dos seus progenitores é selecionado para pertencer à geração seguinte. Esta implementação impede o crescimento excessivo dos indivíduos. No entanto, o controlo de *bloat* só começa a ocorrer quando o limite definido é atingido e, em problemas de elevada complexidade, a sua natureza estática pode impedir que a solução ótima seja encontrada [26].

Limite dinâmico

O limite dinâmico é uma técnica de controlo de *bloat* inspirada no tradicional limite estático que, tal como o nome indica, é dinâmico, alterando-se ao longo das gerações. Mais uma vez, pode ser aplicado ao tamanho ou à profundidade das árvores.

Inicialmente é definido com um valor baixo, pelo menos tão baixo quanto a profundidade máxima das árvores da população inicial. Ao longo da execução do algoritmo podem ocorrer quatro situações:

- O novo indivíduo não é o melhor indivíduo encontrado até ao momento e não excede o limite dinâmico: O novo indivíduo é aceite na população;
- O novo indivíduo não é o melhor indivíduo encontrado até ao momento e excede o limite dinâmico: O novo indivíduo não é aceite na população, sendo substituído por um dos seus progenitores;
- O novo indivíduo é o melhor indivíduo encontrado até ao momento e não excede o limite dinâmico: O novo indivíduo é aceite na população e o limite dinâmico baixa de forma a ser igual à sua profundidade;

- O novo indivíduo é o melhor indivíduo encontrado até ao momento e excede o limite dinâmico: O novo indivíduo é aceite na população e o limite dinâmico aumenta de forma a ser igual à sua profundidade.

A utilização de um limite dinâmico não invalida a utilização de um limite estático.

2.10.2 *Overfitting*

O *overfitting* ocorre quando um modelo aprende os detalhes e o ruído do conjunto de dados de treino. Isto significa que o ruído e as flutuações aleatórias nos dados são aprendidos como conceitos pelo modelo. O problema é que esses conceitos não se aplicam a novos dados, afetando negativamente a capacidade de generalização dos modelos [27].

Para estimar a precisão de um modelo são utilizadas técnicas de reamostragem e/ou é retido um conjunto de dados de validação. A técnica de reamostragem mais popular é a validação cruzada k-fold. Por sua vez, um conjunto de dados de validação é simplesmente um subconjunto dos dados de entrada que não é considerado na geração do modelo, sendo posteriormente utilizado para avaliar o seu comportamento perante dados não vistos.

Esta página foi propositadamente deixada em branco.

Capítulo 3

Trabalhos relacionados

Neste capítulo, em primeiro lugar é feito um resumo de vários estudos recentes englobando diferentes áreas, onde a PG com algumas variantes tem-se demonstrado uma técnica promissora. Em seguida, são apresentados trabalhos focados na previsão e otimização do consumo de combustível e por fim, é feito um apanhado dos pontos chave dos trabalhos apresentados.

3.1 Trabalhos realizados com Programação Genética

Os trabalhos seguintes permitem comprovar o interesse dos investigadores na PG, com a qual é possível criar modelos que representem de forma simplificada uma realidade ou parte dela. Esses modelos permitem analisar fenómenos de sistemas complexos que apenas pela observação da realidade seriam de difícil interpretação.

3.1.1 Programação Genética em farmacocinética preditiva

A eficácia de um medicamento está não só associada às moléculas dos seus compostos, mas também ao caminho que cada uma delas realizada dentro do corpo de um paciente sem que este manifeste reações adversas. Para tal, é necessário conseguir prever o comportamento das moléculas, o que é possível com base em três parâmetros - Biodisponibilidade Oral Humana (%F), Dose Letal Mediana (DL50) e *Plasma Protein Binding* (%PPB) - que são possíveis de prever utilizando técnicas de *Machine Learning* (ML).

Vários estudos realizados [13, 14, 15] têm utilizado variantes de PG para prever esses parâmetros, comparando-as a alguns métodos de regressão mais simples. Para os estudos apresentados, todas as técnicas foram testadas com e sem uma fase de pré-processamento de seleção de atributos.

As variantes de PG testadas foram:

- PG *standard*: tal como é descrita no capítulo 2;

- *DF-GP*: semelhante à versão *standard*, mas com uma função de fitness dinâmica que muda durante a evolução da população;
- *LS2-C-GP*: semelhante à versão *standard*, mas com uma função de fitness que otimiza simultaneamente o EQM e o coeficiente de correlação. Além disso, o conjunto de símbolos terminais é composto por variáveis de ponto flutuante, que representam os 241 descritores moleculares presentes na base de dados, e por Constantes Aleatórias Efêmeras (CAE).

As CAE são um conjunto de constantes aleatórias fixas, geradas como parte do processo de inicialização da população. Sempre que uma é escolhida como símbolo terminal, é gerado um valor aleatório diferente que é usado para essa folha e que permanece fixo para a restante execução.

Os resultados obtidos mostram que as variantes de PG, em especial a *LS2-C-GP*, apresentam melhores resultados comparativamente às restantes técnicas de regressão, tanto do ponto de vista da qualidade da solução como da correlação entre os dados previstos e os dados corretos, demonstrando ser uma das técnicas mais promissoras para a previsão de parâmetros farmacocinéticos. Além disso, é possível afirmar que otimizar mais do que um critério em simultâneo pode ser benéfico.

A aplicação de uma fase de pré-processamento de seleção de atributos é desnecessária em técnicas de PG para que estas apresentem melhores resultados, permitindo afirmar que esta técnica faz essa seleção, poupando tempo e recursos, o que é uma vantagem comparativamente às restantes.

3.1.2 Melhoramento Genético em software

O Melhoramento Genético (MG) é um processo que aperfeiçoa automaticamente o comportamento de um sistema que usa a PG. A sua utilização na manutenção de software tem crescido, com vista a otimizar o código escrito manualmente numa variedade de propriedades funcionais e não funcionais [28].

Num terço das vezes, os critérios de melhoria são propriedades não funcionais, como é o caso do tempo de execução, uso de memória e consumo de energia [29].

Os principais objetivos da aplicação de MG a software já existente são a automatização do processo de correção de falhas [3, 4, 23], tornando-o menos demorado e dispendioso [30], o melhoramento das propriedades não funcionais e a geração automática de código.

3.1.2.1 Correção automática de falhas

A PG tem obtido excelentes resultados quando aplicada a programas já existentes, em especial na correção de falhas, permitindo corrigir automaticamente a maioria delas [31]. Além de uma grande parte dos estudos sobre MG terem sido realizados em laboratório em copias do programa original, o objetivo final é a sua utilização para criar softwares que se auto aperfeiçoam.

As pesquisas atuais sobre este tema ainda apresentam conceitos iniciais. Assim sendo, primeiro é necessário oferecer aos programadores ferramentas que permitam melhorar o software existente e só depois software que se auto ajuste.

Janus Manager

Janus Manager (JM) é um programa desenvolvido para um centro de reabilitação vocacional, permitindo que os especialistas consigam ter uma visão geral do processo de reabilitação dos vários clientes.

Haraldsson *et al.*, num artigo publicado recentemente [3], mostraram que este software foi um dos primeiros passos na correção automática de falhas com PG.

Este software apresenta dois comportamentos distintos consoante a fase do dia em que se encontra. Durante o dia, lado esquerdo da Figura 3.1, sempre que uma solicitação e/ou fornecimento de dados por parte de um utilizador desencadear uma exceção, a informação referente a esse acontecimento é registada no ficheiro de *logs*, representado ao centro na Figura 3.1. Nesse ficheiro ficam registados os dados de entrada da função onde a exceção ocorreu e a respetiva exceção. No final do dia, todas as informações referentes aos acontecimentos inesperados no software encontram-se registadas no ficheiro e são posteriormente utilizadas na correção de falhas.

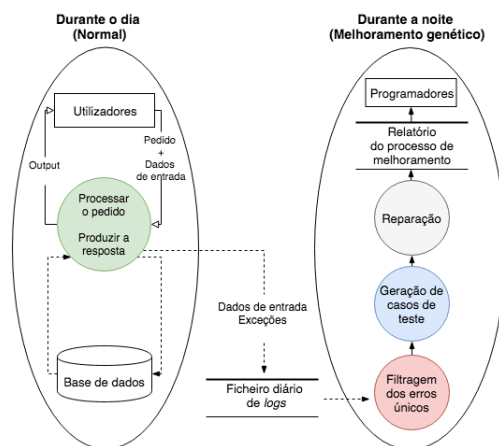


Figura 3.1: Funcionalidade do *Janus Manager* dividida em processo diurno e noturno (adaptada de [3]).

Ao início da noite, quando o último utilizador terminar a sessão é iniciado o processo de MG, lado direito da Figura 3.1. Este processo é composto por três fases de forma ordenada - filtragem, geração de casos de teste e reparação. Dada a modularidade do software, o processo de MG consegue aceder facilmente aos seus componentes, testando e melhorando cada um isoladamente.

Inicialmente, na fase de filtragem, as exceções registadas no ficheiro são ordenadas por ordem decrescente de importância, sendo que as de maior importância correspondem às exceções que aconteceram um maior número de vezes. De seguida, são gerados novos casos de teste com base nas exceções ordenadas. Para cada uma é selecionado de forma aleatória

qual, de entre os valores de entrada que desencadearam a exceção, vai ser substituído por outro que se encontre na sua vizinhança, enquanto que os restantes se mantêm fixos. Só os novos casos de teste, que causem a mesma exceção, é que são mantidos e adicionados aos já existentes no sistema.

Por fim, é aplicada PG para corrigir as falhas encontradas. Com base no conjunto final de casos de testes, a PG cria listas de edição que o código fonte deve sofrer de forma a não apresentar falhas. Cada uma das listas de edição é constituída apenas por operações de mutação (substituição, eliminação ou inserção) que podem recair sobre blocos de código, linhas ou declarações [32].

Cada população é constituída por 50 listas, sendo que de geração em geração metade são selecionadas com base na percentagem de sucesso dos casos de teste que um programa sujeito a essas alterações tem. Cada uma das 25 listas, antes de passar à geração seguinte, sofre um única mutação que pode ser de três tipos: *Grow*, *Prune* e *Single Edit Change*, como mostra a Figura 3.2 [3].

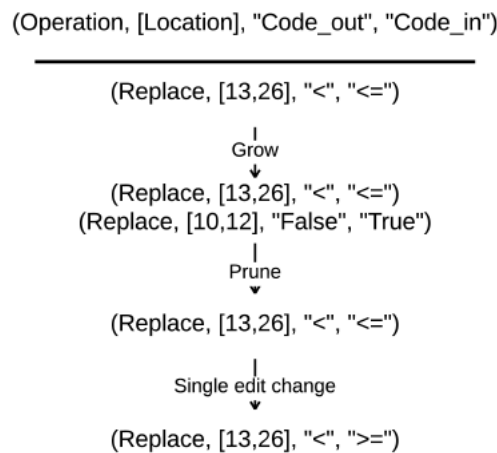


Figura 3.2: Exemplo de uma lista de edição e como ela pode evoluir com uma mutação do tipo *Grow*, *Prune* ou *Single Edit Change* (retirado de [3]).

Depois disso, a essas 25 listas são adicionadas outras tantas geradas aleatoriamente, na tentativa de eliminar a convergência para um ótimo local.

Este processo de MG só pára quando encontrar uma variante do programa que passe em todos os testes ou quando os utilizadores voltarem, na manhã seguinte, a utilizar o programa.

Deste processo resulta um relatório detalhado do que ocorreu durante a noite. Nesse relatório estão as exceções que foram encontradas, os novos casos de teste e uma lista de edição com as possíveis modificações ao programa original. Em caso de existir mais do que uma é apresentada a que reúne um menor número de alterações.

Este sistema de MG tem tido muito sucesso, tendo identificado, localizado e corrigido 22 falhas em apenas 6 meses de utilização. Além disso, nenhuma outra falha foi detetada por recurso a outros métodos durante esse período. Isto demonstra a eficiência e a facilidade de gerar novos casos de teste e a elevada capacidade que a PG tem para corrigir falhas de

código, nomeadamente em software que não tem um uso contínuo, apresentando períodos em que a taxa de utilização é baixa ou nenhuma. Isto permite afirmar, que seria possível aplicar esta estratégia de MG a softwares semelhantes e que isso seria importante para a diminuição dos custos de manutenção.

O facto de operar no código fonte, sem a necessidade de converter o programa numa representação diferente, confere-lhe a facilidade de ser aplicado a outros softwares, escritos noutras linguagens de programação, com pequenas alterações nas configurações.

GenProg

O *Genetic Program Repair* (GenProg) é um algoritmo que utiliza PG para reparar falhas em softwares reais. Este algoritmo evolui programas aplicando-lhes mutações até que a falha seja corrigida, mas mantendo sempre a sua funcionalidade, ou até que os recursos sejam consumidos. Os resultados obtidos em [4] para diferentes softwares demonstram que o algoritmo pode reparar vários tipos de falhas em programas extraídos de múltiplos domínios.

Na maioria dos casos em que a PG é utilizada para melhorias de software, o espaço de procura tem tendência a crescer infinitamente. Desta forma, o GenProg apresenta três inovações para fazer face a esse problema [33]:

- Cada variante do programa opera ao nível da declaração da AST, que inclui todas as instruções do programa, aumentando a granularidade da pesquisa;
- Cada versão do programa inclui uma lista que identifica quais são os operadores genéticos que são efetivamente executados nos casos de teste que falham, ajudando na localização da falha;
- É considerada a hipótese de que um programa que contenha um erro numa área específica do código, provavelmente implementa o comportamento correto noutra parte. Assim sendo, nas sucessivas evoluções do software são utilizadas apenas declarações do próprio código, reduzindo o espaço de procura.

Um exemplo da aplicação do algoritmo GenProg, é a sua utilização para reparar uma falha num servidor web *multithreads*, designado *nullhttpd*. Este servidor apresentava uma falha quando o comprimento do conteúdo fornecido pelo utilizador num cabeçalho HTTP das solicitações POST continha valores negativos. A falha ocorria quando a função `ReadPostData()` do ficheiro `http.c` era executada. No entanto, o processamento do pedido era também realizado no ficheiro `cgi.c`, mas onde o tamanho do cabeçalho HTTP era verificado antes do processamento. O código presente nos dois ficheiros encontra-se representado na Figura 3.3.

```

108 // http.c
109 conn[sid].PostData=
110     calloc(conn[sid].dat->in_ContentLength+1024,
111           sizeof(char));
112 pPostData=conn[sid].PostData;
113 ...
114 do {
115     rc=recv(conn[sid].socket,
116            pPostData, 1024, 0); /* overflow! */
117     ...
118     pPostData+=rc;
119 } while ((rc==1024) ||
120         (x<conn[sid].dat->in_ContentLength));
-----
267 // cgi.c
268 if (conn[sid].dat->in_ContentLength>0) {
269     write(local.out, conn[sid].PostData,
270           conn[sid].dat->in_ContentLength);
271 }

```

Figura 3.3: Código dos ficheiros `http.c` e `cgi.c` do `webservice nullhttpd` (retirado de [4]).

Desta forma, na versão corrigida do software, a chamada da função `ReadPostData()` foi substituída por o processamento de dados POST presente no ficheiro `cgi.c`. Ainda que a correção não seja a que foi implementada por os programadores na versão original, a versão resultante da aplicação do GenProg elimina a falha existente e retém a sua funcionalidade. Outros exemplos presentes no artigo [4], permitem mostrar o sucesso da aplicação do GenProg, onde num total de 16 programas aos quais foi aplicado, perfazendo um total de 1,25 milhões de linhas de código e abrangendo oito tipos de erros diferentes, revelando-se um bom auxiliar na deteção e correção de falhas, com uma taxa média de sucesso de 77%. No entanto, este algoritmo depende dos casos de teste para criar novas versões do software, o que pode ser uma limitação devido à existência de propriedades não determinísticas que são impossíveis de avaliar.

3.1.2.2 Melhoramento Genético de propriedades não funcionais

Em geral, satisfazer os requisitos não funcionais é uma tarefa difícil e, muitas vezes, só é possível por otimização dos compiladores.

A PG tem sido utilizada para aperfeiçoar software já existentes, apresentando uma estrutura capaz de produzir otimizações pouco óbvias, que os compiladores ainda não conseguem.

Software CUDA C++

Compute Unified Device Architecture (CUDA) é uma *Application Programming Interface* (API), desenvolvida pela *Nvidia*, que oferece aos programadores a capacidade de enviar código diretamente para a *Graphics Processing Unit* (GPU) sem terem de utilizar código Assembly.

Um sistema, designado *StereoCamera*, foi desenvolvido por Joe Stam [6] para a primeira versão da CUDA. Este sistema foi o primeiro a mostrar que as GPUs poderiam fornecer processamento de imagens estéreo em tempo real (mais de 30 frames por segundo) [34]. No

entanto, dada a capacidade da GPU de realizar várias operações em simultâneo foi possível criar uma nova versão do código CUDA *stereoKernel* GPU com PG, que se encontra otimizada para os pares de imagens existentes numa base de dados disponibilizada pela Microsoft [21]. Nessa base de dados existem 3010 imagens stereo em que quase todas são imagens 320x240.

A aplicação da PG seguiu uma abordagem *Genetic Improvement of Software for Multiple Objectives Exploration (GISMOE) standard*, no entanto foi usada uma gramática *Backus Normal Form (BNF)* criada a partir do código fonte, em vez de uma AST. Essa gramática não é genérica, apenas permite representar o programa ou as suas variantes, descrevendo quais são as possíveis alterações lineares permitidas no código fonte do *kernel*.

As principais definições do algoritmo de PG utilizado são as seguintes:

- **Criação da população inicial:** A criação da população inicial resulta da aplicação de uma mutação única ao código do *kernel*, aplicada essencialmente ao nível do cabeçalho e das variáveis partilhadas. Além disso, podem ser aplicadas zero ou mais alterações aleatórias ao restante código;
- **Representação dos indivíduos:** Cada indivíduo é representado por uma lista com os 12 valores das constantes, seguidos de uma lista com as mutações GISMOE que a gramática BNF deve sofrer. Essas mutações podem ser: apagar, substituir e inserir linhas de código copiadas do código fonte, tal como acontece em [4];
- **Função de fitness:** Para determinar o valor de fitness de cada indivíduo é escolhido um par de imagens [35] com o qual cada kernel modificado é testado. Esse valor é determinado com base na resposta, se está correta ou não, e depois pela comparação entre o seu tempo de execução e o tempo de execução do código original;
- **Tipo de seleção:** É feita uma seleção por truncamento, com elitismo de 50%;
- **Operadores genéticos:** Os indivíduos selecionados criam um filho através de uma mutação aleatória, que pode alterar um dos valores das constantes ou ser realizadas ao nível do código fonte, e criam outro filho por cruzamento, juntamente com outro indivíduo selecionado, tal como nas operações de cruzamento do GISMOE [22];
- **Número de gerações:** O processo termina ao fim de 50 gerações.

Como acontece em [4] e nesta abordagem GISMOE [22], após a aplicação da PG ter terminado, o melhor indivíduo da última geração é minimizado, isto é, todas as operações de mutação que se demonstrarem desnecessárias são removidas.

Para as GPUs mais antigas, *Quadro NVS 290* e *GeForce GTX 295*, foi impossível a aplicação de PG uma vez que, quando ocorriam acessos incorretos a índices da matriz, o *memcheck* não foi capaz de capturar e prevenir esses problemas, levando ao bloqueio do hardware.

Para as quatro GPUs mais recentes, *Tesla T10*, *Tesla C2050*, *GeForce GTX 580* e *Tesla K20c*, os resultados apresentaram uma aceleração significativa em comparação com o

kernel original, até mesmo quando este apresenta uma melhor configuração dos seus parâmetros. Além disso, existe uma diferença muito pequena de desempenho para imagens de dimensões diferentes daquelas que foram usadas para testes (320x240).

Resumidamente, em todos os casos onde a PG foi possível de ser aplicada, os resultados obtidos foram melhores, com uma velocidade final a variar em 5% para a GPUs mais antiga e 6,8% para a mais recente. Isto permite comprovar a utilidade da PG na criação automaticamente de novas versões do código, adaptadas a novos requisitos da StereoCamera e a novos hardwares.

Bowtie2

Dado o crescimento exponencial dos dados relativos a sequências de Ácido Desoxirribonucleico (ADN), o Bowtie2 foi um dos muitos sistemas criados para combinar e analisar essas sequências. Dada a sua elevada taxa de utilização, a disponibilidade do código fonte, a existência de casos de teste realistas, o seu tamanho e complexidade, este programa representa um sistema ideal para a aplicação e avaliação do MG das propriedades não funcionais, nomeadamente o tempo de execução. Desta forma, em [22], é apresentada a aplicação do MG ao sistema Bowtie2, seguindo uma variante da abordagem GISMOE.

A variante da abordagem utilizada é caracterizada por:

- **Representação gramatical do código fonte:** Para representar o modelo do sistema é criada automaticamente, a partir do código fonte do programa, uma gramática BNF. Esta gramática, além de permitir uma grande liberdade durante a evolução do código também impõe limites ao nível das classes, tipos, funções, estruturas de dados, tipos e nomes de variáveis e da estrutura do programa. Ainda assim, não garante que não existam outras falhas como, por exemplo, a utilização de uma variável inexistente;
- **Representação dos indivíduos:** Para representar as versões resultantes do processo de MG foi utilizada uma abordagem que tinha sido anteriormente usada para ampliar a correção de falhas [4]. Nessa abordagem, cada indivíduo é representado por uma lista ordenada de mudanças [36] a serem aplicadas à gramática. Essas mudanças podem ser de três tipos e representam-se da seguinte maneira:
 - `<for3_sa_rescomb_111><for3_sa_rescomb_69>` - A parte incremental do ciclo `for` na linha 111 que se encontra no ficheiro `sa_rescomb.cpp` é substituída pela parte incremental do ciclo `for` da linha 69 desse mesmo ficheiro.
 - `<_aligner_swsse_ee_u8_804>` - A linha 804 do ficheiro `aligner_swsse_ee_u8.cpp` é eliminada.
 - `<_aligner_result_47>+<_aligner_result_114>` - É feita uma cópia da linha 114 e esta é inserida a seguir à linha 47 no ficheiro `aligner_result.cpp`.
- **Aplicação da amostragem *O-Bin*:** Dado o elevado número de possíveis saídas do sistema, foi aplicado um particionamento *O-Bin* para os casos de teste disponíveis.

Assim sendo, cada geração de cinco sequências de teste, escolhidas de entre as 5250, deve conter:

1. uma sequência que não se pode encontrar no genoma humano
2. uma sequência que corresponde exatamente 1 vez
3. uma sequência que corresponde entre 2 e 10 vezes
4. uma sequência que corresponde entre 11 a 99 vezes
5. uma sequência que corresponde entre 100 e 200 vezes

Esta divisão desempenha um papel importante na avaliação de propriedades funcionais e não funcionais do código. Os dois primeiros casos garantem que as versões melhoradas do Bowtie2 ainda mantêm as suas propriedades essenciais e os casos de teste 3, 4 e 5 determinam quais são, de entre as versões melhoradas, as mais rápidas;

- **Função geral de fitness:** Para cada variante do programa é determinado um valor de fitness, tendo em conta as suas propriedades funcionais e não funcionais. Para tal, são seguidos os seguintes passos:
 1. **Propriedade funcional - Casos de teste:** É contabilizado o número de casos de testes concluídos sem erros de execução. Esses casos de teste podem ser resultado da versão original do programa [4, 28, 29] ou de dados de treino, neste caso, dados do projeto 1000 genomas [37];
 2. **Propriedade funcional - Algoritmo *Smith-Waterman*:** Este algoritmo realiza uma comparação completa, determinando as regiões semelhantes entre duas sequências. Em vez de olhar para a sequência total, o algoritmo compara segmentos de todos os comprimentos possíveis e otimiza a medida de similaridade;
 3. **Propriedade não funcional - Número de linhas de código executadas:** Uma versão que execute menos linhas de código é considerada melhor que outra que execute mais.
- **Análise da sensibilidade das propriedades não funcionais:** A análise da sensibilidade contabiliza o número de execuções de cada linha de código. As linhas que apresentarem um maior impacto têm uma maior probabilidade de serem modificadas com os operadores genéticos durante a evolução do programa;
- **Gestão de ciclos infinitos:** Para combater a existência de ciclos infinitos, que podem surgir durante o processo evolutivo, foi imposto um limite no tempo de execução. Assim sendo, existe um limite de 20 segundos para os primeiros quatro tipos de casos de teste e 1 minuto para o último;
- **Seleção:** Na abordagem aplicada ao Bowtie2 podem ser selecionados até metade dos indivíduos da população para passarem à geração seguinte. No entanto, todos aqueles que se encontrem abaixo do valor de fitness de seleção ou que não compilem não são selecionados para serem pais dos indivíduos da próxima geração;

- **Mutação:** A mutação deve ser efetuada a uma das linhas que sejam executadas pelo menos uma vez durante os casos de teste, onde a linha é escolhida com base na análise de sensibilidade. Para evitar indivíduos iguais ao código fonte ou entre si, sempre que algum apresentar essas características é rejeitado e o pai sofre novamente uma mutação;
- **Cruzamento:** Na variante da abordagem GISMOE utilizada, um cruzamento corresponde à concatenação de dois indivíduos. O primeiro indivíduo é escolhido de entre os da população baseado no valor de fitness e o segundo é criado a partir dos indivíduos da população que foram possíveis de compilar.
Tanto no cruzamento como na mutação, para evitar um rápido crescimento do tamanho dos indivíduos, sempre que algumas das operações presentes numa das listas de edição se anulem ou se mostrem irrelevantes são dispensadas. Além disso, tal como descrito no ponto anterior para a mutação, também no cruzamento é rejeitado um indivíduo se ele já esteve na nova geração ou na atual. Se após um dado número de tentativas o cruzamento não conseguir gerar um indivíduo único, o novo filho é criado mutando um membro adequado da população. Caso o número de pais aptos não seja suficiente para gerar a nova população, os indivíduos em falta são criados de forma aleatória, tal como foram criados os indivíduos da população inicial;
- **Limpeza da solução final:** No final da aplicação da PG, é feita uma limpeza à solução encontrada. Para tal, todas as edições presentes nessa solução que sejam excluídas tornando a versão evoluída pior, são novamente repostas, caso contrário, são eliminadas.

A PG foi aplicada a uma população constituída por 10 indivíduos, durante 200 gerações. Ao longo do processo evolutivo foi notório um elevado melhoramento do tempo de execução. No entanto, a população deixa, por várias vezes, de ter indivíduos suficientemente bons e por isso alguns têm de ser substituídos por novos, gerados aleatoriamente como os da população inicial. Ainda assim, a PG conseguiu melhorar o código do programa, encontrando uma versão melhorada que conseguiu correspondências de ADN com o genoma humano sendo, em média, pelo menos tão boas quanto as da versão original.

A versão melhorada e a original foram testadas para 200 sequências que não correspondem às usadas nos casos de teste. Para essas sequências, a versão melhorada apresentou um desempenho superior, demorando 3.9 horas, enquanto que a versão original demorou 12.2 dias, correspondendo a 74 vezes mais.

Ao melhor indivíduo encontrado foi realizada uma limpeza à lista de edições, reduzindo-a de 39 para 7, continuando ainda assim a produzir uma saída idêntica à versão antes da limpeza e apresentando uma aceleração de 77 vezes em comparação com a versão original. A criação e melhoria de programas de computador ainda são tarefas que requerem um grande esforço manual. Desta forma, a abordagem GISMOE pode ser aplicada a outros programas para otimizar propriedades não funcionais, ainda que o sucesso possa não ser tão grande. Para tal, é necessário encontrar uma maneira mais simples de avaliar os indivíduos de uma população e tornar o processo menos dispendioso.

3.1.3 Programação Genética baseada em gramática

A PG ainda que seja uma técnica que facilmente se adapta a diferentes contextos, não consegue impedir que indivíduos inválidos sejam criados. Desta forma, a associação de uma gramática a esta técnica pretende evitar este problema, forçando os indivíduos criados a serem válidos. Uma gramática corresponde a uma definição de como as árvores de derivação devem ser construídas e de como as operações genéticas devem ser feitas, melhorando a definição do espaço de procura.

3.1.3.1 Evolução Gramatical

A Evolução Gramatical (EG) é uma forma de PG baseada em gramática, que permite a geração de programas de computador em qualquer linguagem de programação [38]. O seu objetivo é encontrar um programa ou um fragmento dele que resolva um dado problema. Em [18] é explorada a evolução de um código escrito em Python para ordenar de forma crescente uma lista de números inteiros.

- **Casos de treino:** Os casos de treino utilizados correspondem a listas desordenadas de tamanho variado. Cada lista é composta por valores inteiros, gerados com um `range` em Python, que varia entre 0 e o tamanho da lista;
- **Função de fitness:** O valor de fitness de cada variante do algoritmo é dado pela soma do número de pares em ordem na lista de saída, como mostra a Figura 3.4, a dividir por o número total de pares possíveis nessa mesma lista, subtraída ao número total de casos de treino. Quanto menor for o valor de fitness, melhor é o indivíduo;

1	0	2	3	4	5	6	7	9	8	(n° pares em ordem = 0)
1	0	2	3	4	5	6	7	9	8	(n° pares em ordem = 1)
1	0	2	3	4	5	6	7	9	8	(n° pares em ordem = 2)
1	0	2	3	4	5	6	7	9	8	(n° pares em ordem = 3)
1	0	2	3	4	5	6	7	9	8	(n° pares em ordem = 3)
...										

Figura 3.4: Exemplo da contabilização do número de pares em ordem.

- **Gramáticas:** As duas gramáticas consideradas são constituídas por funções e símbolos terminais que compõem o conjunto de primitivas da linguagem. Esse conjunto é constituído por `{i, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, for, if, +, -, >, <}` e por as listas de entrada (`x`) e de saída (`guess`). A primeira gramática é constituída por sequências de ciclos e a segunda é igual à primeira, mas também permite a existência de ciclos encadeados. Nesta última gramática, ao conjunto de primitivas é adicionada a variável `j`, que controla o ciclo encadeado. O objetivo é determinar se a EG pode evoluir soluções para o problema de ordenamento de inteiros, usando uma sequência de ciclos e/ou ciclos encadeados;

- **Parâmetros da PG:** Por cada gramática foram realizadas 100 execuções independentes com uma população composta por 500 indivíduos, durante 100 gerações. Além disso, foi utilizada uma probabilidade de cruzamento de ponto único de 0.9 forçado a ocorrer em regiões específicas e a taxa de mutação foi de 0.1. A seleção foi feita por torneio com uma proporção populacional de 0.01 e elitismo com uma proporção de 0.1.

Foram realizados seis conjuntos de experiências, cada um com diferentes configurações. Utilizando os parâmetros que se encontram descritos anteriormente para o primeiro conjunto de experiências, é possível afirmar, com 95% de certeza, que não existe uma diferença significativa na média dos valores da função de fitness dos melhores indivíduos entre as duas gramáticas. No entanto, no segundo conjunto de experiências, em que foram utilizadas apenas as constantes 0 e 1 reduzindo o espaço de procura, é possível afirmar com a mesma percentagem de certeza que existe uma diferença significativa. Essa diferença está no desempenho dos ciclos encadeados, no entanto, a solução para o problema não é encontrada.

No terceiro conjunto de experiências, são permitidas várias declarações dentro do corpo dos ciclos. Ainda assim, não são apresentados quaisquer dados sobre esta experiência.

No quarto conjunto de experiências, o tamanho da população foi aumentado para 10 mil indivíduos. Comparando este conjunto ao primeiro, é notório, com 95% de certeza, que uma população maior cria uma diferença estatisticamente significativa no desempenho, no entanto, a solução para o problema não é encontrada.

O quinto conjunto foi realizado utilizando uma gramática modificada, onde os pontos de cruzamento são apenas expressões de condição. Mesmo assim, não foram notados quaisquer ganhos de desempenho.

No sexto e último conjunto, a cada gramática foi adicionada uma função de troca, responsável pela troca dos valores das posições i e j na lista de saída. Desta vez, a gramática com ciclos encadeados conseguiu resolver o problema com êxito, mas a gramática com ciclos em sequência não.

Estes resultados permitem demonstrar a dificuldade que um algoritmo de PG tem na resolução de um problema simples de ordenação de números inteiros, usando um conjunto primitivo de funções e símbolos terminais. Depois das experiências realizadas com duas gramáticas e tamanhos populacionais diferentes, é possível afirmar que a inclusão de uma função de troca permitiu ao algoritmo de ciclos encadeados solucionar o problema, utilizando o tamanho padrão da população (500 indivíduos). Além disso, tal como em estudos anteriores, existe uma dificuldade de desenvolver um algoritmo de ordenamento geral, com primitivas de baixo nível.

3.1.3.2 Programação Genética guiada por gramática

Dada a flexibilidade da PG para evoluir soluções, esta é utilizada na resolução de uma vasta gama de problemas. No entanto, ela não requer uma representação específica, como

árvores ou representações lineares, existindo diferentes representações que se adaptam melhor ou pior a diferentes tipos de problemas.

Os sistemas de Programação Genética Guiada por Gramática (P3G) são independentes de uma representação do problema, uma vez que as soluções produzidas apenas exigem que a sua sintaxe seja definida por uma gramática. Estes tipos de sistemas parecem ser adequados para problemas de síntese de um programa por três motivos:

- Já existem gramáticas definidas para as várias linguagens de programação;
- Problemas semelhantes têm de ser resolvidos em múltiplas linguagens de programação, tornando-se fácil a adaptação de uma gramática entre linguagens;
- As gramáticas podem ser usadas em todos os tipos de sistema de PG sem que seja necessário a sua re-implementação, o que facilita a realização de experiências.

Stefan Forstenlechner *et al.*, em [19], abordam problemas de síntese de um programa, utilizando um sistema padrão P3G baseado em árvores [39]. Este sistema apresenta algumas adaptações para que se torne o mais flexível possível, permitindo desenvolver código em qualquer linguagem de programação e para que seja extensível ao uso de bibliotecas.

Esta abordagem consiste em duas partes que devem ser fornecidas e adaptadas consoante a linguagem de programação do código que se pretende evoluir. Essas duas partes são descritas de seguida:

- **Gramática:** A gramática permite definir a aparência dos indivíduos válidos e o espaço de procura, uma vez que nem todos os problemas requerem os mesmos tipos de dados. Desta forma, deve ser desenhada de maneira a que os indivíduos que não compilem ou que levem a exceções não sejam produzidos. Além disso, deve ser facilmente adaptável, permitindo que bibliotecas que estejam disponíveis sejam usadas na resolução do problema. No entanto, o uso de uma gramática pode trazer algumas desvantagens, uma vez que os indivíduos consistem apenas em partes definidas da gramática e, portanto, os nomes de todas as variáveis devem ser definidos previamente (variáveis de entrada, variáveis de saída e variáveis intermédias). Nesta abordagem, é criar uma gramática para cada tipo de dados e estas são combinadas para o problema em questão;
- **Esqueleto:** O esqueleto é a parte do *design patterns* que é executada no final para avaliar os indivíduos. No esqueleto encontram-se definidos o cabeçalho do método que contem o código que vai ser evoluído, a função de fitness e os métodos protegidos.

O cabeçalho do método define o tipo de retorno, bem como as variáveis de entrada. O corpo do método é substituído pelo o código de um indivíduo e é executado para determinar o seu valor da função de fitness. A função de fitness faz parte do esqueleto em vez de fazer parte do sistema de PG. Os métodos protegidos, que representam a parte mais importante do esqueleto, fornecem segurança na avaliação de um indivíduo, tal como as

operações de divisão protegidas fornecem segurança em problemas de regressão utilizando PG. Embora um indivíduo seja sintaticamente correto e muitos dos métodos possam ser substituídos por versões protegidas, nada garante que exceções relacionadas com o tempo de execução e avaliações demoradas não possam ocorrer. Para evitar isso, é utilizado um parâmetro que define o tempo limite. Além disso, cada ciclo contém um bloco de código que verifica o número de iterações executadas e quando esse número atinge um determinado valor a execução é forçada a terminar. Sempre que não é possível avaliar um indivíduo, é-lhe atribuído o pior valor possível de fitness.

O PushGP é um sistema de PG que utiliza a linguagem Push. A linguagem Push foi concebida para Computação Evolucionária (CE), sendo utilizada para representar os programas em evolução. Dada a sua arquitetura em pilha, existindo uma pilha para cada tipo de dados, não é necessário a declaração de variáveis e sempre que um determinado tipo de dados não estiver disponível, a instrução simplesmente não é executada.

Para testar esta abordagem foi utilizado um conjunto de *benchmark* composto por 29 problemas e os resultados foram comparados com os resultados do PushGP. Para cada um desses problemas são combinadas as gramáticas dos tipos de dados necessários sem qualquer adaptação adicional.

Os parâmetros típicos de PG utilizados na experiência são os mais próximos possíveis das configurações do PushGP e encontram-se representados na Tabela 3.1.

Parâmetro	Valor
Número de execuções	100
Número de gerações	300
Tamanho da população	1000
Tamanho do torneio	7
Probabilidade de cruzamento	0.9
Probabilidade de mutação	0.05
Tamanho da elite	1
Número limite de nós	250
Variáveis por tipo	3
Tempo máximo de execução	1s

Tabela 3.1: Valores dos parâmetros de configuração de Programação Genética quando aplicada ao P3G.

Além disso, foram utilizados dois operadores de seleção, seleção por torneio e seleção *lexicase* [40], que também são usados no PushGP.

Os resultados do P3G são comparados aos resultados do PushGP presentes em [41].

Na seleção por torneio, o PushGP conseguiu resolver cinco problemas para os quais o P3G não conseguiu encontrar a solução correta e o P3G resolveu dois problemas que o PushGP não conseguiu. Além disso, em muitos casos em que o P3G conseguiu encontrar uma solução, o número de soluções corretas encontradas por 100 execuções foi consideravelmente maior do que com o PushGP. No entanto, a diferença do número de soluções encontradas é bastante pequena para problemas em que o PushGP se demonstrou melhor que o P3G.

Na seleção *lexicase*, o P3G só é capaz de resolver um problema que o PushGP não é, no entanto, o PushGP é capaz de resolver sete que o P3G não consegue. Além disso, nos

problemas que ambos os sistemas conseguem resolver, o P3G é capaz de encontrar soluções mais corretas.

Com isto, é notório que ambos os sistemas conseguem resolver mais problemas com a seleção *lexicase* do que com a seleção por torneio e que, na maioria das vezes, as soluções encontradas são mais corretas.

Analisando em que gerações é que as soluções corretas foram encontradas, é possível afirmar, que com um número reduzido de gerações, o P3G encontra resultados muito semelhantes para os dois tipos de seleção. Além disso, a pequena quantidade de indivíduos inválidos para P3G acontece devido ao tipo de segurança das gramáticas, bem como aos métodos protegidos presentes no esqueleto.

Posto isto, é possível concluir que a P3G é uma boa abordagem para os problemas gerais de síntese de programas, onde gramáticas podem combinar-se para se adequar a um problema específico. Essas gramáticas podem ser usadas na maioria dos sistemas de PG baseada em gramática sem terem de ser ajustadas a um problema em específico.

3.1.4 Geração de software com Programação Genética

O processo de desenvolvimento de software tem tido alguns progressos no decorrer dos anos, no entanto, ainda continua a depender, em grande parte, da mão humana. Desta forma, vários estudos têm sido realizados com vista a automatizar o processo de geração de software.

Em 2014, Harman *et al.* [20] apresentam evidências empíricas que crescer e enxertar são operações realizáveis, que permitem aumentar o código de um sistema introduzindo-lhe novas funcionalidades. Essas operações consistem no seguinte:

- **Crescer:** Esta operação consiste na aplicação de PG a fragmentos de código para o fazer crescer, guiado por sugestões que são fornecidas por programadores. Essas sugestões podem conter nomes de funções de uma biblioteca que um programador ache importante e restrições de ordenamento parcial sobre onde essas funções devem ser chamadas. Desta forma, esta operação acaba por ter intervenção humana para restringir o espaço de procura. No entanto, o resto do processo é totalmente automatizado, não sendo necessária qualquer especificação sobre, por exemplo, nomes de variáveis ou expressões.
- **Enxertar:** Da operação anterior resulta um fragmento de código que executa uma funcionalidade específica de forma totalmente isolada. Assim sendo, é necessário enxertar esse fragmento com a nova funcionalidade num sistema já existente. Para isso, é necessário determinar o(s) local(ais) onde deve(m) ser feita(s) a(s) inserção(ões) e determinar as expressões necessárias para criar a ligação entre o sistema existente e o código da nova funcionalidade.

Estas duas operações foram aplicadas a um programa do mundo real, designado Pidgin. Este sistema permite a troca de mensagens instantâneas, tendo cerca de vários milhões de

utilizadores a nível mundial.

A este programa foram feitas a adição de duas novas funcionalidades. A primeira funcionalidade é meramente ilustrativa e a segunda já representa o enxerto de um fragmento maior de código, permitindo mostra a aplicabilidade das duas etapas na adição de novas funcionalidades.

O primeiro fragmento de código enxertado no sistema, permitia que as mensagens de todos os utilizadores incluíssem o tempo restante até ao congresso dedicado à Engenharia de Software Baseado em Pesquisa, desse mesmo ano (2014). Este fragmento de código foi escrito por um programador que não se encontrava familiarizado com o Pidgin. O fragmento recebia como parâmetro de entrada a hora atual e devolvia uma cadeia de caracteres com o número de dias que restavam até ao congresso. Dada a simplicidade desta funcionalidade, o código tinha apenas uma linha o que faz com que a operação crescer não fosse significativa. No entanto, a operação de enxertar foi realizada de forma totalmente automática, identificando os possíveis sítios para o enxerto e as formas de conexão entre o código do sistema e o novo fragmento.

Cada variante do código foi compilada, executada e submetida aos casos de testes, sendo que apenas duas variantes é que passaram em todos os testes. É de notar que um enxerto num sistema pequeno foi realizado de forma automática, mas esta operação pode ser um grande desafio quando aplicada a sistemas maiores.

A segunda funcionalidade adicionada, designada *Babel Fish*, corresponde a um fragmento de código substancialmente maior, em que o objetivo é traduzir as mensagens de Inglês para Português ou Coreano. Uma vez que o sistema é fortemente constituído por chamadas e atribuições de funções, para crescer o fragmento de código foi utilizada PG baseada em gramática. Nessa gramática é especificada uma lista dos tipos de dados e das funções, que foram sugeridas por um programador como sendo as mais úteis. Assim sendo, para o crescimento do *Babel Fish* estavam incluídas nas sugestões o nome da API do GoogleTranslate para Português e Coreano, juntamente com alguns nomes de funções de bibliotecas de processamento de strings (`concat` e `strlen`). Estas sugestões são uma grande ajuda para a PG, que dificilmente iria utilizar o GoogleTranslate e as funções de bibliotecas externas. A aplicação da PG para fazer crescer o código é feita utilizando um cruzamento de ponto único com uma probabilidade de 0.8. Após o cruzamento, é aplicada uma de três mutações possíveis, escolhida com base numa probabilidade uniforme. Essas operações de mutação são as seguintes: substituição de uma variável, substituição de uma declaração e troca de uma declaração. Além disso, é utilizada uma seleção de elitismo agressiva, onde o melhor indivíduo de uma geração é replicado e inserido na geração seguinte, entre 1 a 250 vezes, seguindo uma distribuição *Coupon Collector* com uma média de 197. Esta abordagem de elitismo permite garantir que um esquema promissor do código é mantido ao longo das gerações. Cada população é constituída por 500 indivíduos e a PG só termina quando o melhor indivíduo permanecer inalterado durante 20 gerações.

A aplicação da PG ao fragmento de código foi repetida 30 vezes para 8 funções de fitness diferentes. Essas diferentes funções de fitness permitem entender a relação de compromisso entre o esforço de um programador e o esforço da PG, determinando a informação

necessária que teria de ser fornecida *à priori* - propriedades necessárias para um boa solução, nomes de funções de várias bibliotecas, restrições da ordem parcial das chamadas feitas durante a execução, entre outras. No entanto, mesmo que um programador tivesse de fornecer toda essa informação, o esforço exigido dele seria relativamente baixo, mais baixo do que o necessário para escrever o código da nova funcionalidade a partir do zero e inseri-lo no código.

Estes resultados constituem uma prova de que aos sistemas reais podem ser adicionadas novas funcionalidades com uma simples orientação humana.

Uma das versões criadas pela PG, que exigiu um esforço por parte do programador, foi enxertada automaticamente nos 46 espaço possíveis no código. Desses 46 enxertos, 17 foram realizados com sucesso e, portanto, são igualmente bons. Isto permite mostrar que a operação de enxertar é fácil, eficaz e não tem uma solução única.

Num mundo ideal, o objetivo passaria pela utilização da PG em todo o processo de crescimento do software, sem que fosse necessário qualquer esforço por parte de um programador. Mesmo assim, a existência desse esforço serve apenas para oferecer algumas sugestões que permitem reduzir o espaço de procurar e encaminhar o software para uma boa solução, devendo essa informação ser a menor possível, mas a suficiente para orientar a PG. Além disso, é mais uma vez demonstrada a importância do MG, desta vez para o desenvolvimento de características de código isoladamente, em grande parte inconscientes do sistema em que vão ser posteriormente enxertadas, sendo apenas necessária uma pequena orientação humana e conhecimento do domínio.

3.2 Trabalhos realizados sobre previsão e otimização do consumo de combustível

Nesta secção, são apresentados alguns estudos realizados nos últimos anos sobre a previsão e otimização do consumo de combustível com base em diferentes fatores, utilizando técnicas de Inteligência Artificial (IA). A diversidade dos modelos criados que mapeiam o consumo de combustível está associada a uma elevada quantidade de parâmetros que podem influenciar direta ou indiretamente esse consumo. Quando o consumo de combustível passa a ser influenciado por outros fatores que o modelo não tem em conta, acontecem discrepâncias entre os valores previstos e os reais, tal como tem acontecido ao longo dos anos para os modelos fornecidos pelas empresas de construção automóvel, que criam o modelo em laboratório sem ter em conta diversos fatores como, por exemplo, a temperatura do ar ou a pressão dos pneus.

3.2.1 Otimização do consumo de combustível com Programação Genética

Com as limitações associadas à emissão de gases de efeito de estufa e o aumento do preço dos combustíveis é importante reduzir o consumo do veículo. A solução ideal seria circular sempre a uma velocidade constante [7], no entanto, esse cenário é irrealista e qualquer

solução deverá atender aos diferentes perfis de estrada. Vários estudos já demonstraram que a altitude tem influência no consumo de um veículo, onde cerca de 40% da energia produzida pelo motor é gasta para contrariar a força da gravidade [42].

Em [43], Sina Torabi e Mattias Wahde tentaram determinar a velocidade ideal a que um veículo deveria circular levando em consideração o troço de estrada em que se encontra, algo que já tinha sido avaliado em várias simulações [44]. Para tal, foram considerados dois tipos de perfis:

- **Perfil da estrada:** No perfil de estrada é apenas considerado o movimento longitudinal do veículo. Assim sendo, para qualquer posição do veículo, existe um índice que indica em que secção de estrada é que se encontra e a respetiva altitude;
- **Perfil de velocidade:** Para qualquer posição do veículo existe um índice que indica em que secção de estrada é que se encontra e qual a respetiva velocidade.

Assim sendo, a otimização do consumo de combustível pode ser reduzido a um problema de ajuste de um perfil de velocidade a um perfil de estrada. De forma a poder comprova-lo, o algoritmo de otimização foi aplicado a 10 perfis de estrada diferentes, cada um com 10 km de extensão, onde o consumo de combustível foi medido.

Os resultados obtidos mostram que a utilização de perfis de velocidade otimizados funciona igualmente bem quando aplicados em contexto real. As melhorias foram na ordem dos 10.2%, enquanto que nas simulações foram de 11.5%. Essa pequena discrepância deve-se a fatores como, por exemplo, o efeito do vento frontal, que nas simulações não é tido em conta.

Com base nos resultados apresentados é possível concluir que:

- A representação por perfis de estradas e de velocidades utilizados tem um efeito positivo no desempenho do algoritmo de otimização, permitindo encontrar melhores soluções mais rapidamente;
- A validação dos modelos em contexto real é fundamental;
- A altitude é um fator determinante no valor de consumo do veículo.

Contudo, estes não são os únicos perfis estudados para otimizar o consumo de combustível. Em [45], é otimizado o perfil de condução para obter uma redução do consumo de combustível consoante o vento frontal e a informação de trânsito ao longo de todo o percurso. Esta abordagem transforma a questão num problema de otimização em tempo real incluindo variações dinâmicas de tráfego. Para tal, foi considerado uma abordagem multi-objetivo, em que a decisão sobre qual o melhor indivíduo a escolher envolve a otimização simultânea de mais do que uma função de fitness.

Os resultados obtidos mostram que o aumento do vento frontal, aumenta o tempo necessário para completar o percurso e, conseqüentemente, o consumo. No entanto, a combinação de todos os fatores externos como, por exemplo o vento frontal e o atrito, que

limitam a velocidade do veículo, obrigando-o a deslocar-se a uma velocidade constante, representa uma boa estratégia para minimizar o consumo de combustível. Além disso, para otimizar o consumo em condições de tráfego intenso, o veículo deve reagir ao movimento dos restantes veículos individualmente.

Concluindo, este estudo permite demonstrar que os fatores externos como, por exemplo, a direção e a velocidade do vento, influenciam o consumo de igual forma que os fatores internos. Posto isto, estes devem ser tidos em conta nos modelos criados para que consigam mapear de forma precisa a realidade.

3.2.2 Previsão do consumo de combustível com Redes Neuronais Artificiais

Em [16], Adnan Parlak *et al.*, tentaram prever o consumo de combustível e a temperatura de exaustão para um motor a diesel com Redes Neuronais Artificiais (RNA). Para tal, foram recolhidos dados de um motor do tipo Ricardo E6.

Os dados recolhidos continham informações sobre a velocidade do motor, a pressão efetiva média de travagem, o tempo de injeção, o consumo de combustível e a temperatura de exaustão. Estes incluem 80 valores, onde 60 foram utilizados para treino e os restantes foram selecionados aleatoriamente para testes, sendo que todos os valores foram normalizados no intervalo de 0.1 a 0.9.

A utilização de RNA permite testar a capacidade de previsão do consumo de combustível e a temperatura de exaustão específicos para um dado motor. Estas utilizaram um algoritmo de aprendizagem *back propagation (BP)* com uma configuração 3-7-2, dada a sua popularidade em aplicações de engenharia. Esta rede possui uma camada de entrada (com 3 neurónios), uma camada oculta (com 7 neurónios) e uma camada de saída (com 2 neurónios). Assim sendo, para a RNA criada os dados de entrada são a velocidade do motor, a pressão efetiva média de travagem e o tempo de injeção e os dados de saída são o consumo de combustível e a temperatura de exaustão.

Quando os dados de treino são apresentados à rede, os pesos são ajustados de forma a minimizar o erro entre o valor de saída e o valor real.

No período de treino, tanto para o consumo como para a temperatura de exaustão o nível de erro foi satisfatório, sendo o erro médio inferior a 1.5%, o que mostra que a rede conseguiu aprender com exatidão os dados. Durante a fase de teste, o erro foi de 1.93% na previsão do consumo de combustível e de 2.36% na previsão da temperatura de exaustão, mostrando uma capacidade preditiva igualmente satisfatória.

A previsão destes dois parâmetros com uma RNA, utilizando um algoritmo de aprendizagem *back propagation (BP)* com uma configuração 3-7-2, apresentaram resultados com um erro muito pequeno. No entanto, esta abordagem é geralmente demorada para que consiga desenvolver a melhor configuração na fase de treino, ainda que consiga aprender e realizar previsões com precisão.

Em [46], Togun e Baysec, desenvolveram um modelo matemático com PG para prever o binário do motor a gasolina e o consumo de combustível específico de travagem. A pre-

visão desses valores é feita em função de *spark advance*, posição do acelerador e velocidade do motor. O *spark advance* é quando, à medida que as rotações do motor aumentam, é necessário que a faísca aconteça cada vez mais cedo.

O desempenho dos modelos gerados com PG foi comparado ao desempenho dos modelos de RNA desenvolvidos pelos mesmo autores em [47].

Para testar e treinar os modelos foram recolhidos dados de um motor do tipo Fiat Tofaş. Desta forma, para cada conjunto de dados de entrada, *spark advance*, posição do acelerador e velocidade do motor, foram registados os valores do binário do motor e do consumo específico de combustível. Os dados foram posteriormente separados em parte de treino, com a qual os modelos foram gerados, e parte de teste, que permitiram avaliar a capacidade de previsão dos modelos.

Os resultados conseguidos mostram que os modelos gerados com PG aprenderam rapidamente a relação entre as variáveis do sistema. Além disso, para os dados de teste foi notória uma grande capacidade de previsão. Quando comparados com a RNA apresenta em [47], em que os mesmos dados de treino e teste foram utilizados, foi notório que ambas as técnicas apresentam igualmente bons resultados.

Concluindo, este estudo permite demonstrar que a PG pode ser efetivamente utilizada para obter modelos não-lineares complexos, com resultados igualmente bons aos obtidos pelo modelo de RNA apresentado em [47], dado a facilidade com que ambos os modelos apresentaram resultados muito próximos dos reais.

3.2.3 Previsão e otimização do consumo de combustível e de emissão de gases com Redes Neurais Artificiais e Algoritmos Genéticos

Os limites admissíveis de emissão de gases libertados pelos veículos são cada vez mais baixos. Para cumprir esses novos limites a indústria de motores a diesel sofreu um grande desenvolvimento, criando novas estratégias que tentam atender simultaneamente aos novos limites impostos e aos requisitos de consumo de combustível [48].

Em [49], José M. Alonso *et al.* tentaram estudar a viabilidade do uso de RNA em conjunto com Algoritmos Genéticos (AG) para otimizar o consumo de combustível em função das configurações dos motores a diesel definidas pelos fabricantes.

Para tal, foram utilizadas RNA para modelar a emissão de gases e o consumo de combustível específico de travagem, através da combinação dos parâmetros de operação do motor. Depois disso, foram aplicados AG para conseguir otimizar o consumo de combustível, mantendo as emissões dentro dos limites permitidos.

Para a realização da experiência foi recolhida uma grande quantidade de dados, para a criação de casos de testes representativos (440 casos de teste). Além disso, foram medidas as emissões de gases para os diferentes valores dos parâmetros do motor e definidos os limites de emissão de gases.

Inicialmente, foram construídas diferentes RNA (uma para cada saída possível) em vez de ter sido criada uma geral. A arquitetura de cada rede foi baseada no *Multilayer Perceptron* (MLP), com apenas uma camada oculta. O desenvolvimento desta abordagem seguiu os

seguintes passos:

- **Processo de treino:** O processo de treino é dividido em três etapas:
 1. **Divisão aleatória em conjunto de treino e de teste:** Esta etapa foi realizada 50 vezes, fazendo diferentes divisões em cada um das execuções. Além disso, foi feita uma média do erro de treino, que será utilizada nas etapas seguintes;
 2. **Avaliação do número de neurónios na qualidade de previsão:** Nesta etapa os casos de teste foram divididos em 90% para treino e 10% para avaliação da previsão. Desta forma, cada rede foi treinada até que o erro médio de previsão alcançasse o erro calculado na etapa anterior;
 3. **Treino da rede** - A rede foi treinada usando todos os conjuntos de teste até que o erro médio atingisse o erro calculado na etapa 1.
- **Transformação dos dados:** Como existem grandes diferenças nos valores absolutos das possíveis entradas, o processo de aprendizagem de uma RNA pode ser dificultado. Desta forma, todas as entradas e saídas foram previamente normalizadas;
- **Arquitetura da RNA:** A seleção do número exato de neurónios na camada oculta para cada saída foi baseada na exatidão dos resultados, seguindo um procedimento de tentativa-erro. Este ajuste permitiu que a complexidade do problema modelado fosse tida em consideração;
- **Qualidade preditiva da RNA:** Antes da fase de otimização, a qualidade de previsão das redes foi avaliada, sendo que as que permitiram prever o consumo de combustível apresentam uma boa previsão e uma baixa dispersão entre elas. No entanto, as previsões de emissões de NOx foram as que apresentaram os resultados mais exatos.

Como a otimização baseada em todos os parâmetros do motor não era viável, devido a um grande número de parâmetros envolvidos, o processo teve de ser simplificado.

O processo de otimização do modelo foi realizado utilizando um AG. Desse algoritmo é importante reter as seguintes configurações.

- **População:** Para representar os indivíduos da população foi utilizado um vetor de sete valores reais, que se encontram dentro de certos limites definidos. A população foi criada com um número de indivíduos entre 100 e 150.
- **Função de fitness:** Como o objetivo era reduzir o consumo de combustível, ainda que sujeito a certas limitações dos parâmetros operacionais e aos limites das emissões de gases, a função de fitness tem em conta o valor das emissões e quanto maior for esse valor, maior é a penalização dada ao indivíduo. Essa penalização aumenta o valor de fitness, reduzindo a probabilidade desse indivíduo ser selecionado para fazer parte da população seguinte.

- **Seleção, Cruzamento e Mutação:** Neste caso, quanto mais baixo for o valor de fitness, melhor é o indivíduo. Ao longo das gerações, os diferentes indivíduos são sujeitos a cruzamentos e a mutações, sendo que estas últimas apresentam uma probabilidade alta para introduzir diversidade à população.

Os resultados mostram que a conjugação destas duas técnicas, permitiu encontrar melhorias importantes, reduzindo as emissões de gases e o consumo de combustível. Isto aconteceu sem que fosse necessário o recurso a novos dispositivos, apenas alterando os valores dos parâmetros operacionais considerados pelos fabricantes. Além disso, é possível concluir que as RNA não suportam uma grande quantidade de variáveis de entrada.

3.3 Conclusões

Os trabalhos apresentados permitem mostrar que a PG *standard* com pequenas variâncias e/ou em conjunto com outras estratégias tem sido amplamente utilizada não só para melhorar, como também para gerar novos modelos capazes de representar realidades tão complexas e dispares entre si.

Assim sendo, as direções futuras passam por explorar o comportamento da técnica de PG na criação de um modelo capaz de prever uma realidade que aparenta ser demasiado complexa. Para tal, é possível reter alguns aspetos importantes dos trabalhos apresentados anteriormente, referentes à técnica de PG:

- **Abordagem utilizada no JM:** Esta abordagem demonstra que um modelo criado com PG tem a capacidade de se ajustar à medida que novos dados sobre o contexto analisado vão surgindo, sendo uma mais valia quando o problema depende de um elevado número de variáveis;
- **Correta definição dos parâmetros de PG e dos conjuntos de funções e símbolos terminais:** É de salientar a importância de ajustar os parâmetros e os conjuntos de funções e símbolos terminais ao contexto do problema para se conseguir obter bons resultados. A sua incorreta definição pode impedir o algoritmo de encontrar uma solução ótima para o problema em análise;
- **Fase de pré-processamento de seleção de atributos desnecessária:** A PG faz a seleção de atributos por si só, não sendo obrigatória uma fase de pré-processamento de seleção de atributos para a obtenção de melhores resultados. No entanto, em problemas de elevada complexidade a sua realização facilita o processo de procura, tornando o algoritmo computacional mais leve;
- **Introdução de alterações no algoritmo de PG:** A introdução de pequenas alterações pode evitar algumas limitações e ajudar na procura de uma solução ótima.

Também é possível salientar alguns aspetos importantes referentes ao problema de previsão do consumo de combustível:

- **O consumo de combustível depende de uma grande quantidade de fatores:**
No consumo de combustível, no limite, dependerá da reunião dos diferentes fatores referidos nos estudos apresentados anteriormente, correspondendo a uma quantidade elevada de fatores internos e externos ao funcionamento do veículo;
- **Os modelos já existentes dependem de um número reduzido de fatores:**
Dependem de não mais de três ou quatro fatores por questões de simplificação;
- **Os modelos existentes são criados em condições ideais:** Aos serem gerados em condições ideais apresentam elevados valores de erro quando testados em contexto real;
- **A PG é maioritariamente utilizada para otimizar modelos já existentes:**
Na maioria dos casos os modelos são gerados com RNA tendo de ser simplificados devido ao tempo necessário para o treino da rede.

Esta página foi propositadamente deixada em branco.

Capítulo 4

Caracterização dos dados do estudo

Neste capítulo são descritas detalhadamente as variáveis que compõem os diferentes conjuntos de dados e as abordagens utilizadas para selecionar quais delas é que devem ser incluídas no conjunto de variáveis de entrada do algoritmo de PG.

Os dados descritos estão associados às características e ao funcionamento dos veículos, fornecidos pela empresa Stratio Automotive, e a dados meteorológicos, obtidos por recurso a uma *Application Programming Interface* (API).

4.1 Dados do veículo

O primeiro conjunto de dados foi fornecido pela empresa numa fase inicial, correspondendo a uma viagem de 11.6 quilómetros com uma duração de aproximadamente 38 minutos, realizada no dia 28 de Janeiro de 2017. Para a viagem em questão existem 878 entradas, recolhidas em intervalos de tempo não constantes, sendo que cada uma delas é caracterizada por valores de 37 variáveis, que representam as características do veículo e as condições do seu deslocamento, e pelo consumo de combustível acumulativo, em litros por 100 quilómetros, estimado pelo modelo matemático gerado manualmente pela empresa. As variáveis que foram incluídas são as seguintes:

1. **Tempo de funcionamento do veículo (segundos)** - unidade temporal expressa em segundos;
2. **Distância (metros)** - distância percorrida pelo veículo, expressa em metros. A distância percorrida em $t = i$ é dada por:

$$d_{t=i} = d_{t=i-1} + \Delta d_i \quad (4.1)$$

onde $d_{t=i-1}$ corresponde à distância percorrida em $t = i - 1$ e Δd_i à variação do espaço entre $t = i - 1$ e $t = i$;

3. **Altitude (metros)** - distância medida na vertical entre o ponto atual e o nível médio do mar, expressa em metros;
4. **Altura (metros)** - diferença entre a altitude no instante atual e a altitude mínima atingida em todo o percurso. A altura é dada por:

$$altura_{t=i} = altitude_{t=i} - altitude_{min} \quad (4.2)$$

onde $altitude_{t=i}$ corresponde à altitude no instante $t = i$ e $altitude_{min}$ corresponde à altitude mínima atingida ao longo de todo o percurso;

5. **Velocidade Média (m/s)** - velocidade média do veículo, expressa em metros por segundo, é dada pelo espaço percorrido em função do tempo. Este valor dá a indicação da rapidez com que o espaço percorrido foi realizado e corresponde à seguinte expressão:

$$v_m = \frac{\Delta d}{\Delta t} \quad (4.3)$$

onde Δd corresponde à variação do espaço e Δt corresponde à variação do tempo;

6. **Aceleração (m/s²)** - a aceleração do veículo corresponde à taxa de variação da velocidade em função do tempo, expressa em metros por segundo ao quadrado. Em outras palavras, a aceleração indica o aumento ou a diminuição da velocidade com o passar do tempo e é dada pela seguinte expressão:

$$a = \frac{dv}{dt} \quad (4.4)$$

onde dv corresponde à derivada da velocidade e dt corresponde à derivada do tempo;

7. **Velocidade de derrapagem (km/h)** - a velocidade de derrapagem corresponde à velocidade máxima com a qual o veículo pode efetuar a curva em que se encontra sem derrapar, expressa em quilômetros por hora;
8. **C_x (adim)** - coeficiente adimensional utilizado para quantificar a força da resistência do ar segundo o eixo dos xx ;
9. **μ (adim)** - coeficiente adimensional utilizado para quantificar o atrito lateral exercido nos pneus do veículo;
10. **Relação de Transmissão do veículo** - relação entre a velocidade de rotação do motor e a velocidade de rotação das rodas do veículo;
11. **Número de rotações do motor (rpm)** - velocidade de rotação do motor, expressa em número de rotações por minuto;
12. **Raio (metros)** - raio da curva, expresso em metros;
13. **α (graus)** - ângulo longitudinal da estrada, expresso em graus;

14. **FR (N)** - a resistência ao rolamento é a resultante das forças que cada pneu opõe ao deslocamento do veículo, expressa em Newton. Esta resistência representa as perdas de energia do veículo devido às deformações sofridas pela superfície de contacto do pneu com o asfalto e com os obstáculos com que se depara.

15. **FCTP (N)** - força da componente tangencial do peso correspondente à componente tangencial da força gravitacional exercida no veículo, dada pela expressão seguinte:

$$FCTP = massa_{total} * g * sen(\theta) \quad (4.5)$$

em que $massa_{total}$ corresponde à massa total do veículo (massa do veículo + massa dos passageiros) expressa em quilogramas, g corresponde à aceleração da gravidade e θ corresponde ao ângulo longitudinal da estrada, expresso em radianos $\left(\theta = \frac{\alpha\pi}{180}\right)$;

16. **FD (N)** - força de arrasto aerodinâmico que cria resistência ao movimento do veículo durante o seu deslocamento, expressa em Newton;

17. **FTCD (N)** - força da componente tangencial de arrasto correspondente à força exercida nos pneus quando o veículo se encontra a curvar, expressa em Newton;

18. **FP (N)** - força de propulsão responsável pelo deslocamento do veículo, expressa em Newton;

19. **Energia (J)** - energia mecânica total do veículo, expressa em Joule;

20. **Estado do motor** - o estado do motor indica se este se encontra ou não ligado. O valor 1 corresponde ao motor ligado e o valor 0 corresponde ao motor desligado;

21. **Rendimento do sistema de transmissão** - valor do rendimento de transmissão em função da velocidade do veículo;

22. **Binário de forças do motor (N.m)** - o binário de forças do motor corresponde à medida da força que atua sobre a cambota¹ de um motor, medida em Newton por metro, representando a força do motor;

23. $\frac{Sp}{Sp(bsfc_{min})}$ - relação entre a velocidade média do pistão do motor e a velocidade média do pistão no ponto de consumo específico de combustível mínimo de travagem;

24. **Trabalho realizado pelo motor (J)** - o trabalho realizado pelo motor corresponde à medida de energia transferida pela resultante das forças que provocam o deslocamento do veículo, expresso em Joule;

25. **Trabalho das Forças não Conservativas Resistentes (J)** - trabalho realizado pelas forças não conservativas como, por exemplo, atrito e resistência do ar. Esta medida é expressa em Joule;

¹**Cambota ou eixo de manivelas** - responsável por fazer girar vários dispositivos do veículo como, por exemplo, bomba da direção hidráulica, bomba de ar-condicionado, bomba de água, entre outros.

26. **Consumo Específico de Combustível de Travagem (g/kW.h)** - medida da eficiência do combustível do motor. Esta medida é dada pela taxa de consumo de combustível a dividir pela potência produzida, sendo expressa em gramas por kilowatt hora;

27. $\frac{bmep(bsfc_{min})}{bmep_{max}}$ - Relação entre a pressão média efetiva de travagem no ponto de consumo específico de combustível mínimo de travagem e a pressão média máxima efetiva de travagem;

28. $\frac{bmep(bsfc_{min})}{bmep}$ - Relação entre a pressão média efetiva de travagem no ponto de consumo específico de combustível mínimo de travagem e a pressão média efetiva de travagem, calculada da seguinte maneira:

$$\frac{bmep(bsfc_{min})}{bmep} = \begin{cases} \frac{bmep(bsfc_{min})}{bmep} & bmep > 0 \\ default & bmep \leq 0 \end{cases} \quad (4.6)$$

29. $\left(\frac{bmep(bsfc_{min})}{bmep}\right)_{final}$ - Esta variável representa o mesmo que a anterior, porém, de forma a garantir que o valor da pressão média efetiva de travagem é sempre menor que o valor da pressão média efetiva de travagem máxima, ambos em $t = i$, esta é calculada da seguinte maneira:

$$\left(\frac{bmep(bsfc_{min})}{bmep}\right)_{final} = \max\left(\frac{bmep(bsfc_{min})}{bmep_{max}}; \frac{bmep(bsfc_{min})}{bmep}\right) \quad (4.7)$$

30. a , b e c - Características do binário e do consumo do motor;

31. $\frac{bsfc}{bsfc_{min}}$ - relação entre consumo específico de combustível na travagem e o consumo específico de combustível mínimo na travagem;

32. **Massa de combustível consumido:**

- **quando a velocidade do veículo é inferior à velocidade mínima da primeira mudança:** massa de combustível consumido pelo motor do veículo quando este se encontra em ralenti, para um deslocamento elementar (um metro). Este valor é expresso em gramas;
- **quando a velocidade do veículo é igual ou superior à velocidade mínima da primeira mudança:** massa de combustível consumido pelo motor do veículo quando este se encontra fora de ralenti, para um deslocamento elementar (um metro). Este valor é expresso em gramas;

33. **Massa de combustível consumido (g)** - Massa total de combustível consumido pelo motor entre o $t = i - 1$ e $t = i$, expressa em gramas;

34. **Massa de combustível consumido acumulado (g)** - Massa de combustível consumido pelo motor desde o início do trajeto até $t = i$, expressa em gramas, dada pela seguinte expressão:

$$mfac = \sum_{j=1}^i mftotal_j \quad (4.8)$$

onde $mftotal_i$ corresponde à massa do combustível consumido em $t = i$;

35. **Distância acumulativa (metros)** - distância percorrida desde o início do trajeto até $t = i$, expressa em metros, dada pela expressão seguinte:

$$dac_i = \sum_{j=1}^i \Delta d_j \quad (4.9)$$

onde Δd_j corresponde à variação do espaço percorrido entre $t = j - 1$ e $t = j$.

Numa fase posterior, foram fornecidos os registos das 8765 viagens e dos 1621 abastecimentos dos 71 veículos pesados, referentes ao mês de Janeiro de 2018.

Os registos de abastecimentos indicam, para cada veículo, a data, a hora e o número de litros abastecidos. Os registos de viagens indicam, para cada viagem, a data e hora de início e fim e um conjunto de valores de 14 variáveis, que representam as condições do veículo e as forças de atuação durante a viagem. Essas variáveis são: o tempo de funcionamento do veículo, a velocidade do veículo, a distância percorrida, a altitude, o raio da curva, a força de resistência ao rolamento, a componente tangencial da força gravitacional, a força de arrasto aerodinâmico, a força da componente tangencial de arrasto, a força de propulsão, a energia mecânica total do veículo, o rendimento do sistema de transmissão, o binário de forças do motor e o número de rotações do motor, tal como se encontram descritas anteriormente. Estas apresentam, em média, 919 entradas recolhidas em intervalos de tempo não constantes, uma distância média percorrida de 33 quilómetros e uma duração média de 2 horas e meia.

4.2 Dados meteorológicos

Com vista a explorar a influência que as condições meteorológicas têm no consumo de combustível de um veículo foi utilizada a *Weather API*² com a qual é possível aceder a um resumo das condições meteorológicas para cada dia do ano corrente, bem como de anos anteriores. Os dados fornecidos por esta API são recolhidos por equipamentos próprios, designados *Weather Stations*, espalhados por todo o mundo. Em Coimbra existem duas *Weather Stations* localizadas em diferentes pontos da cidade (Figura 4.1).

²Para mais informações consultar: www.wunderground.com/weather/api/

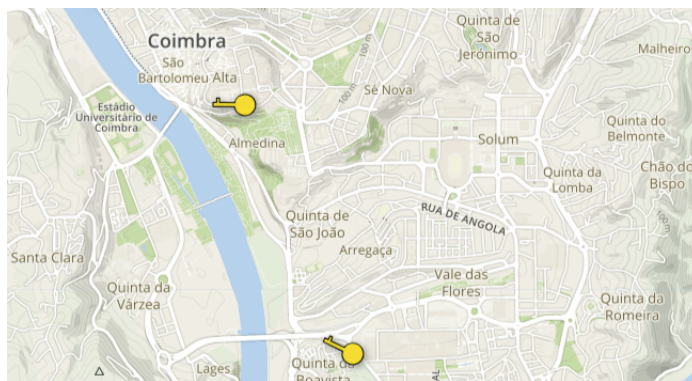


Figura 4.1: Localização das duas *Weather Stations* de Coimbra.

Assim sendo, foram recolhidos os sumários diários das condições meteorológicas registadas durante os mesmos períodos temporais dos dados fornecidos pela empresa. Cada sumário diário é composto pelas seguintes informações:

- **Pluviosidade:** Variável booleana que indica a ocorrência ou não de precipitação pelo menos uma vez durante todo o dia (Figura 4.2);

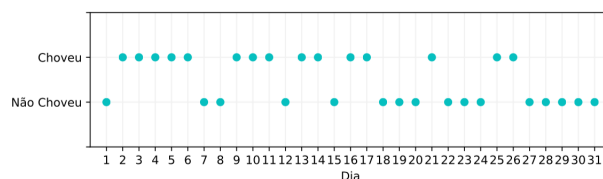


Figura 4.2: Pluviosidade regista em Coimbra de 1 a 31 de Janeiro de 2018.

- **Pluviosidade média (mm):** Quantidade média de água da chuva que, devido à precipitação, ficou acumulada num determinado local durante um determinado período de tempo, expressa em milímetros (Figura 4.3);

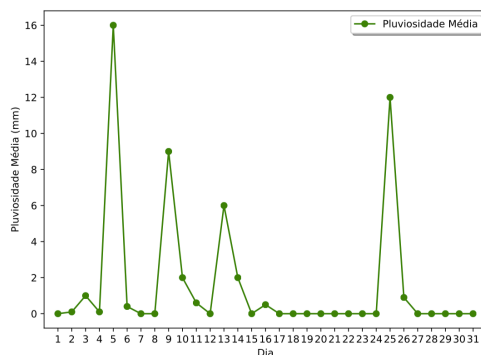


Figura 4.3: Pluviosidade média regista em Coimbra de 1 a 31 de Janeiro de 2018.

- **Temperatura máxima (°C):** Temperatura máxima atingida durante o dia, expressa em °Celsius (Figura 4.4);
- **Temperatura média (°C):** Temperatura média sentida durante o dia, expressa em °Celsius (Figura 4.4);

- **Temperatura mínima (°C):** Temperatura mínima atingida durante o dia, expressa em °Celsius (Figura 4.4);

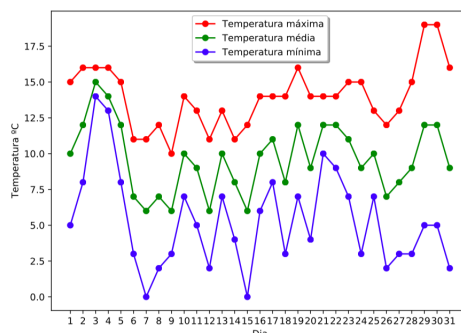


Figura 4.4: Temperatura máxima, média e mínima regista em Coimbra de 1 a 31 de Janeiro de 2018.

- **Pressão atmosférica máxima (mBar):** Pressão atmosférica máxima atingida durante o dia, expressa em milésimos de bar (Figura 4.5);
- **Pressão atmosférica média (mBar):** Pressão atmosférica média sentida durante o dia, expressa em milésimos de bar (Figura 4.5);
- **Pressão atmosférica mínima (mBar):** Pressão atmosférica mínima atingida durante o dia, expressa em milésimos de bar (Figura 4.5);

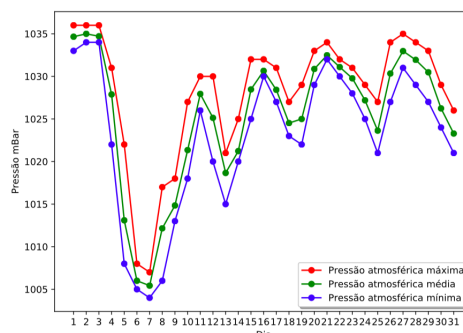


Figura 4.5: Pressão atmosférica máxima, média e mínima regista em Coimbra de 1 a 31 de Janeiro de 2018.

- **Velocidade máxima do vento (kph):** Velocidade máxima do vento atingida durante o dia, expressa em quilómetros por hora (Figura 4.6);
- **Velocidade média do vento (kph):** Velocidade média do vento sentida durante o dia, expressa em quilómetros por hora (Figura 4.6);
- **Velocidade mínima do vento (kph):** Velocidade mínima do vento atingida durante o dia, expressa em quilómetros por hora (Figura 4.6);

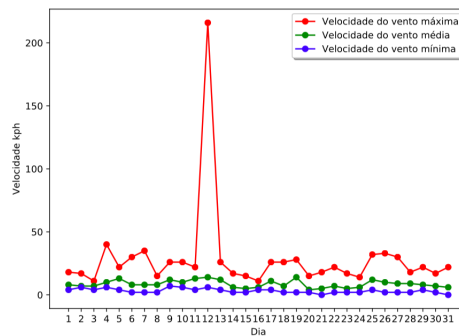


Figura 4.6: Velocidade máxima, média e mínima do vento regista em Coimbra de 1 a 31 de Janeiro de 2018.

- **Direção média do vento (°):** Direção média do vento durante o dia, onde 0° corresponde à direção Norte aumentando no sentido dos ponteiros do relógio, expressa em graus (Figura 4.7).

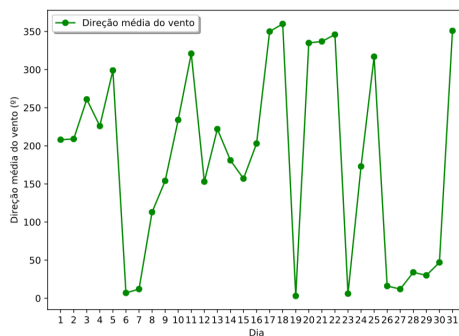


Figura 4.7: Direção média do vento regista em Coimbra de 1 a 31 de Janeiro de 2018.

4.3 Criação dos conjuntos de dados

Com os dados enunciados anteriormente foi possível criar quatro conjuntos com os quais foram realizadas as experiências apresentadas nos capítulos 5, 6 e 7. Os conjuntos de dados são os seguintes:

- **1º Conjunto de dados:** É composto pelos valores das 37 variáveis (secção 4.1), pelos valores das variáveis meteorológicas (secção 4.2) e pelo valor do consumo de combustível acumulativo, em litros por 100 quilómetros, estimado pelo modelo da empresa. Estes registos são referentes a uma única viagem de um veículo realizada no dia 28 de Janeiro de 2017;
- **2º Conjunto de dados:** É composto pelos valores das 14 variáveis (referidas no final da secção 4.1), pelos valores das variáveis meteorológicas (secção 4.2) e pelo número real de litros gastos. Esses registos são referentes à última recolha de cada viagem, realizadas durante o mês de Janeiro de 2018. Neste conjunto foram considerados os períodos de abastecimento que contemplam apenas uma viagem, perfazendo

um total de 190 períodos. Desta forma, o número de litros abastecidos corresponde ao número de litros gastos durante a única viagem realizada;

- **3º Conjunto de dados:** É um subconjunto do anterior, constituído apenas por 45 entradas, as únicas que não apresentam valores em falta para as 14 variáveis;
- **4º Conjunto de dados:** No quarto conjunto de dados foram considerados apenas os períodos de abastecimento que contemplam duas viagens, perfazendo um total de 28 períodos. Desta forma, o número de litros abastecidos corresponde à soma do número de litros gastos em cada viagem. Assim sendo, cada registo deste conjunto é composto pelos valores das 14 variáveis (referidas no final da secção 4.1) e das condições meteorológicas (secção 4.2) para as duas viagens e o número de litros total gastos.

4.4 Seleção de variáveis

Uma das vantagens da técnica de PG é que esta utiliza na geração dos modelos apenas as variáveis que se demonstrem relevantes na modelação da solução para um dado problema. No entanto, quando o conjunto de entrada é muito grande a deteção dessas variáveis torna-se difícil, bem como a eliminação do ruído.

O objetivo desta seleção é eliminar o ruído, bem como simplificar o modelo matemático, reduzindo, tanto quanto possível, o número de variáveis envolvidas na geração do modelo. Além disso, permite reduzir o poder computacional exigido.

4.4.1 Análise correlacional

Para cada par de variáveis foi calculado o Coeficiente de Correlação de *Pearson*. Este coeficiente representa o grau de relação linear entre duas variáveis quantitativas e é dado pela Equação 4.10.

$$Correl(x, y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (4.10)$$

O seu valor varia entre -1 e 1 , onde 0 indica que não existe correlação linear, 1 indica que existe uma correlação linear positiva e -1 indica que existe uma correlação linear negativa. Quanto mais próximo o valor do coeficiente estiver de -1 ou de 1 , mais forte é a relação entre as duas variáveis (Figura 4.8).

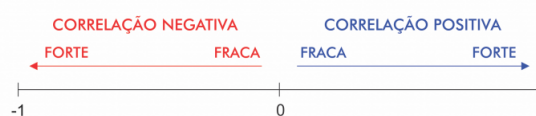


Figura 4.8: Representação do intervalo de valores da Correlação de *Pearson* [5].

Com o cálculo dos Coeficientes de Correlação de *Pearson* foram encontrados quatro pares de variáveis que têm entre si uma correlação linear positiva (com valor exatamente igual a 1). Esses pares são:

- Altitude e altura;
- Número de rotações do motor e $\frac{Sp}{Sp(bsfc_{min})}$;
- Distância e distância acumulativa;
- $\frac{bsfc}{bsfc_{min}}$ e consumo específico de combustível de travagem.

Assim sendo, as variáveis altura, $\frac{Sp}{Sp(bsfc_{min})}$, distância acumulativa e consumo específico de combustível de travagem podem ser excluídas do conjunto de variáveis de entrada, não sendo consideradas pelo algoritmo de PG.

Todos os resultados da análise correlacional encontram-se no Apêndice A.

4.4.2 Seleção de variáveis pelos técnicos

Da primeira reunião com os técnicos da empresa foi possível concluir que a maioria das variáveis fornecidas são referentes a características da constituição do motor. Essas variáveis, por questões negociais, não são do conhecimento público e a sua obtenção junto das empresas de fabrico automóvel envolve grandes custos. Além disso, o seu valor é baseado num veículo composto apenas por peças de origem e em correto funcionamento, algo que dificilmente se verifica.

Ainda que essas variáveis possam ser determinadas com base em modelos aproximados, a solução ideal seria que o modelo de previsão do consumo de combustível não fosse dependente de nenhuma delas, passando a incluir somente variáveis referentes ao funcionamento do veículo, as quais são facilmente obtidas com os sensores veiculares.

As variáveis referentes ao funcionamento do veículo são: tempo de funcionamento, distância percorrida, altitude, velocidade média, raio da curva, energia, rendimento de transmissão e o número de rotações do motor.

Outras variáveis que podem ainda ser incluídas na criação do modelo são as forças, uma vez que não representam qualquer despesa para a empresa. Essas forças são as seguintes: força de resistência ao rolamento, força da componente tangencial do peso, força de arrasto, força da componente tangencial de arrasto e a força de propulsão.

Assim sendo, existem dois conjuntos de variáveis a serem considerados, um constituído apenas pelas variáveis de funcionamento do veículo e outro constituído pelas variáveis de funcionamento e as forças de atuação no veículo.

4.5 Arquitetura das experiências

Com os diferentes conjuntos de dados apresentados foram realizadas as experiências descritas nos capítulos 5, 6 e 7. Essas experiências foram realizadas segundo duas arquiteturas.

Na primeira arquitetura (Figura 4.9) o algoritmo de PG é aplicado a um conjunto de dados, o qual devolve o melhor modelo gerado.

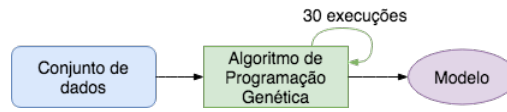


Figura 4.9: Arquitetura 1 das experiências realizadas com os conjuntos de dados do estudo.

Na segunda arquitetura (Figura 4.10) o algoritmo de PG é aplicado três vezes, uma vez por cada divisão treino/teste. Por cada uma dessas execuções é devolvido o melhor modelo gerado com base no conjunto de dados de treino.

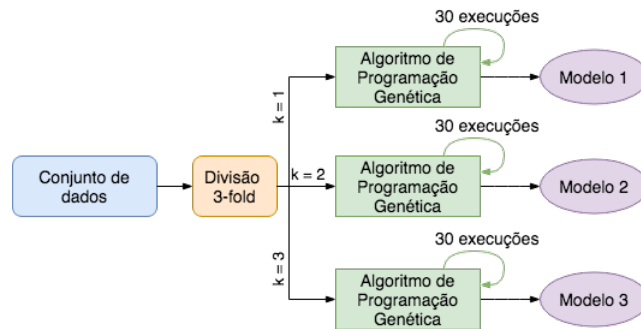


Figura 4.10: Arquitetura 2 das experiências realizadas com os conjuntos de dados do estudo.

Esta página foi propositadamente deixada em branco.

Capítulo 5

Geração de modelo baseado no modelo da empresa

Neste capítulo são apresentadas as experiências realizadas com vista a gerar um modelo semelhante ao fornecido pela empresa, mas variando o número de variáveis utilizadas. A abordagem seguida encontra-se dividida em três fases, como mostra o esquema da Figura 5.1:

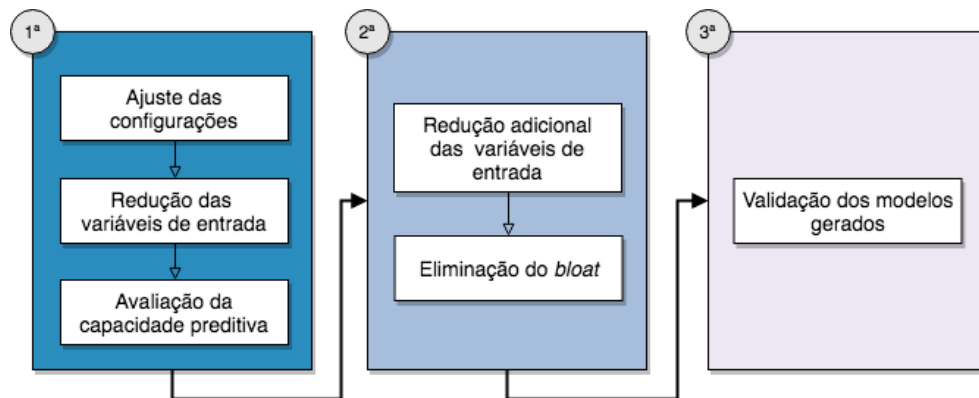


Figura 5.1: Arquitetura da abordagem da geração de um modelo baseado no modelo fornecido.

- **1ª Fase:** Esta fase reflete os ajustes necessários dos parâmetros do algoritmo de PG ao contexto do problema e a geração dos primeiros modelos semelhantes ao fornecido pela empresa, mas dependentes de um número inferior de variáveis. Esta primeira fase encontra-se dividida nas seguintes etapas:
 1. **Ajuste das configurações:** Em primeiro lugar, foram ajustados os valores do número de gerações e do tamanho da população ao contexto do problema. Os valores mais adequados são mantidos para as restantes execuções;
 2. **Redução do número de variáveis de entrada:** Nesta etapa foram utilizadas diferentes abordagens para reduzir o número de variáveis pertencentes ao conjunto de símbolos terminais;

3. **Avaliação da capacidade preditiva:** Por fim, foi avaliada a capacidade de previsão de modelos gerados com os conjuntos de configurações e variáveis que, nas etapas anteriores, apresentaram um erro face ao modelo fornecido inferior ou igual a 2%. Para tal, foi implementado o método de validação cruzada k-fold, calculando o erro absoluto médio entre os valores estimados pelo modelo e os valores reais de consumo, para o conjunto de casos de teste. A avaliação da capacidade preditiva foi mantida para as restantes execuções.
- **2ª Fase:** Esta fase foi realizada depois que foi possível reunir com os técnicos de onde surgiram dois novos conjuntos de variáveis de entrada (secção 4.4.2). Esta fase encontra-se dividida em duas etapas:
 1. **Redução adicional das variáveis de entrada:** Nesta etapa foram gerados novos modelos com os dois conjuntos de variáveis de entrada escolhidos pelos técnicos;
 2. **Eliminação do *bloat*:** Posteriormente foram adicionados diferentes limites dinâmicos ao algoritmo de PG, na tentativa de eliminar o *bloat* e reduzir o tamanho das soluções encontradas.
 - **3ª Fase - Validação dos modelos gerados:** Na última fase foi feita uma validação dos modelos gerados, aplicando-os a um conjunto de dados diferente daqueles que foram utilizados para treino e teste, tentando perceber se estes modelos são generalizáveis.

Na 1ª e 2ª fase foi utilizado o 1º conjunto de dados e na 3ª fase foi utilizado o 3º (previamente descritos na secção 4.3).

5.1 Considerações gerais

Em problemas de regressão a métrica mais utilizada para avaliar uma solução é o Erro Quadrático Médio (EQM). Desta forma, nas experiências realizadas o valor de fitness de cada indivíduo é dado pelo EQM entre o modelo gerado com PG e o modelo da empresa (Equação 5.1).

$$EQM = \frac{\sum_{n=1}^n (x_{pgi} - x_{modelo_i})^2}{n} \quad (5.1)$$

onde x_{pgi} e x_{modelo_i} correspondem aos valores de saída para o registo i dado pelo modelo gerado com PG e pelo modelo da empresa, respetivamente. Quanto menor for esse valor, melhor é o indivíduo.

Relativamente às experiências efetuadas ainda que existam configurações que vão variando ao longo das experiências, existem outras que são transversais a todas elas. Essas configurações encontram-se na Tabela 5.1.

Parâmetros	Valores
Número de execuções	30
Tamanho mínimo das árvores da população inicial	4
Tamanho máximo das árvores da população inicial	6
Tamanho do torneio	3
Tamanho da elite	1
Probabilidade de cruzamento	90%
Probabilidade de mutação	10%

Tabela 5.1: Configurações do algoritmo de Programação Genética transversais a todos os conjuntos de experiências apresentados no capítulo 5.

O algoritmo de PG é executado 30 vezes para cada experiência, em que cada execução gera uma possível solução. No final, dos 30 modelos gerados, aquele que apresentar o melhor valor de fitness é devolvido como solução da experiência.

Na execução do algoritmo, a geração da população inicial é feita de forma aleatória com um dos três métodos de inicialização considerados, em que a profundidade das árvores varia entre 4 e 6. Depois disso, é atribuído um valor de fitness a todos os indivíduos, quantificando a capacidade que cada um deles tem para resolver o problema.

Além disso, é utilizada uma probabilidade de cruzamento de 90% e uma probabilidade de mutação de 10%. A seleção é feita por torneio de tamanho 3, em que três indivíduos são selecionados de forma aleatória da população e o melhor deles é sujeito a operações genéticas, e um elitismo de valor 1, em que um indivíduo é copiado de uma geração para a seguinte sem que sofra qualquer operação genética.

O conjunto de funções utilizado é igual para todas as experiências, sendo composto pelas operações de adição (+), subtração (-), multiplicação (*), divisão protegida (/) e cosseno (cos). Por sua vez, o conjunto de símbolos terminais é composto por Constantes Aleatórias Efêmeras (CAE), que podem tomar os valores -1, 0 ou 1, e por um conjunto de variáveis de entrada que varia para cada experiência.

5.2 1ª Fase

Nesta secção são apresentados os resultados da primeira fase de experiências e as conclusões retiradas.

5.2.1 Ajuste das configurações

Ainda que a PG seja na prática uma técnica robusta, onde vários conjuntos de configurações possam ser aceitáveis para realizar um número suficiente de comparações [2], é necessário ajustar os valores de alguns parâmetros consoante o contexto do problema.

Esses ajustes estão maioritariamente associados ao número de gerações e ao tamanho populacional. Os dois primeiros conjuntos de experiências correspondem ao ajuste desses mesmos parâmetros.

5.2.1.1 Configurações e variáveis de entrada

No primeiro conjunto de experiências foi evoluída uma população de 500 indivíduos, durante 500 gerações. Depois disso, ambos os valores foram ajustados até ser possível gerar uma solução ótima ou próxima dela.

Nas experiências realizadas foram utilizadas todas as 37 variáveis como conjunto de variáveis de entrada, tal como estão descritas na secção 4.1.

Cada conjunto de configurações foi executado para os três métodos de inicialização da população (**Full**, **Grow** e **Ramped Half-and-half** descritos na secção 2.3).

5.2.1.2 Resultados

Na Figura 5.2 é possível verificar que, para o primeiro conjunto de configurações, o valor de fitness médio dos melhores indivíduos nas últimas gerações ainda não está totalmente estabilizado. No entanto, quando foi utilizado o dobro do número de indivíduos e gerações, a partir da geração 900 já não existem grandes melhorias (durante as últimas 100 gerações o valor de fitness não aumenta uma unidade). Deste modo, não é necessário testar mais nenhum conjunto de configurações, terminando assim a fase de ajuste.

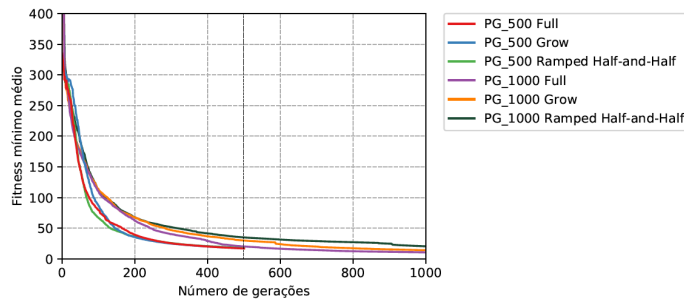


Figura 5.2: Fitness mínimo médio dos conjuntos de configurações *PG_500* e *PG_1000*, para os três métodos de inicialização da população, durante 500 e 1000 gerações, respetivamente.

Na Tabela 5.2 encontra-se o valor de fitness, o número de nós e o erro face ao modelo fornecido para as diferentes soluções geradas. Aquelas que apresentam valores de fitness mais baixos são também as que apresentam o número de gerações e o tamanho populacional maiores. Isto verifica-se para todos os métodos de inicialização. No entanto, em alguns casos, a percentagem de erro em relação ao modelo é um pouco maior. Essa percentagem é dada pela seguinte expressão:

$$Erro_Modelo(\%) = \frac{|consumo_{PG} - consumo_{MF}|}{consumo_{MF}} * 100 \quad (5.2)$$

onde $consumo_{PG}$ e $consumo_{MF}$ representam, respetivamente, o consumo dado pelo modelo gerado com PG e o consumo dado pelo modelo da empresa, para a totalidade da viagem (último registo).

Conjunto de experiências	Inicialização	Fitness mínimo	Nº de nós da solução	Erro modelo (%)
<i>PG_500</i>	Full	2.9830	845	0.82%
	Grow	5.0801	391	6.49%
	R. Half-and-Half	3.0639	362	0.20%
<i>PG_1000</i>	Full	1.0854	665	1.65%
	Grow	3.3493	606	4.79%
	R. Half-and-Half	2.7424	586	2.74%

Tabela 5.2: Resultados dos conjuntos de experiências *PG_500* e *PG_1000* realizados durante a etapa de ajuste de configurações.

O aumento dessa percentagem de erro ocorre porque quando existe um maior número de comparações, os modelos gerados mapeiam de forma mais exata o conjunto de dados de entrada. No entanto, 85% delas correspondem a momentos em que o veículo se encontra em movimento (número de rotações em média de 1242 rotações por minuto e velocidade média a rondar os 7.09 metros por segundo) e o último registo corresponde a um momento em que o veículo está quase a parar (com 652 rotações por minuto e com uma velocidade média de 1.49 metros por segundo). Desta forma, é possível afirmar que os melhores modelos acabam por mapear de forma mais exata os 85% dos dados referentes a momentos do movimento do veículo, mas apresentam-se não tão exatos para momentos em que o veículo se encontra parado ou quase parado.

5.2.1.3 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- Tal como referido por Koza, não é necessário despender muito tempo no ajuste das configurações do algoritmo de PG para que este funcione adequadamente [2];
- Evoluir uma população de 1000 indivíduos durante 1000 gerações é suficiente para a geração de uma solução com taxas de erro razoáveis e aceitáveis para a empresa;
- Os melhores modelos apresentam-se mais exatos ao longo do funcionamento do veículo, mas menos exatos para a globalidade da viagem dada a desproporcionalidade do conjunto de dados, entre momentos de deslocamento do veículo e momentos em que este se encontra parado.

5.2.2 Redução das variáveis de entrada

Nos trabalhos relacionados foi possível verificar que existem diferentes variáveis que quando devidamente conjugadas dão origem a modelos de consumo semelhantes. Desta forma, nesta etapa foram aplicadas diferentes abordagens para reduzir o número de variáveis de

entrada, na tentativa de gerar modelos semelhantes ao fornecido e para os quais a sua geração implique um poder computacional inferior.

5.2.2.1 Configurações e variáveis de entrada

Utilizando as configurações que se mostraram mais adequadas ao contexto do problema na etapa anterior, foram testadas três abordagens para tentar reduzir o conjunto de variáveis de entrada. Essas abordagens são as seguintes:

- **Variáveis não utilizadas (*PG_1000_NU*):** Ao conjunto total das 37 variáveis foram retiradas aquelas que no primeiro conjunto de experiências (*PG_500*) não foram continuamente incluídas nos modelos gerados. Essas variáveis são: relação de transmissão, alfa, FTCD, Energia, $\left(\frac{bmep(bsfc_{min})}{bmep}\right)_{final}$, massa de combustível consumido (quando a velocidade do veículo é inferior à velocidade mínima da primeira mudança) e massa de combustível consumido (quando a velocidade do veículo é igual ou superior à velocidade mínima da primeira mudança);
- **Análise correlacional (*PG_1000_CORR*):** Ao conjunto total das 37 variáveis foram retiradas quatro através da análise correlacional (secção 4.4.1);
- **Variáveis não utilizadas e análise correlacional (*PG_1000_NU_CORR*):** Ao conjunto total das 37 variáveis foram excluídas aquelas que no primeiro conjunto de experiências (*PG_500*) não foram continuamente incluídas nos modelos gerados e quatro variáveis através da análise correlacional.

Cada abordagem foi executada para os três métodos de inicialização da população (Full, Grow e Ramped Half-and-half descritos na secção 2.3).

5.2.2.2 Resultados

Cada conjunto de experiências tem um número diferente de variáveis de entrada. Em *PG_1000_NU* foram consideradas 30 variáveis de entrada, em *PG_1000_CORR* foram consideradas 33 e em *PG_1000_NU_CORR* foram consideradas 26.

Na Tabela 5.3 estão ilustrados os resultados obtidos nas experiências realizadas durante esta etapa. Os resultados mostram que a abordagem que apresentou soluções com um valor de fitness menor e com uma percentagem de erro mais baixa em relação ao modelo fornecido foi a que eliminou variáveis que não foram continuamente incluídas nos modelos gerados durante o primeiro conjunto de experiências (*PG_500*). Isto deve-se ao facto do algoritmo de PG, ao tentar encontrar as melhores soluções o mais rapidamente possível, dar prioridade às variáveis que demonstrem ter uma maior influência na modelação da solução do problema. Assim sendo, quando o número de comparações é insuficiente, o que se verificou nas experiências *PG_500*, as variáveis que não são tão relevantes acabam por não chegar a ser incluídas nos modelos gerados.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro modelo (%)	Nº variáveis da solução	Nº de nós da solução
<i>PG_1000_NU</i>	Full	2.9119	0.76%	31	727
	Grow	1.9502	0.39%	30	961
	R. Half-and-Half	2.3039	2.38%	25	726
<i>PG_1000_CORR</i>	Full	3.9428	1.28%	29	663
	Grow	5.8174	14.99%	30	1002
	R. Half-and-Half	2.2791	4.20%	22	525
<i>PG_1000_NU_CORR</i>	Full	3.3143	4.97%	25	677
	Grow	2.2875	2.50%	24	943
	R. Half-and-Half	4.5679	2.48%	21	420

Tabela 5.3: Resultados dos conjuntos de experiências *PG_1000_NU*, *PG_1000_CORR* e *PG_1000_NU_CORR* realizados durante a etapa de redução das variáveis de entrada.

Por outro lado, o método de inicialização que apresenta a melhor relação entre o valor de fitness mínimo e o número de variáveis utilizadas é o **Ramped Half-and-Half**, excluindo entre 5 a 11 variáveis, para além das que já tinham sido eliminadas dos conjuntos de variáveis de entrada, e apresentando soluções compostas por um número de nós inferiores às restantes. Porém, os modelos gerados com este método de inicialização apresentam um erro não inferior a 2% face modelo fornecido o que, em comparação com as restantes soluções, não representa o valor mais baixo atingido.

A eliminação prévia de variáveis com uma análise correlacional não se demonstrou uma abordagem adequada, apresentado percentagens de erro e valores de fitness superiores aos do conjunto de experiências no qual foi utilizada a totalidade das variáveis (*PG_1000*). Isto demonstra que essas variáveis são relevantes na geração do modelo de previsão de combustível e que a sua eliminação influencia os resultados obtidos.

5.2.2.3 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- A técnica de PG é adequada ao contexto do problema;
- Foi possível gerar vários modelos semelhantes ao fornecido dependentes de um número inferior de variáveis, algumas eliminadas do conjunto de variáveis de entrada pelas diferentes abordagens utilizadas e outras excluídas automaticamente durante a execução do algoritmo;
- A utilização de técnicas complementares para reduzir o conjunto de variáveis de entrada pode ser uma mais valia, forçando os modelos a serem constituídos por um conjunto de variáveis mais restrito e necessitando de um poder computacional menor;
- As variáveis eliminadas por não serem continuamente utilizadas quando o número de avaliações foi insuficiente, correspondem às mesmas variáveis que se demonstraram menos relevantes na geração dos modelos, tendo esta informação sido validada com os técnicos da empresa.

5.2.3 Teste da capacidade preditiva

Para testar a capacidade preditiva dos modelos gerados com PG foi implementada uma validação cruzada. Esta técnica tem como objetivo estimar o quão preciso é um modelo, avaliando o seu comportamento perante um novo conjunto de dados. O conceito fundamental desta técnica é a divisão de um conjunto de dados em k subconjuntos mutuamente exclusivos, onde $k-1$ são utilizados na geração do modelo e o restante é utilizado na validação do mesmo.

O método de validação cruzada utilizado foi o k -fold com valor de k igual a 3, uma vez que este valor permite uma proporção próxima dos 70-30% para a divisão treino/teste, habitualmente utilizada em problemas de ML. Assim sendo, este método consiste na divisão da totalidade dos dados em 3 subconjuntos do mesmo tamanho, como mostra a Figura 5.3, onde 2 subconjuntos são utilizados para treino e 1 é utilizado para teste.

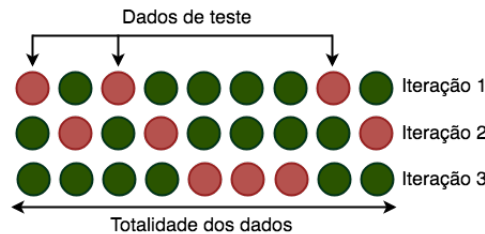


Figura 5.3: Exemplo de uma divisão 3-fold aplicada a um conjunto de dados.

Este processo é realizado 3 vezes, uma por cada divisão treino/teste. Por cada execução é gerado um novo modelo, avaliando a sua capacidade preditiva, como mostra a Figura 5.4.

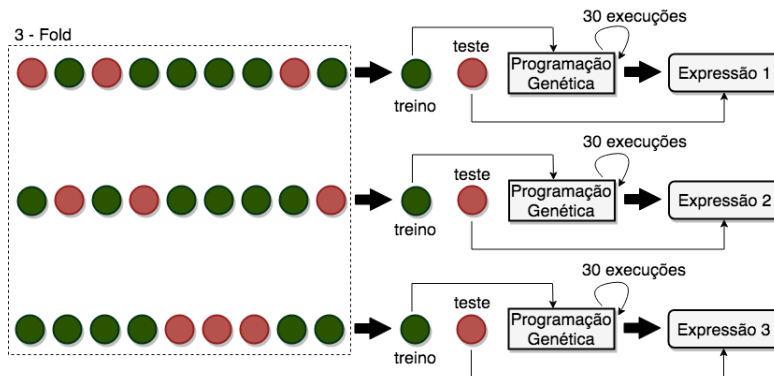


Figura 5.4: Arquitetura da utilização de uma divisão 3-fold no algoritmo de Programação Genética.

5.2.3.1 Configurações e variáveis de entrada

As experiências que nas etapas anteriores apresentaram um erro inferior ou igual a 2% em relação ao modelo da empresa foram repetidas, desta vez com uma divisão 3-fold. Essas experiências são: *PG_500* com o método de inicialização *Full*, *PG_500* com o método de inicialização *Ramped Half-and-half*, *PG_1000_NU* com o método de inicialização *Full*,

PG_1000_NU com o método de inicialização *Grow*, *PG_1000* com método de inicialização *Full* e *PG_1000_CORR* com o método de inicialização *Full*.

5.2.3.2 Resultados

Os resultados do conjunto de experiências realizadas com uma divisão 3-fold encontram-se na Tabela 5.4.

Conjunto de experiências	Experiência	Fitness mínimo	Erro modelo (%)	Erro previsão (L/100km)	Nº de variáveis da solução
<i>PG_1000_E2</i>	<i>PG_500 (Full)</i> 1	2.1767	4.43%	±1.2072	8
	<i>PG_500 (Full)</i> 2	2.0097	1.96%	±2.0097	13
	<i>PG_500 (Full)</i> 3	4.3601	6.44%	±1.6604	17
	<i>PG_500 (Half)</i> 1	6.0703	2.48%	±2.1745	17
	<i>PG_500 (Half)</i> 2	2.8806	2.74%	±1.3181	14
	<i>PG_500 (Half)</i> 3	3.8110	2.17%	±1.8518	15
	<i>PG_1000_NU (Full)</i> 1	1.5703	2.90%	±1.6160	31
	<i>PG_1000_NU (Full)</i> 2	2.9306	2.29%	±1.5675	31
	<i>PG_1000_NU (Full)</i> 3	1.9353	1.39%	±1.4104	30
	<i>PG_1000_NU (Grow)</i> 1	0.9815	2.04%	±1.1438	24
	<i>PG_1000_NU (Grow)</i> 2	2.1568	1.49%	±1.5186	23
	<i>PG_1000_NU (Grow)</i> 3	1.1357	2.60%	±0.9070	19
	<i>PG_1000 (Full)</i> 1	1.8346	2.60%	±1.6027	35
	<i>PG_1000 (Full)</i> 2	1.2788	1.75%	±1.2578	32
	<i>PG_1000 (Full)</i> 3	1.2817	5.99%	±1.2817	34
	<i>PG_1000_CORR (Full)</i> 1	6.5096	0.26%	±2.2420	32
	<i>PG_1000_CORR (Full)</i> 2	3.7811	6.57%	±1.6478	30
	<i>PG_1000_CORR (Full)</i> 3	1.7347	0.43%	±1.7347	33

Tabela 5.4: Resultados do conjunto de experiências *PG_1000_E2* realizado durante a etapa de teste da capacidade preditiva.

Todos os modelos gerados apresentam erros de previsão baixos. Esses erros variam entre 1 e 2 litros por 100 quilômetros, o que numa viagem de 100 quilômetros corresponde a aproximadamente 1% da capacidade total do tanque de combustível do veículo do qual os dados utilizados foram recolhidos.

Nas experiências em que o tamanho populacional e o número de gerações foi 500, as soluções encontradas apresentam valores de fitness e percentagens de erro em relação ao modelo da empresa, em média, superiores. No entanto, são aquelas que são constituídas por um menor número de variáveis, provando que quanto mais variáveis forem utilizadas na geração dos modelos menor será o erro.

Nas restantes experiências, os melhores resultados foram obtidos quando as variáveis que não foram continuamente incluídas nos modelos gerados no conjunto *PG_500*, foram eliminadas do conjunto de símbolos terminais. Além disso, é de salientar que, dentro do conjunto de experiências com as mesmas configurações, são aquelas que apresentam soluções dependentes de um número mais baixo de variáveis.

Tal como foi referido em etapas anteriores, excluir variáveis através de uma análise correlacional (*PG_1000_CORR*) leva a resultados piores do que quando é utilizada a sua totalidade (*PG_1000*), apresentado erros de previsão superiores em 0.5 a 1 litro por 100 quilômetros. Ainda assim, todos os modelos gerados dependem de um número inferior de

variáveis face ao modelo fornecido e apresentam baixos valores de erro nas previsões realizadas (quase sempre inferiores a 2 litros por 100 quilómetros). Contudo, é de salientar o modelo gerado pela experiência *PG_1000_NU Grow 1* devido ao seu baixo valor de fitness, utilizando apenas 50% das variáveis. Esse modelo encontra-se no Apêndice C.

5.2.3.3 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- Os modelos gerados com PG utilizando uma validação cruzada 3-fold apresentaram erros de previsão correspondentes, no máximo, a 1% da capacidade total do tanque de combustível para uma viagem de 100 quilómetros (aproximadamente 2.2 litros);
- A abordagem que se revelou mais eficaz foi a que excluiu do conjunto de símbolos terminais as variáveis que não foram constantemente incluídas nos modelos gerados pelo primeiro conjunto de configurações (*PG_500*), tendo também apresentado os erros de previsão mais baixos, que rondam 1 litro e meio por cada 100 quilómetros;
- O modelo gerado pela experiência *PG_1000_NU Grow 1* foi o que apresentou um erro de previsão menor e um valor de fitness mínimo entre os valores mais baixos atingidos em todas as experiências, dependente de apenas 19 variáveis ($\approx 50\%$).

5.2.4 Conclusões

Depois de realizadas as experiências das etapas da primeira fase foi possível concluir que, numa fase inicial, para qualquer algoritmo de PG é necessário a existência de uma etapa de ajuste das configurações. No entanto, não é preciso despende demasiado tempo nesta tarefa, dado que existem diferentes conjuntos para os quais o algoritmo funciona adequadamente. No final da primeira etapa, foi possível perceber que evoluir uma população de 1000 indivíduos durante 1000 gerações é suficiente para o contexto do problema. Isto acontece porque quanto maior for o número de avaliações, em princípio, melhor será o desempenho dos modelos gerados.

A utilização de técnicas complementares, que eliminam variáveis do conjunto de símbolos terminais numa fase anterior à execução do algoritmo, podem ser uma mais valia, reduzindo o número de variáveis do modelo e o poder computacional necessário. No final da segunda etapa, foi possível perceber que, eliminar previamente as variáveis que não foram incluídas nos modelos gerados com um conjunto de configurações que realiza um número insuficiente de avaliações, gerou modelos com melhores resultados, ainda que o mesmo não se tenha verificado para as restantes abordagens implementadas. Ainda assim, esses modelos são constituídos por um número de variáveis inferior ao criado manualmente pela empresa.

Por sua vez, a utilização de uma validação cruzada para avaliar a capacidade de generalização dos modelos foi importante, na medida em que permitiu perceber que não ocorreu

overfitting na geração dos modelos, uma vez que quando foram aplicados a um conjunto de dados de teste desconhecido, apresentaram erros de previsão baixos, que variam entre 1 e 2 litros por 100 quilômetros (1% da capacidade do tanque de combustível).

A melhor solução encontrada foi resultante da experiência *PG_1000_NU Grow 1*, salientando a utilização de 19 das 37 variáveis iniciais ($\approx 50\%$).

5.3 2ª Fase

Após o término da 1ª fase foi marcada uma reunião com a empresa onde os resultados foram discutidos. Dessa reunião foi possível concluir que os modelos gerados dependem maioritariamente de variáveis referentes a características do motor do veículo, variáveis essas que representam uma despesa para a empresa por serem difíceis de obter junto dos fabricantes. No entanto, uma vez que com a PG foi possível gerar modelos dependentes de um menor número de fatores, alguns deles eliminados previamente do conjunto de entrada, a empresa estabeleceu dois conjuntos restritos de variáveis que devem ser considerados para a criação de novos modelos. As variáveis que constituem esses conjuntos encontram-se descritas na seção 4.4.2 e são referentes ao funcionamento do veículo e às forças de atuação durante o seu deslocamento.

Nesta seção são apresentados os resultados da segunda fase de experiências e as conclusões retiradas.

5.3.1 Redução adicional das variáveis de entrada

A primeira etapa desta fase tem como objetivo verificar se é possível gerar um modelo, semelhante ao fornecido, utilizando apenas as variáveis que constituem um dos dois conjuntos pedidos pelos técnicos.

5.3.1.1 Configurações e variáveis de entrada

Utilizando as configurações e a validação cruzada estabelecidas na fase anterior foram realizados dois conjuntos de experiências, um por cada conjunto de variáveis, descritas na seção 4.4.2:

- ***PG_1000_VF***: Neste conjunto de experiências foram consideradas apenas as variáveis referentes ao funcionamento do veículo;
- ***PG_1000_VF_F***: Neste conjunto de experiências foram consideradas as variáveis referentes ao funcionamento do veículo e às forças de atuação durante o seu deslocamento.

Cada conjunto de experiências foi executado para os três métodos de inicialização da população (**Full**, **Grow** e **Ramped Half-and-half**).

5.3.1.2 Resultados

Os resultados para os dois conjuntos de experiências encontram-se presentes na Tabela 5.5.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro modelo (%)	Erro previsão (L/100km)	Nº de nós da solução
<i>PG_1000_VF</i>	Full 1	7.3683	1.80%	±2.5479	658
	Full 2	5.7900	7.34%	±2.5368	742
	Full 3	7.1260	14.03%	±2.9357	1012
	Grow 1	6.8770	10.45%	±2.3193	923
	Grow 2	6.7955	1.32%	±2.5673	999
	Grow 3	6.3247	1.17%	±1.8994	682
	R. Half-and-Half 1	6.3028	2.35%	±2.4495	737
	R. Half-and-Half 2	6.8418	7.66%	±2.2915	832
	R. Half-and-Half 3	5.5353	3.42%	±2.1256	727
<i>PG_1000_VF_F</i>	Full 1	1.8860	0.73%	±1.3826	1180
	Full 2	3.1240	2.06%	±1.4415	585
	Full 3	2.2291	0.57%	±1.6978	574
	Grow 1	1.5717	2.78%	±1.4930	604
	Grow 2	2.6730	1.91%	±1.7838	907
	Grow 3	1.7892	2.64%	±1.0577	1018
	R. Half-and-Half 1	2.8523	5.07%	±2.5139	624
	R. Half-and-Half 2	2.1432	0.34%	±1.1529	1070
	R. Half-and-Half 3	2.8810	0.20%	±1.3257	875

Tabela 5.5: Resultados dos conjuntos de experiências *PG_1000_VF* e *PG_1000_VF_F* realizados durante a etapa de redução adicional das variáveis de entrada.

O conjunto composto apenas por variáveis relativas ao funcionamento do veículo demonstrou-se insuficiente, apresentando os piores valores de fitness para as soluções encontradas. Tal como está na Tabela 5.5, onde uma variação de 2 litros para o conjunto de experiências *PG_1000_VF*, ao contrário do que aconteceu nas experiências iniciais (*PG_500*), não aconteceu devido ao tamanho reduzido da população e/ou das gerações, uma vez que os gráficos da Figura 5.5 mostram que o valor de fitness estagnou por volta da geração 800, para todos os métodos de inicialização.

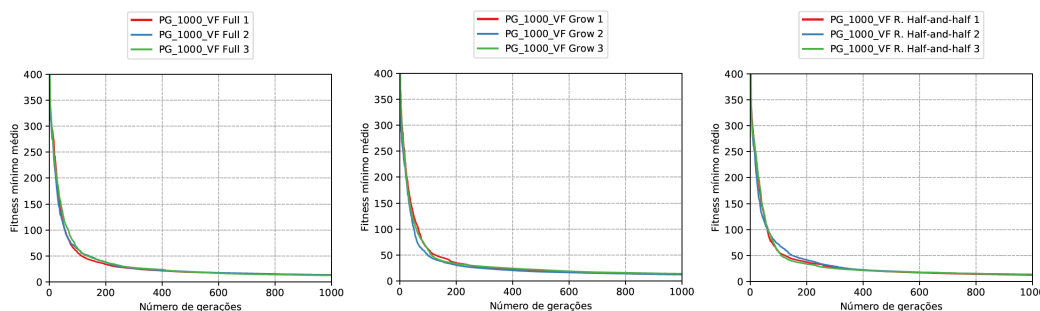


Figura 5.5: Fitness mínimo médio do conjunto de experiências *PG_1000_VF* para os três métodos de inicialização da população, durante 1000 gerações.

Por outro lado, o conjunto que além das variáveis de funcionamento do veículo também inclui as forças demonstrou-se suficiente para atingir resultados muito perto dos atingidos anteriormente, em que o número de variáveis era superior. Nesse conjunto, as experiências realizadas com o método de inicialização *Ramped Half-and-half* foram as que apresentaram globalmente piores resultados, tendo os valores de fitness mínimo e os erros de

previsão mais elevados. Pelo contrário, os restantes métodos de inicialização geraram modelos igualmente bons comparando com o modelo fornecido, com percentagens de erro de previsão admissíveis pela empresa e com um número reduzido de variáveis consideradas. No entanto, as árvores resultantes do algoritmo de PG são compostas por, no mínimo, 600 nós e, como é notório nos gráficos da Figura 5.6, existe um crescimento no tamanho das árvores sem que se verifiquem melhorias substanciais no valor de fitness mínimo médio dos indivíduos (entre as gerações 900 e 1000 o valor de fitness desceu, em média, 0.87).

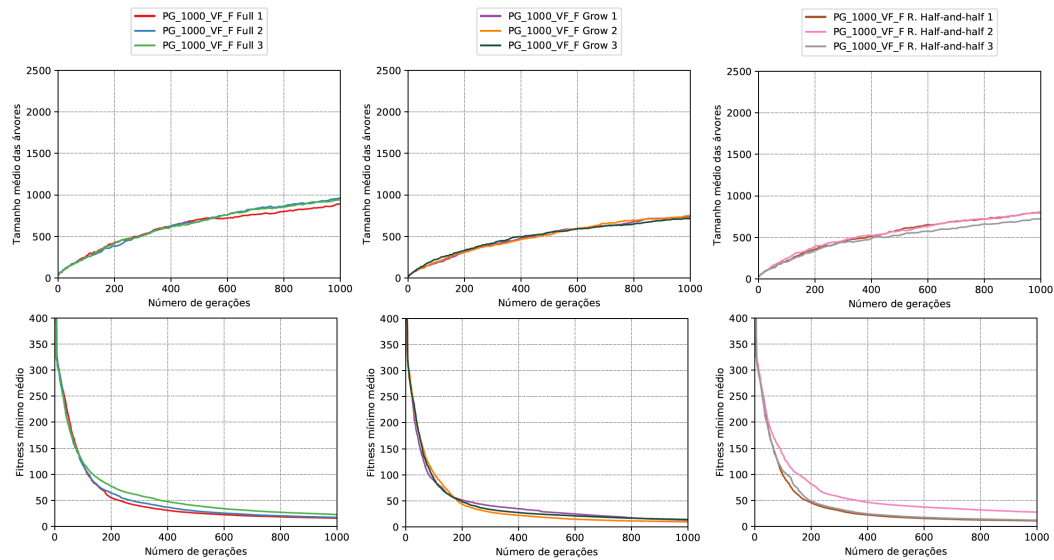


Figura 5.6: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_VF_F* para os três métodos de inicialização da população, durante 1000 gerações.

5.3.1.3 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- Dos dois conjuntos de variáveis estabelecidos pelos técnicos, aquele que é composto pelas variáveis de funcionamento do veículo e pelas forças de atuação durante o seu deslocamento foi o que apresentou melhores resultados, sendo este conjunto utilizado nas restantes experiências;
- O conjunto composto apenas pelas variáveis de funcionamento do veículo apresentou erros de previsão, em média, superiores em 1 litro por cada 100 quilómetros e uma percentagem de erro em relação ao modelo da empresa superior em aproximadamente 3.5% comparativamente ao outro conjunto, por não reunir todos os fatores adequados para gerar modelos capazes de mapear e prever o consumo de combustível do veículo e não por ter um número insuficiente de avaliações;
- Os modelos gerados pelo conjunto de experiências *PG_1000_VF_F* são compostos pelas 14 variáveis seleccionadas pelos técnicos, aproximadamente 38% das variáveis

iniciais, apresentado resultados muito semelhantes ao modelo fornecido e com baixos erros de previsão (em média, um litro e meio por 100 quilômetros);

- O crescimento do tamanho das árvores ao longo das gerações não foi acompanhado por melhorias proporcionais no valor do fitness mínimo, correspondendo a uma das limitações da PG designada *bloat*, existindo mecanismos adequados para tentar atenuar esse problema.

5.3.2 Eliminação do *bloat*

Tal como referido na fase anterior, o tamanho das árvores continua a crescer sem uma aparente melhoria no valor de fitness dos indivíduos. Ainda que a solução mais óbvia seja escolher uma condição de paragem menos exigente para que o algoritmo termine mais cedo, isto não é suficiente para garantir que a solução encontrada é aceitável, dado que não é possível prever a sua complexidade [26].

Assim sendo, nesta etapa foram adicionados diferentes limites dinâmicos ao algoritmo de PG, tal como se encontram descritos em secção 2.10.1, na tentativa de atenuar ou eliminar o *bloat* e reduzir o tamanho das soluções encontradas.

5.3.2.1 Configurações e variáveis de entrada

Nesta etapa foram utilizadas as configurações e a validação cruzada já estabelecidas na 1ª fase e o conjunto de variáveis que na etapa anterior se demonstrou o mais adequado. Os conjuntos de experiências realizados foram os seguintes:

- ***PG_1000_VF_F_LD17***: Neste conjunto de experiências, ao limite estático com valor 17 foi adicionado um limite dinâmico aplicado à profundidade das árvores. Este conjunto foi executado para os três métodos de inicialização da população (Full, Grow e Ramped Half-and-half);
- ***PG_1000_VF_F_LD30***: Neste conjunto de experiências, foi utilizado um limite estático com valor 30 e um limite dinâmico aplicado à profundidade das árvores. Este conjunto foi executado apenas para o método de inicialização da população Grow;
- ***PG_1000_VF_F_LD35***: Nesta experiência, foi utilizado um limite estático com valor 35 e um limite dinâmico aplicado à profundidade das árvores. Esta experiência foi executada para o método de inicialização da população Grow, apenas para uma das três divisões da validação cruzada;
- ***PG_1000_VF_F_LDTA***: Neste conjunto de experiências, foi utilizado um limite estático com valor 17 e um limite dinâmico aplicado ao número de nós das árvores. Este conjunto foi executado para os três métodos de inicialização da população (Full, Grow e Ramped Half-and-half).

O valor 17 para o limite estático corresponde ao limite utilizado por defeito pelo algoritmo de PG, enquanto que os valores 30 e 35 foram escolhidos com base em algumas execuções realizadas com o limite máximo suportado computacionalmente (de valor 90), em que a profundidade máxima das árvores da maioria da população na última geração variou entre esse dois valores.

O conjunto *PG_1000_VF_F_LD17* foi repetido duas vezes, em que numa delas foi aplicada uma truncação ao valor devolvido pelo modelo gerado.

Caso esse valor seja inferior a 0 ou superior a 100 é substituído por 0 ou 100, respetivamente (equação 5.3). Essa truncação é justificada pelos valores normais de consumo dos veículos pesados rondarem os 50 litros a cada 100 quilómetros.

$$\text{consumo}(L/100km) = \begin{cases} 0 & \text{modelo}_{PG} < 0 \\ \text{modelo}_{PG} & 0 \leq \text{modelo}_{PG} \leq 100 \\ 100 & \text{modelo}_{PG} > 100 \end{cases} \quad (5.3)$$

Assim sendo, mesmo tendo sido realizadas mais experiências, nesta secção só são apresentadas aquelas às quais foi aplicada a truncação. Os resultados das restantes experiências encontram-se no Apêndice B.

5.3.2.2 Resultados

Os resultados da aplicação de diferentes limites dinâmicos à profundidade das árvores geradas encontram-se na Tabela 5.6.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro modelo (%)	Erro previsão (L/100km)	Nº de nós da solução
<i>PG_1000_VF_F_LD17</i>	Full 1	2.2327	1.74%	±1.4412	637
	Full 2	2.7629	0.01%	±1.6671	735
	Full 3	2.9800	0.37%	±1.4612	552
	Grow 1	2.5981	3.35%	±2.1546	634
	Grow 2	1.6932	0.46%	±2.1570	963
	Grow 3	2.5120	0.01%	±1.3762	569
	R. Half-and-Half 1	1.0677	1.31%	±1.3254	779
	R. Half-and-Half 2	2.0348	1.68%	±1.7186	461
	R. Half-and-Half 3	2.1966	3.51%	±1.3937	1030
<i>PG_1000_VF_F_LD30</i>	Grow 1	1.2373	1.34%	±1.8447	1517
	Grow 2	1.9775	4.70%	±1.7513	2362
	Grow 3	2.0914	1.97%	±1.3399	1150
<i>PG_1000_VF_F_LD35</i>	Grow 1	1.9131	0.46%	±1.4404	1983

Tabela 5.6: Resultados dos conjuntos de experiências *PG_1000_VF_F_LD17*, *PG_1000_VF_F_LD30* e *PG_1000_VF_F_LD35* realizados durante a etapa de eliminação do *bloat*.

Comparando o primeiro conjunto de experiências (*PG_1000_VF_F_LD17*) com os resultados da etapa anterior é notório que o número de nós que compõem as soluções encontradas é globalmente inferior. Além disso, os valores de fitness das soluções e a percentagem de erro em relação ao modelo fornecido são mais baixos. No entanto, nos gráficos da Figura 5.7 é possível verificar que, mesmo quando é adicionado um limite dinâmico aplicado à profundidade das árvores, continua a existir *bloat*, onde o valor de fitness se encontra quase estagnado, em que as melhorias nas últimas 100 gerações são, em média, de 0.87.

Ainda assim, dada a complexidade do problema em análise, para o qual a profundidade 17

podia ser insuficiente para gerar uma solução ótima, foram realizados mais dois conjuntos de experiências em que o valor do limite estático foi aumentando para 30 e 35. Esses conjuntos de experiências só foram repetidos para o método de inicialização *Grow* pois é aquele que, tal como mostram os gráficos centrais da Figura 5.7, gera soluções com um tamanho e um valor de fitness, em média, mais baixos.

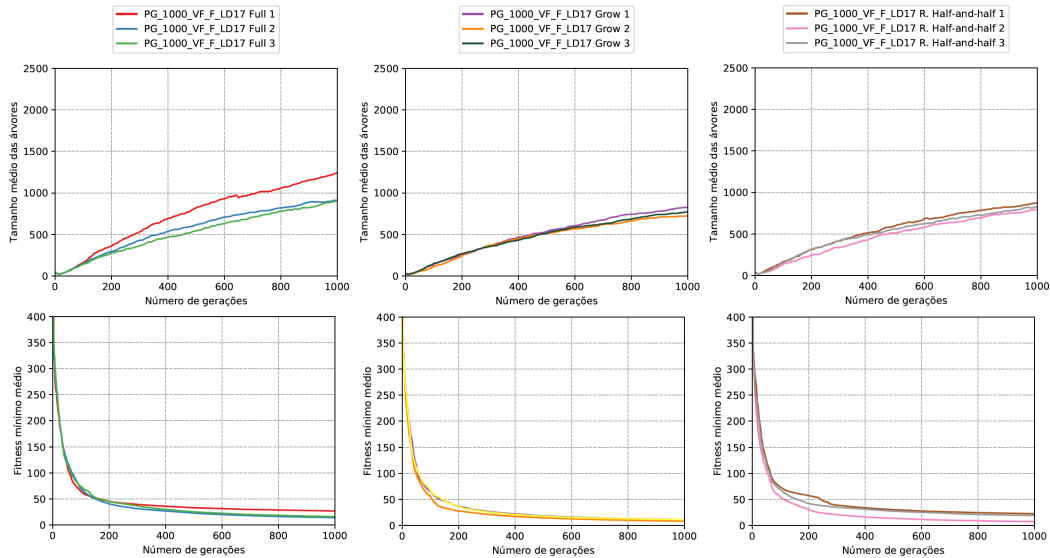


Figura 5.7: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_VF_F_LD17* para os três métodos de inicialização da população, durante 1000 gerações.

Com base nos gráficos da Figura 5.8 é notório que quanto maior é o valor do limite estático, maior é o número de nós que compõem as soluções encontradas. Assim sendo, essas soluções acabam por ser tão ou mais complexas do que quando é utilizado apenas um limite estático com valor 17 ou quando este é conjugado com um limite dinâmico aplicado à profundidade das árvores. Além disso, apresentam o dobro do tempo de execução.

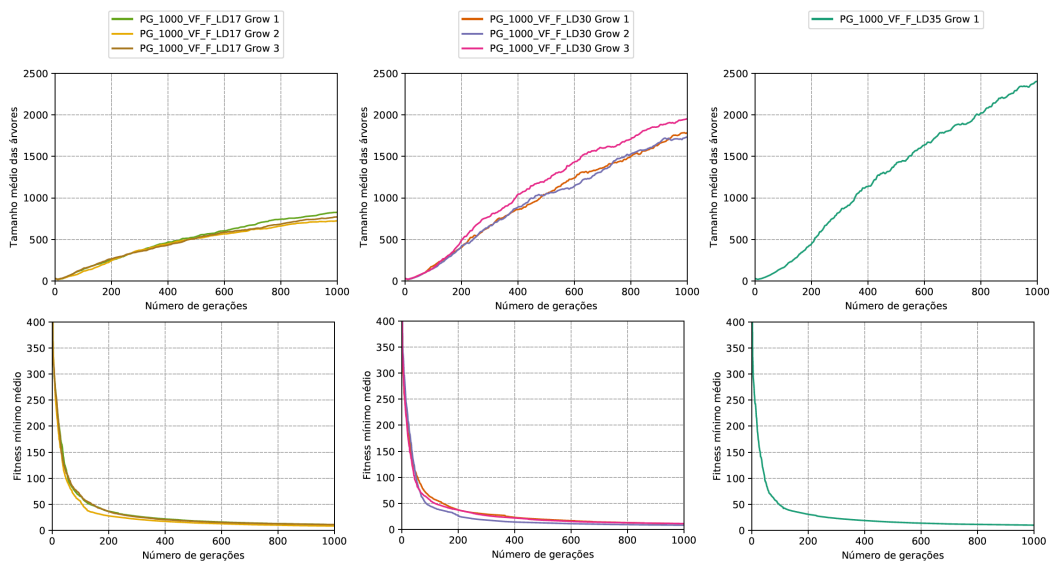


Figura 5.8: Fitness mínimo médio e tamanho médio das árvores dos conjuntos de experiências *PG_1000_VF_F_LD17*, *PG_1000_VF_F_LD30* e *PG_1000_VF_F_LD35* para o método de inicialização da população *Grow*, durante 1000 gerações.

Ainda que o limite dinâmico possa aumentar e diminuir durante a execução, o que se verificou é que durante as primeiras gerações este foi crescendo até atingir o valor máximo (valor do limite estático), acabando por excluir alguns modelos durante esse processo. Desta forma, o limite dinâmico forçou o algoritmo a encontrar melhores soluções mais rapidamente, atingindo valores de fitness mais baixos durante as gerações iniciais, porém necessitou de um elevado poder computacional. Uma prova disso é que para um limite estático com valor 35 só foi possível executar o algoritmo para uma das divisões da validação cruzada, devido ao tempo necessário para o processamento (aproximadamente um mês).

Uma outra abordagem explorada foi a utilização de um limite estático de valor 17 em conjunto com um limite dinâmico aplicado ao tamanho das árvores (número de nós). Os resultados para este conjunto de experiências encontram-se na Tabela 5.7.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro modelo (%)	Erro previsão (L/100km)	Nº de nós da solução
<i>PG_1000_VF_F_LDTA</i>	Full 1	1.8957	0.90%	±1.4267	689
	Full 2	2.8223	3.32%	±1.9944	558
	Full 3	2.6441	6.75%	±1.9935	596
	Grow 1	1.8790	2.77%	±1.3874	706
	Grow 2	1.4589	1.78%	±1.0782	701
	Grow 3	2.2139	1.13%	±3.9959	593
	R. Half-and-Half 1	1.2896	2.42%	±1.7904	662
	R. Half-and-Half 2	2.1111	4.38%	±1.1853	402
	R. Half-and-Half 3	2.7557	8.10%	±1.2284	388

Tabela 5.7: Resultados do conjunto de experiências *PG_1000_VF_F_LDTA* realizado durante a etapa de eliminação do *bloat*.

Comparando este conjunto de experiências com o conjunto *PG_1000_VF_F_DL17*, é notório que não existem grandes discrepâncias nos resultados obtidos, exceto no número de nós que compõem as soluções encontradas. Ao utilizar um limite dinâmico aplicado ao tamanho das árvores, as soluções encontradas apresentam resultados igualmente bons aos anteriores, mas são em média mais pequenas (Figura 5.9).

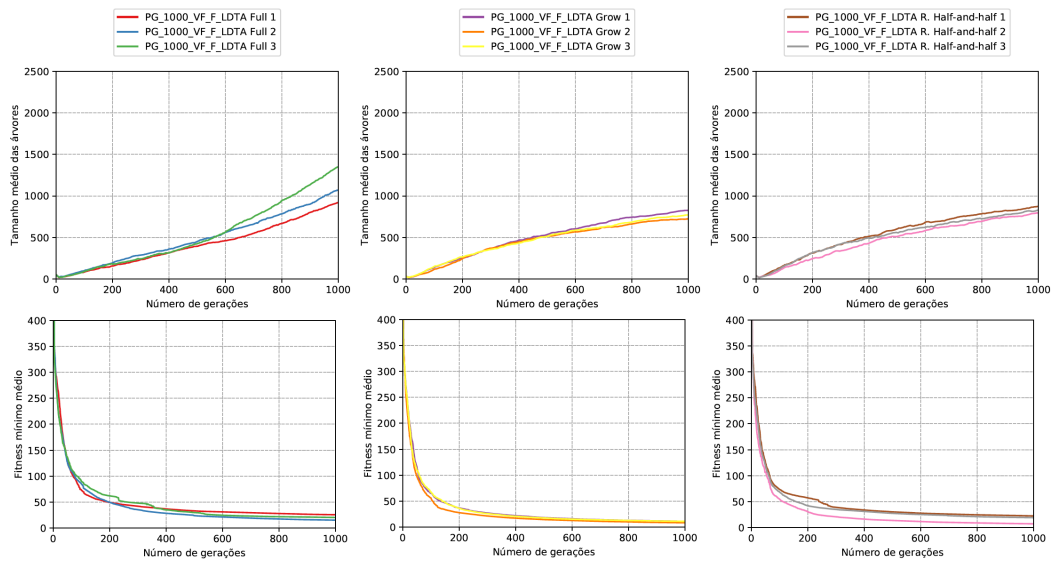


Figura 5.9: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_VF_F_LDTA* para os três métodos de inicialização da população, durante 1000 gerações.

Ainda que nos gráficos seja notório um comportamento semelhante no crescimento das árvores sem melhorias substanciais no valor de fitness (entre as gerações 900 e 1000 o fitness baixou, em média, 0.84), em especial para o método de inicialização Full, esse crescimento permitiu gerar soluções com resultados melhores.

5.3.2.3 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- A aplicação dos limites dinâmicos ao algoritmo de PG permitiu reduzir o tamanho das soluções encontradas;
- Durante a execução do algoritmo de PG com limites dinâmicos aplicados à profundidade das árvores, ainda que o valor máximo de profundidade tenha sido atingido ainda nas gerações iniciais e não tenha voltado a descer, foi possível reduzir o número de nós das soluções encontradas e gerar modelos com melhores resultados;
- Um limite máximo para a profundidade das árvores superior a 17 implica um maior tempo de execução, onde as melhorias encontradas não são proporcionais (valores de fitness não variam em mais de uma unidade);
- A utilização de um limite que permita realizar uma filtragem mais restrita, sendo aplicado ao número de nós e não à profundidade das árvores, apresentou resultados igualmente bons e soluções globalmente mais pequenas, porque contrariamente aos limites dinâmicos anteriores, este não estagna num valor máximo, eliminando um grande número de indivíduos ao longo das várias gerações.

5.3.3 Conclusões

Baseado nas experiências realizadas durante a 2ª fase é possível concluir que para o problema de previsão do consumo de combustível existem diferentes conjuntos de variáveis que podem gerar modelos semelhantes ao gerado manualmente pela empresa. Das experiências realizadas, aquelas que utilizaram o conjunto de variáveis de funcionamento do veículo e as forças de atuação durante o seu deslocamento apresentaram modelos muito próximos do modelo fornecido.

Os limites estáticos e dinâmicos são algumas das abordagens mais utilizadas para eliminar o *bloat* e reduzir o número de nós das soluções encontradas. A conjugação de ambos, ainda que não tenha conseguido cumprir nenhum desses dois, permitiu melhorar os resultados obtidos. Além disso, quando o limite dinâmico foi aplicado à profundidade das árvores com um limite estático de valor superior a 17 apresentou um maior tempo de processamento sem que tenham sido obtidas grandes melhorias. Pelo contrário, o limite dinâmico aplicado ao número de nós permitiu encontrar modelos mais pequenos com valores de fitness em média mais baixos. Isto acontece porque este limite tem influência durante todo o processamento do algoritmo, não estagnando num valor máximo.

Os modelos que se apresentaram mais próximos do modelo fornecido foram gerados pelas experiências *PG_1000_VF_F_LD17 Ramped Half-and-half 1* e *PG_1000_VF_F_LDTA Grow 2*.

5.4 3ª Fase

Os modelos resultantes das experiências da 2ª fase foram gerados com base em entradas de uma única viagem. Para essa viagem existem 878 entradas para as quais o valor de consumo foi estimado pelo modelo gerado manualmente pela empresa.

Em contexto real, a empresa não tem equipamentos capazes de medir o número de litros gastos a cada momento do funcionamento do veículo. Apenas é possível saber o número de litros abastecidos por cada veículo, assumindo que esse valor corresponde aos litros gastos nas viagens realizadas entre o abastecimento anterior e o considerado. No limite, se forem considerados períodos de abastecimento onde ocorreu apenas uma viagem, o número de litros abastecidos correspondem ao número de litros gastos na viagem.

Para validar os modelos gerados foram escolhidos 45 períodos de abastecimentos que contemplam apenas uma viagem cada um, tendo sido calculado o consumo acumulativo, em litros por 100 quilômetros, para cada um deles. As viagens consideradas foram realizadas por diferentes veículos pesados.

Para cada uma das viagens foi calculado o erro absoluto entre os diferentes modelos gerados na 2ª fase e o valor real de consumo, dado pela Equação 5.4.

$$Erro_absoluto(\%) = \frac{|consumo_{PG} - consumo_{real}|}{consumo_{real}} * 100 \quad (5.4)$$

onde $consumo_{PG}$ e $consumo_{real}$ representam, respetivamente, o consumo dado pelo modelo gerado com PG e o consumo real, ambos em litros por 100 quilômetros. Para cada um dos modelos, o valor $consumo_{PG}$ é dado pela equação 5.3.

Os resultados obtidos para cada uma das viagens encontram-se na tabela do Apêndice D.1.

5.4.1 Validação dos modelos gerados

Na Tabela 5.8 encontram-se as médias dos erros absolutos dos diferentes modelos gerados na fase anterior para as 45 viagens consideradas, bem como o desvio padrão e o número de nós que compõem cada uma das soluções.

Os resultados mostram que, para o conjunto de dados utilizado para validação, 11 dos 31 modelos gerados na fase anterior apresentaram erros absolutos médios mais baixos do que o erro do modelo fornecido, o que corresponde a 35% dos modelos gerados.

Para o conjunto de experiências *PG_1000_VF_F*, onde o algoritmo de PG não utiliza qualquer limite dinâmico, existem dois modelos que se demonstraram melhores do que o modelo fornecido. No entanto, o modelo com inicialização **Grow 3**, ainda que apresente melhorias

de aproximadamente 6%, tem um desvio padrão superior o que significa que, para algumas viagens, pode apresentar percentagens de erro maiores. Além disso, é composto por um elevado número de nós, onde grande parte deles podem ser redundantes dada a existência de *bloat* na geração das soluções para este conjunto de experiências, tal como se verificou na fase anterior.

Conjunto de experiências	Inicialização	Erro absoluto médio (%)	Desvio Padrão	Nº de nós da solução
<i>PG_1000_VF_F</i>	Full 1	62.65%	43.73%	1180
	Full 2	48.63%	42.14%	585
	Full 3	47.15%	36.23%	574
	Grow 1	61.37%	44.62%	604
	Grow 2	56.22%	43.83%	907
	Grow 3	41.44%	38.69%	1018
	R. Half-and-Half 1	48.54%	37.57%	624
	R. Half-and-Half 2	48.75%	34.73%	1070
	R. Half-and-Half 3	47.99%	42.56%	875
<i>PG_1000_VF_F_LD17</i>	Full 1	52.91%	36.64%	637
	Full 2	60.02%	34.60%	735
	Full 3	32.03%	16.52%	552
	Grow 1	64.24%	43.16%	634
	Grow 2	48.07%	42.49%	963
	Grow 3	52.57%	44.26%	569
	R. Half-and-Half 1	55.93%	38.65%	779
	R. Half-and-Half 2	39.92%	30.76%	461
	R. Half-and-Half 3	68.17%	34.58%	1030
<i>PG_1000_VF_F_LD30</i>	Grow 1	43.13%	24.76%	1517
	Grow 2	44.77%	38.29%	2362
	Grow 3	36.76%	21.41%	1150
<i>PG_1000_VF_F_LD35</i>	Grow 1	53.43%	38.24%	1983
<i>PG_1000_VF_F_LDTA</i>	Full 1	59.70%	43.26%	689
	Full 2	40.47%	24.34%	558
	Full 3	32.87%	16.10%	596
	Grow 1	33.59%	28.61%	706
	Grow 2	48.93%	35.88%	701
	Grow 3	64.47%	37.91%	593
	R. Half-and-Half 1	61.33%	21.45%	662
	R. Half-and-Half 2	35.81%	29.04%	402
	R. Half-and-Half 3	50.89%	40.00%	388
Modelo da empresa Stratio Automotive		47.35%	26.22%	-

Tabela 5.8: Resultados da validação dos modelos gerados pelos conjuntos de experiências: *PG_1000_VF_F*, *PG_1000_VF_F_LD17*, *PG_1000_VF_F_LD30*, *PG_1000_VF_F_LD35* e *PG_1000_VF_F_LDTA*.

Para os conjuntos de experiências em que foi aplicado um limite dinâmico à profundidade das árvores (*PG_1000_VF_F_LD17*, *PG_1000_VF_F_LD30* e *PG_1000_VF_F_LD35*), 5 dos modelos gerados apresentaram resultados melhores em comparação ao modelo da empresa. Desses três conjuntos de experiências é de salientar o modelo gerado na experiência *PG_1000_VF_F_LD17 Full 3* que apresenta um erro absoluto médio de 32.03% e um desvio-padrão de 16.52%.

Do conjunto de experiências *PG_1000_VF_F_LDTA* os modelos que se demonstraram melhores que o fornecido apresentam os valores de erro e desvio padrão globalmente mais baixos.

Os cinco melhores modelos gerados, apresentam valores de erro abaixo dos 40%, aproximadamente 7% abaixo do modelo fornecido, com um desvio padrão a não exceder os 30%.

5.4.2 Conclusões

Depois da fase de validação dos modelos é possível concluir que, com a técnica de PG, foi possível gerar modelos que se demonstraram semelhantes ou melhores na previsão do consumo real de combustível em comparação com o modelo fornecido pela empresa Stratio Automotivo. Além disso, tal como em fases anteriores, as soluções geradas utilizando limites dinâmicos apresentaram melhores resultados, onde muitos indivíduos foram eliminados durante a execução do algoritmo, levando a que as soluções obtidas sejam, em média, melhores que as restantes.

As soluções que apresentaram melhores resultados durante a fase de validação foram resultantes das experiências *PG_1000_VF_F_LD17 Full 3* e *PG_1000_VF_F_LDTA Full 3*, com erros de previsão de ± 26.24 e ± 25.78 litros por 100 quilómetros, respetivamente.

5.5 Conclusões gerais

No final das diferentes fases é possível concluir que, com a técnica de PG, foi possível cumprir o primeiro objetivo estabelecido, gerando um modelo baseado no modelo fornecido, mas dependente de um número inferior de variáveis, aproximadamente 45% das variáveis consideradas inicialmente, umas referentes ao funcionamento do veículo e outras referentes às forças que atuam durante o seu deslocamento.

Das várias técnicas de pré-processamento de seleção de atributos que foram exploradas, forçando os modelos gerados a serem constituídos por um menor número de variáveis, a que apresentou melhores resultados foi a que eliminou as variáveis que não foram constantemente incluídas nos modelos gerados quando o algoritmo apresentou um número de comparações insuficiente. No entanto, sendo fundamental para a empresa a eliminação das variáveis das características do motor do veículo devido aos custos da sua obtenção junto dos fabricantes, foi possível, em conjunto com os técnicos da empresa, encontrar um conjunto de variáveis de entrada mais restrito com o qual foram gerados vários modelos, alguns deles com resultados melhores do que os do modelo fornecido. Essas variáveis sendo referentes ao funcionamento do veículo e por isso sem qualquer relação com a constituição do motor, tornam os modelos gerados genéricos.

Com os limites dinâmicos não foi possível atenuar o *bloat* na geração de modelos de previsão do consumo de combustível. Isto revela que estes mecanismos podem não ser a melhor opção, quando o problema em análise contiver uma elevada complexidade, onde a utilização de um limite estático superior a 17 leva a que o algoritmo de PG apresente tempos de execução muito demorados (entre duas semanas a um mês por cada divisão treino/teste). O que se verificou é que, dado o tamanho populacional e o crescimento excessivo dos indivíduos, ao fim de poucas gerações os operadores genéticos ficam inutilizáveis, uma vez que o processo do cálculo do valor de fitness torna-se excessivamente longo. No entanto, ainda que nada tenha sido pedido por parte da empresa para reduzir o tamanho dos modelos gerados, a sua utilização demonstrou-se uma mais valia na obtenção de melhores modelos, constituídos por um número mais baixo de nós.

Ainda que tenham sido encontrados dois modelos em que o erro absoluto médio ronda os 30%, existem três motivos para os valores desses mesmos erros não serem ainda mais baixos. Em primeiro lugar, todos os modelos foram gerados em aproximação ao modelo fornecido pela empresa, criado manualmente em condições ideais, com o objetivo de, no limite, gerar um modelo com um erro semelhante, mas que não seja dependente de um certo tipo de variáveis. Em segundo lugar, o conjunto de dado com o qual os modelos foram gerados não mapeia em igual proporção certos comportamentos que se verificam na maioria dos dados reais, entre eles a existência de períodos em que o veículo continua em funcionamento e portanto a consumir combustível, mas nos quais não existe movimento. Por fim, dado que os registos não são feitos em intervalos de tempo constantes, algumas vezes o último registo é recolhido ainda o veículo está em deslocamento enquanto que noutras essa recolha acontece quando o veículo se encontra parado, representando grandes discrepâncias no valor de algumas variáveis.

A fase de validação revelou-se fundamental na avaliação do comportamento dos diferentes modelos gerados perante os dados reais, uma vez que aqueles que durante a 2ª fase apresentaram os melhores resultados não correspondem aos mesmos que obtiveram os erros absolutos médios mais baixos durante a fase de validação.

As soluções que apresentaram melhores resultados na fase de validação foram as seguintes: *PG_1000_VF_F_LD17 Full 3* e *PG_1000_VF_F_LDTA Full 3*, ambas constituídas por aproximadamente 500 nós. Estes modelos, para os dados utilizados para validação apresentam erros de previsão entre os 25 e os 27 litros por 100 quilómetros, aproximadamente 8 litros a menos do que o modelo da empresa.

Capítulo 6

Geração de modelo baseado em dados reais

Neste capítulo são apresentadas as experiências que têm como objetivo gerar um modelo que consiga mapear o consumo real de combustível de veículos pesados. A abordagem seguida é composta por duas fases, como mostra a Figura 6.1:

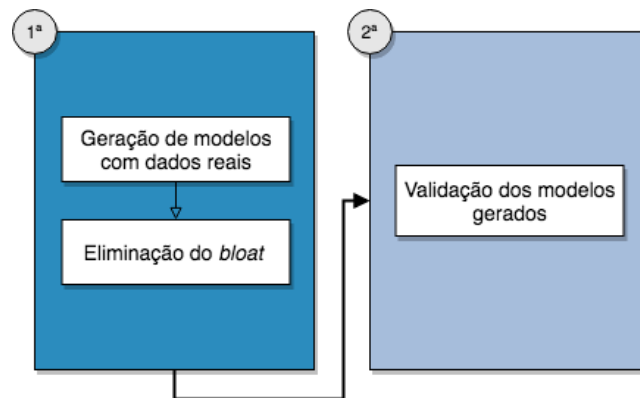


Figura 6.1: Arquitetura da abordagem da geração de um modelo baseado em dados reais.

- **1ª Fase:** Nesta fase foram realizados os conjuntos de experiências com os dados reais. Esta primeira fase encontra-se dividida nas seguintes etapas:
 1. **Geração de modelos com dados reais:** Nesta etapa foram realizadas experiências apenas com variáveis relativas ao funcionamento do veículo e às forças de atuação durante o seu deslocamento;
 2. **Eliminação do bloat:** Nesta etapa foram adicionados diferentes limites dinâmicos ao algoritmo de PG, na tentativa de atenuar ou eliminar o *bloat* presente nas experiências da etapa anterior.
- **2ª Fase - Validação dos modelos gerados:** Nesta fase foi realizada uma validação dos modelos gerados na fase anterior, aplicando-os a um conjunto de dados diferentes daqueles que foram utilizados para treino e teste.

Para os conjuntos de experiências realizados foi utilizado o 2º conjunto de dados, tal como está descrito em 4.3, e o 4º conjunto na validação dos modelos.

6.1 Considerações gerais

Nos conjuntos de experiências apresentados de seguida a função de fitness é novamente o EQM, desta vez entre o modelo gerado com PG e valor real de consumo (Equação 6.1). Quanto menor for esse valor, melhor é o indivíduo gerado.

$$EQM = \frac{\sum_{n=1}^n (x_{pgi} - x_{real_i})^2}{n} \quad (6.1)$$

onde x_{pgi} e x_{real_i} correspondem, para a viagem i , ao número de litros estimados pelo modelo de PG e ao número real de litros gastos, respetivamente.

Na Tabela 6.1 encontram-se as configurações que são transversais a todos os conjuntos de experiências apresentados neste capítulo.

Parâmetros	Valores
Número de execuções	30
Tamanho do torneio	3
Tamanho da elite	1
Probabilidade de cruzamento	0.9
Probabilidade de mutação	0.1
Tamanho mínimo das árvores da população inicial	4
Tamanho máximo das árvores da população inicial	6
Número de gerações	1000
Tamanho populacional	1000
Profundidade máxima das árvores da população	17

Tabela 6.1: Configurações do algoritmo de Programação Genética transversais a todos os conjuntos de experiências apresentados no capítulo 6.

O algoritmo de PG é mais uma vez executado 30 vezes, durante 1000 gerações, para uma população de 1000 indivíduos. No final dessas execuções, o melhor modelo gerado é devolvido como solução do problema. Durante a execução do algoritmo, a população inicial é criada de forma aleatória por um dos três métodos de inicialização considerados, em que as árvores geradas apresentam uma profundidade entre 4 e 6, podendo aumentar até ao valor máximo 17 (limite estático). A cada uma delas é atribuído um valor de fitness, consoante a sua capacidade para resolver o problema em análise.

Mais uma vez, a probabilidade de cruzamento é de 90% e a probabilidade de mutação é de 10%. A seleção é feita por torneio de tamanho 3 e o parâmetro elitismo tem valor 1. O conjunto de funções manteve-se e o conjunto de símbolos terminais é composto pelas Constantes Aleatórias Efémeras (CAE) e pelo seguinte conjunto de variáveis: tempo, velocidade média, distância, altitude, raio, força de arrasto aerodinâmico, energia, rendimento

de transmissão e número de rotações do motor. Este conjunto não inclui a totalidade das 14 variáveis que se encontram descritas na secção 4.4.2 e que foram consideradas nas experiências de aproximação ao modelo fornecido, uma vez que algumas delas apresentavam um elevado número de valores em falta e por isso não foram consideradas.

A todas as experiências foi aplicada uma truncação ao valor de saída do modelo, uma vez que não existem valores de consumo negativos (Equação 6.2).

$$\text{consumo}(\text{Litros}) = \begin{cases} 0 & \text{modelo}_{PG} < 0 \\ \text{modelo}_{PG} & 0 \leq \text{modelo}_{PG} \end{cases} \quad (6.2)$$

6.2 1ª Fase

Nesta secção são apresentados os resultados das experiências realizadas com os dados reais do funcionamento dos veículos e as forças de atuação durante o seu deslocamento, bem como as conclusões retiradas.

6.2.1 Geração de modelos com dados reais

A primeira etapa tem como objetivo gerar um modelo de previsão do consumo de combustível baseado nos dados reais, utilizando apenas as variáveis para as quais não existem valores em falta no 2º conjunto de dados.

Nesta etapa não existem configurações adicionais, além das apresentadas na secção 6.1.

6.2.1.1 Resultados

Os resultados para o primeiro conjunto de experiências encontram-se na Tabela 6.2.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro previsão (Litros)	Nº de nós da solução
<i>PG_1000_DR</i>	Full 1	284.35	±36.40	1129
	Full 2	307.59	±25.50	1128
	Full 3	273.49	±44.99	816
	Grow 1	317.06	±30.17	1264
	Grow 2	210.92	±32.02	837
	Grow 3	249.94	±30.10	846
	R. Half-and-Half 1	249.57	±84.07	1232
	R. Half-and-Half 2	312.40	±268.94	2045
	R. Half-and-Half 3	274.23	±33.61	1314

Tabela 6.2: Resultados do conjunto de experiências *PG_1000_DR* realizado durante a etapa da geração de modelos com dados reais.

Perante os resultados é notório que os valores de fitness e os erros de previsão dos modelos gerados são superiores aos obtidos nas experiências do capítulo anterior. Ainda assim, é de salientar que as soluções que apresentam os melhores resultados foram geradas com o método de inicialização **Grow**, sendo também aquelas que apresentam um número inferior

de nós. Dessas soluções é de destacar a gerada pela experiência *PG_1000_DR Grow 3* com um erro de previsão de 30 litros, aproximadamente 13.6% da capacidade de um tanque de combustível dos veículos pesados em análise.

Baseado nos gráficos da Figura 6.2 é visível a existência de um crescimento constante no tamanho das árvores ao longo das gerações, sem que se verifiquem melhorias proporcionais nos valores de fitness dos indivíduos a partir da geração 800. Isto acontece para os três métodos de inicialização, que apresentam resultados muito semelhantes para o valor de fitness mínimo médio e para o tamanho médio das árvores, durante a execução do algoritmo de PG.

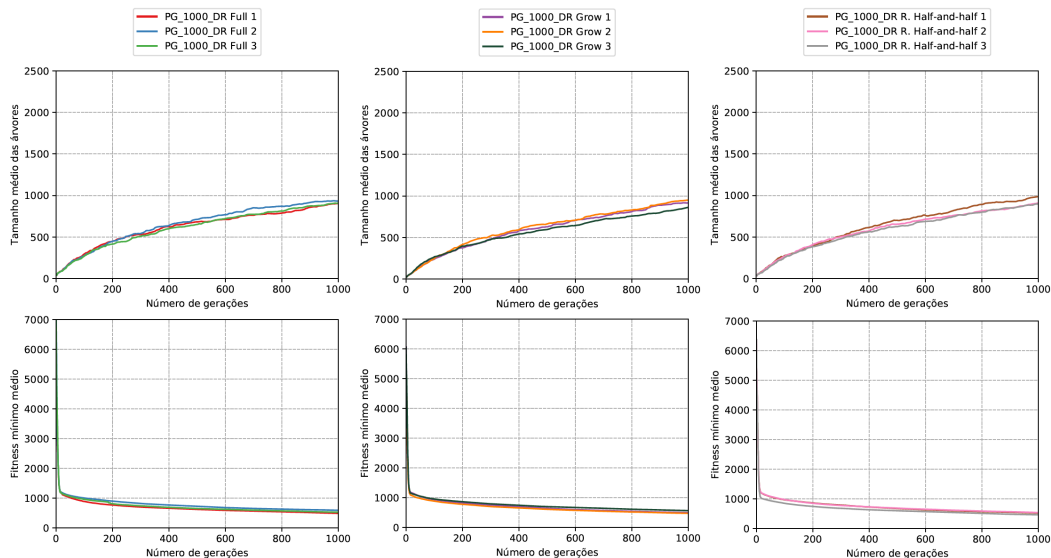


Figura 6.2: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_DR* para os três métodos de inicialização da população, durante 1000 gerações.

6.2.1.2 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- Os resultados obtidos para as primeiras experiências da geração de um modelo baseado em dados reais são piores do que os obtidos na geração de um modelo baseado no modelo fornecido, uma vez que existem diversos fatores presentes nos dados de treino que são difíceis de mapear (viagens com grandes períodos de funcionamento sem que haja movimento);
- À semelhança das experiências do capítulo anterior, o método de inicialização da população que apresenta maioritariamente melhores resultados foi o *Grow*, gerando soluções mais pequenas e com erros de previsão mais baixos;
- O crescimento constante das árvores ao longo das gerações sem um melhoramento proporcional no valor de fitness dos indivíduos gerados deve-se a uma das limitações da PG, o *bloat*;

- Tendo em consideração os resultados do capítulo anterior, podem ser utilizados limites dinâmicos em conjunto com o limite estático de valor 17 que, ainda que mais uma vez possam não conseguir atenuado o *bloat*, podem ser uma mais valia na melhoria dos resultados obtidos.

6.2.2 Eliminação do *bloat*

Nesta etapa foram aplicados diferentes limites dinâmicos no sentido de atenuar o *bloat* existente no conjunto de experiências anterior. Assim sendo, foram considerados dois tipos de limites dinâmicos, um aplicado à profundidade das árvores e outro aplicado ao número de nós dos indivíduos, em conjunto com o limite estático de valor 17.

De seguida são apresentados os resultados obtidos e as conclusões retiradas.

6.2.2.1 Resultados

Os resultados obtidos da aplicação dos dois tipos de limites dinâmicos encontram-se na Tabela 6.3.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro previsão (Litros)	Nº de nós da solução
<i>PG_1000_DR_LD17</i>	Full 1	249.02	±35.22	888
	Full 2	272.66	±27.45	989
	Full 3	232.91	±38.81	1382
	Grow 1	202.56	±67.76	1063
	Grow 2	155.57	±29.90	3120
	Grow 3	159.90	±33.55	1488
	R. Half-and-Half 1	199.91	±33.87	2001
	R. Half-and-Half 2	257.24	±87.38	1146
	R. Half-and-Half 3	104.30	±30.02	2000
<i>PG_1000_DR_LDTA</i>	Full 1	295.45	±27.19	3395
	Full 2	91.73	±32.88	2689
	Full 3	117.11	±33.49	3530
	Grow 1	164.45	±35.25	869
	Grow 2	222.04	±26.59	2339
	Grow 3	242.23	±39.37	2133
	R. Half-and-Half 1	176.71	±29.90	3003
	R. Half-and-Half 2	184.88	±38.36	1869
	R. Half-and-Half 3	262.32	±26.60	2738

Tabela 6.3: Resultados dos conjuntos de experiências *PG_1000_DR_LD17* e *PG_1000_DR_LDTA* realizado durante a etapa da eliminação do *bloat*.

Os conjuntos de experiências em que foram aplicados limites dinâmicos, geraram algumas soluções com resultados melhores do que os apresentados na etapa anterior. No entanto, as soluções geradas são compostas por um maior número de nós.

O conjunto que apresentou melhores resultados foi aquele que utilizou um limite dinâmico aplicado ao tamanho das árvores (*PG_1000_DR_LDTA*). Desse conjunto é possível salientar a solução encontrada na experiência *PG_1000_DR_LDTA Full 2*, com um valor de fitness de 91.73 e um erro de previsão de ±32.88 litros.

Para o conjunto de experiências *PG_1000_DR_LD17*, os gráficos da Figura 6.3 mostram que a utilização de um limite dinâmico aplicado à profundidade das árvores da população

não permitiu atenuar o *bloat*, porém a sua utilização permitiu mais uma vez gerar soluções com melhores resultados, forçando o algoritmo a atingir mais rapidamente valores mais baixos de fitness.

A utilização de um limite dinâmico aplicado ao número de nós, ainda que também não tenha conseguido eliminar o *bloat*, onde o crescimento das árvores se tornou quase que exponencial para todos os métodos de inicialização (Figura 6.4), permitiu gerar as melhores soluções encontradas até ao momento para as experiências baseadas nos dados reais de consumo.

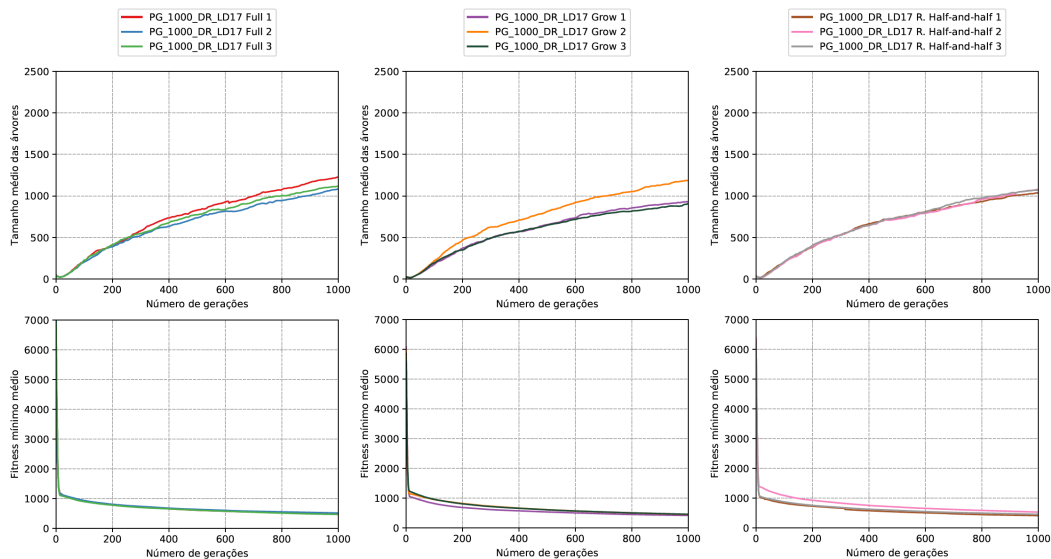


Figura 6.3: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_DR_LD17* para os três métodos de inicialização da população, durante 1000 gerações.

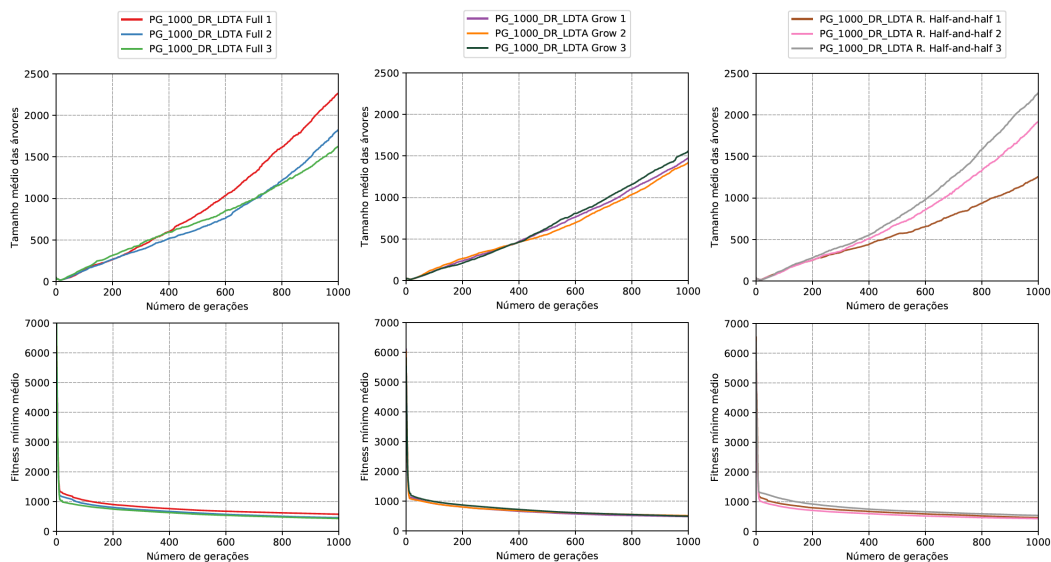


Figura 6.4: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_DR_LDTA* para os três métodos de inicialização da população, durante 1000 gerações.

6.2.2.2 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- A utilização de limites dinâmicos ainda que não tenha sido eficiente na eliminação do *bloat*, permitiu gerar vários modelos com resultados melhores do que os apresentados na etapa anterior;
- Para o limite aplicado à profundidade das árvores o que se verificou mais uma vez é que este atinge o seu valor máximo (valor do limite estático) durante as gerações iniciais e não volta a descer;
- A utilização de um limite dinâmico com uma filtragem mais restrita, sendo aplicado ao tamanho das árvores, foi o que gerou as melhores soluções, uma vez que quando o limite máximo da profundidade das árvores é atingido (limite estático), estas continuam a crescer horizontalmente;
- Dos modelos gerados, aquele que se destaca é o *PG_1000_DR_LDTA Full 2*, com um valor de fitness de 91.73 e um erro de previsão de ± 32.88 litros, aproximadamente 15% da capacidade total do tanque do veículo pesado.

6.2.3 Conclusões

No final da primeira fase é possível concluir que, de maneira geral, as experiências de regressão realizadas com dados reais têm tendência a apresentar piores resultados do que quando são realizadas em aproximação a um conjunto de dados para o qual se sabe que existe um modelo capaz de os mapear. Isto acontece, em grande parte devido às características do conjunto de dados de treino que se tornam difíceis de mapear, as mesmas que foram enunciadas no final do capítulo 5.

A conjugação de diferentes limites dinâmicos com o limite estático de valor 17 ainda que não tenha atenuado o *bloat*, permitiu atingir valores mais baixos de fitness, em alguns casos com melhorias de 50%. Dos limites dinâmicos considerados, aquele que é aplicado ao número de nós foi o que gerou as melhores soluções.

Até ao final desta fase, o modelo que apresentou os melhores resultados foi gerado pela experiência *PG_1000_DR_LDTA Full 2*, com um valor de fitness de 91.73 e um erro de previsão de 32.88 litros, aproximadamente 15% da capacidade total do tanque do veículo pesado. Contudo, é necessário aplicar cada um dos modelos gerados a um novo conjunto de dados para comprovar a validade dos resultados obtidos.

6.3 2ª Fase

Com o objetivo de validar os resultados obtidos na fase anterior foram aplicados os modelos gerados a um conjunto de dados diferentes daqueles que foram usados para treino e teste.

Assim sendo, foram calculados os erros absolutos entre o número de litros abastecidos e a soma dos consumos das duas viagens, estimados por cada um dos modelos. O erro absoluto é calculado com a Equação 5.4, onde o valor de $consumo_{PG}$ é dado pela Equação:

$$consumo_{PG} = consumo_{PG_1} + consumo_{PG_2} \quad (6.3)$$

em que $consumo_{PG_1}$ e $consumo_{PG_2}$ correspondem aos consumos (Equação 6.2), em litros, estimados para a viagem um e dois, respetivamente.

No entanto, ao considerar períodos de abastecimento com mais do que uma viagem faz com que os valores da validação seguinte possam, em algumas situações, não corresponder à realidade. Por exemplo, se para a primeira viagem a estimativa apresentar cinco litros a mais do que o que foi realmente consumido e a segunda viagem apresentar uma estimativa de cinco litros a menos, quando os consumos são somados esses erros anulam-se.

Para validar os modelos foram utilizados 28 períodos de abastecimento, perfazendo um total de 56 viagens.

6.3.1 Validação dos modelos gerados

Na Tabela 6.4 encontram-se as médias dos erros absolutos, os desvios padrão e os números de nós que compõem cada um dos diferentes modelos gerados durante a fase anterior. Os erros absolutos para cada período de abastecimento encontram-se no Apêndice D.2.

Conjunto de experiências	Inicialização	Erro absoluto médio (%)	Desvio Padrão	Nº de nós da solução
<i>PG_1000_DR</i>	Full 1	69.19%	32.93%	1129
	Full 2	76.99%	37.04%	1128
	Full 3	74.93%	31.90%	816
	Grow 1	74.15%	41.52%	1264
	Grow 2	80.82%	48.08%	837
	Grow 3	91.59%	43.77%	846
	R. Half-and-Half 1	79.55%	32.62%	1232
	R. Half-and-Half 2	72.80%	41.28%	2045
	R. Half-and-Half 3	71.31%	37.98%	1314
<i>PG_1000_DR_LD17</i>	Full 1	68.96%	37.65%	888
	Full 2	75.60%	51.47%	989
	Full 3	87.62%	45.96%	1382
	Grow 1	67.62%	37.26%	1063
	Grow 2	71.53%	44.18%	3120
	Grow 3	81.33%	44.38%	1488
	R. Half-and-Half 1	74.39%	35.82%	2001
	R. Half-and-Half 2	74.59%	37.54%	1146
	R. Half-and-Half 3	68.89%	39.11%	2000
<i>PG_1000_DR_LDTA</i>	Full 1	81.54%	41.32%	3395
	Full 2	70.82%	40.03%	2689
	Full 3	77.79%	40.97%	3530
	Grow 1	81.85%	47.47%	869
	Grow 2	76.42%	32.34%	2339
	Grow 3	73.74%	43.16%	2133
	R. Half-and-Half 1	71.14%	44.82%	3003
	R. Half-and-Half 2	58.53%	39.93%	1869
	R. Half-and-Half 3	79.35%	43.26%	2738
Modelo da empresa Stratio Automotive		72.00%	25.22%	-

Tabela 6.4: Resultados da validação dos modelos gerados pelos conjuntos de experiências: *PG_1000_DR*, *PG_1000_DR_LD17* e *PG_1000_DR_LDTA*.

Para os períodos de abastecimento utilizados para validação, 9 dos 27 modelos apresentaram erros absolutos médios mais baixos do que o erro do modelo gerado manualmente pela empresa, o que representa 33.33% dos modelos gerados.

Para o conjunto de experiências *PG_1000_DR*, em que não foram utilizados quaisquer limites dinâmicos, existem dois modelos que apresentam melhores previsões do que o modelo da empresa, porém têm um desvio padrão superior. Já os conjuntos de experiências *PG_1000_DR_LD17* e *PG_1000_DR_LDTA*, em que foram utilizados limites dinâmicos aplicados à profundidade e ao tamanho das árvores respectivamente, geraram um maior número de soluções com resultados melhores do que os apresentados pelo modelo fornecido.

O conjunto de experiências *PG_1000_DR_LD17*, ainda que tenha gerado um maior número de melhores soluções, foi o conjunto *PG_1000_DR_LDTA* que gerou o modelo que apresenta o erro absoluto médio mais baixo. Esse modelo é resultante da experiência *PG_1000_DR_LDTA R. Half-and-half 2* com um erro de 58.53%, aproximadamente 93.05 litros de erro por cada 100 quilômetros. No entanto, é de frisar que os períodos de abastecimento considerados são compostos por duas viagens, o que influencia os resultados obtidos.

As melhorias conseguidas face ao modelo da empresa rondam os 13% , contudo os desvios padrão apresentados são superiores.

6.3.2 Conclusões

Com a PG foi possível gerar modelos que demonstraram ser melhores na previsão do consumo de combustível dos veículos pesados em comparação ao modelo criado manualmente pela empresa Stratio Automotive.

Mais uma vez, as experiências em que foram utilizados limites dinâmicos conseguiram gerar um maior número de modelos com erros absolutos, em média, mais baixos. Isto deve-se à quantidade de indivíduos com elevados valores de fitness que foram excluídos durante as gerações iniciais.

O modelo que apresentou melhores resultados durante a fase de validação foi resultante da experiência *PG_1000_DR_LDTA R. Half-and-half 2* apresentando o erro absoluto médio mais baixo, com erros de previsão mais baixos do que o modelo da empresa em aproximadamente 27 litros por cada 100 quilômetros. Contudo, estes dados podem não corresponder à realidade, dado o conjunto de dados utilizados para validação.

6.4 Conclusões gerais

Com a técnica de PG foi possível cumprir o segundo objetivo estabelecido, gerando um modelo com base em dados reais apenas com as variáveis que no capítulo anterior se demonstraram essenciais. Contudo, algumas dessas variáveis foram excluídas do conjunto de entrada por apresentarem valores em falta, tendo sido consideradas apenas 9 variáveis, aproximadamente 25% das variáveis iniciais. Essas variáveis sendo comuns a veículos pesados de diferentes marcas e modelos tornam os modelos genéricos.

A conjugação dos limites dinâmicos com o limite estático de valor 17 não permitiu atenuar o *bloat*, porém gerou soluções com melhores resultados ainda que com um maior número de nós. Os limites dinâmicos aplicados ao tamanho das árvores permitiram gerar soluções com valores de fitness, em média, mais baixos e com erros de previsão menores.

A fase de validação mais uma vez mostrou-se fundamental, dado que os modelos que apresentaram melhores resultados durante essa fase não correspondem aos mesmos que obtiveram os melhores resultados durante a 1ª fase. Contudo, ao serem utilizados períodos de abastecimento compostos por duas viagens, ainda que em algumas situações os erros de ambas as viagens possam anular-se, noutras isso não se verifica, gerando erros de previsão superiores do que quando são utilizados períodos de abastecimento com apenas uma viagem. Além disso, o conjunto de dados de entrada apresenta comportamentos difíceis de mapear, os quais já foram enumerados no final do capítulo anterior.

Capítulo 7

Incorporação de variáveis meteorológicas

Neste capítulo são apresentadas as experiências que têm como objetivo estudar a influência dos fatores meteorológicos na previsão do consumo de combustível. A abordagem seguida encontra-se dividida em duas partes, cada uma com duas fases (Figura 7.1):

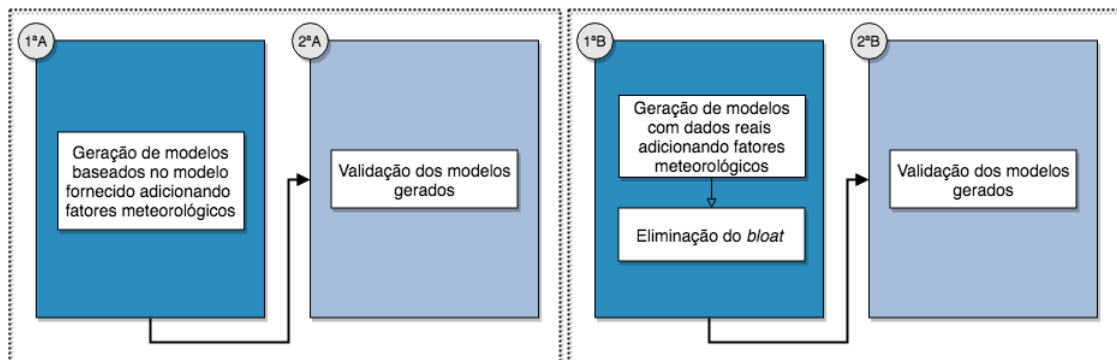


Figura 7.1: Arquitetura da abordagem da incorporação de variáveis meteorológicas.

- **1ª Fase A - Geração de modelos baseados no modelo fornecido adicionando fatores meteorológicos:** Nesta fase foram realizados vários conjuntos de experiências gerando modelos baseados no modelo fornecido, incorporando fatores meteorológicos no conjunto de variáveis de entrada, com o objetivo de avaliar a influência que estes têm na previsão do consumo de combustível;
- **2ª Fase A - Validação dos modelos gerados:** Nesta fase foi realizada uma validação dos modelos gerados na fase anterior, aplicando-os a um conjunto de dados diferentes daqueles que foram utilizados para treino e teste;
- **1ª Fase B:** Nesta fase foram realizados vários conjuntos de experiências com os dados reais e com os dados meteorológicos, avaliando a influência que estes últimos têm na previsão do consumo de combustível. Esta fase encontra-se dividida em duas etapas:

1. **Geração de modelos com os dados reais adicionando fatores meteorológicos:** Nesta etapa foram realizadas experiências com o mesmo conjunto de variáveis do capítulo anterior, adicionando-lhe as variáveis meteorológicas;
 2. **Eliminação do *bloat*:** Nesta etapa foram adicionados dois tipos de limites dinâmicos ao algoritmo de PG na tentativa de atenuar ou eliminar o *bloat* existente no conjunto de experiências da etapa anterior.
- **2ª Fase B - Validação dos modelos gerados:** Nesta fase foi realizada uma validação dos modelos gerados na fase anterior, aplicando-os a um conjunto de dados diferentes daqueles que foram utilizados para treino e teste.

Para as experiências realizadas foi utilizado o 1º conjunto de dados na 1ª fase A e o 2º na 1ª fase B. Os 3º e 4º conjuntos de dados foram utilizados nas 2ªs fases de validação A e B, respectivamente.

7.1 Considerações gerais

Nos conjuntos de experiências apresentados de seguida a função de fitness é novamente o EQM, tal como mostra a equação 6.1. Quanto menor for esse valor, melhor é o indivíduo gerado.

As configurações transversais a todos os conjuntos de experiências são as mesmas que foram descritas no capítulo anterior e que se encontram na Tabela 6.1.

O conjunto de funções foi mantido e o conjunto de símbolos terminais é composto por Constantes Aleatórias Efémeras (CAE) e pelo seguinte conjunto de variáveis: tempo; velocidade média; distância; altitude; raio; força de arrasto aerodinâmico; energia; rendimento de transmissão; número de rotações do motor; pluviosidade; pluviosidade média; temperatura máxima, média e mínima; pressão atmosférica máxima, média e mínima; velocidade máxima, média e mínima do vento e direção média do vento.

7.2 1ª Fase A

Nesta fase foram incorporadas as variáveis meteorológicas ao conjunto de símbolos terminais das experiências que geraram modelos em aproximação ao modelo fornecido.

Nas seguintes secções encontram-se os resultados obtidos e as conclusões retiradas durante esta fase.

7.2.1 Geração de modelos baseados no modelo fornecido adicionando fatores meteorológicos

A primeira etapa desta fase tem como objetivo determinar um modelo, baseado no modelo fornecido, incorporando fatores meteorológicos na sua geração.

Nesta etapa não existem configurações adicionais, além das apresentadas em 6.1, e os resultados das experiências realizadas nesta fase encontram-se na Tabela 7.1.

Conjunto de experiências	Experiência	Fitness mínimo	Erro modelo (%)	Erro previsão (L/100km)	Nº de variáveis da solução
<i>PG_1000_VF_F_CM</i>	Full 1	0.88	0.18%	±0.79	572
	Full 2	0.53	0.59%	±0.59	804
	Full 3	0.48	0.12%	±1.42	991
	Grow 1	0.49	1.53%	±0.85	726
	Grow 2	0.54	0.40%	±0.68	705
	Grow 3	0.60	0.25%	±1.29	438
	R. Half-and-Half 1	0.52	1.68%	±0.89	679
	R. Half-and-Half 2	0.47	0.03%	±0.69	479
	R. Half-and-Half 3	0.67	2.05%	±2.26	670

Tabela 7.1: Resultados do conjunto de experiências *PG_1000_VF_F_CM* realizado durante a etapa da geração de modelos baseados no modelo fornecido, incorporando fatores meteorológicos.

Os modelos gerados pelo conjunto de experiências *PG_1000_VF_F_CM* apresentaram valores de fitness muito próximos do valor ótimo (próximos de zero). Além disso, apresentam percentagens de erro face ao modelo fornecido inferiores a 2% e erros de previsão, em média, de 0.57 litros por 100 quilómetros o que numa viagem de 100 quilómetros corresponde a 0.26% da capacidade do depósito.

Das soluções geradas é de salientar a solução resultante da experiência *PG_1000_VF_F_CM R. Half-and-half 2* que apresenta o valor de fitness, bem como o erro em relação ao modelo e o erro de previsão entre os valores mais baixos atingidos.

Nos gráficos da Figura 7.2 é notório que o tamanho das árvores está quase estagnado durante as últimas gerações, quando já não se verificam melhorias no valor de fitness dos indivíduos.

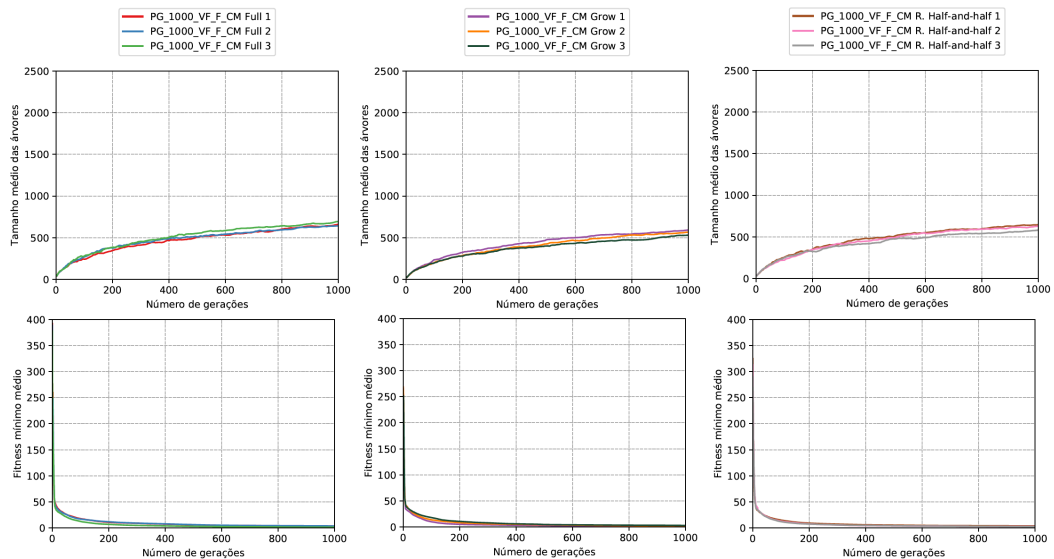


Figura 7.2: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_VF_F_CM* para os três métodos de inicialização da população, durante 1000 gerações.

7.2.2 Conclusões

Conjugando os dados do funcionamento do veículo, as forças de atuação durante o deslocamento e os resumos diários das condições meteorológicas foi possível com a PG gerar modelos muito semelhantes ao fornecido, com valores de fitness muito próximos de zero e com erros face ao modelo de, no máximo, 2%.

O modelo que apresenta melhores resultados durante a fase de teste foi resultante da experiência *PG_1000_VF_F_CM R. Half-and-half 2*, com erros de previsão a rondar os 0.5 litros por 100 quilómetros.

Dado que o crescimento das árvores se encontra quase que estagnado, não se justifica a aplicação das técnicas de eliminação de *bloat*.

7.3 2ª Fase A

Para validar os modelos gerados na fase anterior estes foram aplicados a um conjunto de dados reais. O conjunto de dados utilizados foi o mesmo que foi usado na validação das experiências do capítulo 5, para que seja possível perceber a influência que os fatores meteorológicos têm na previsão do consumo de combustível.

7.3.1 Validação dos modelos gerados

Na Tabela 7.2 encontram-se as médias dos erros absolutos, os desvios padrão e o número de nós que compõem cada uma das soluções encontradas. Os erros absolutos para as 45 viagens encontram-se no Apêndice D.3.

Conjunto de experiências	Inicialização	Erro absoluto médio (%)	Desvio Padrão	Nº de nós da solução
<i>PG_1000_VF_F_CM</i>	Full 1	69.18%	32.93%	572
	Full 2	76.99%	37.04%	804
	Full 3	82.76%	18.77%	991
	Grow 1	45.58%	38.40%	726
	Grow 2	79.57%	32.97%	705
	Grow 3	43.05%	25.07%	438
	R. Half-and-Half 1	31.49%	26.59%	679
	R. Half-and-Half 2	56.57%	34.22%	479
	R. Half-and-Half 3	62.77%	35.07%	670
	Modelo da empresa Stratio Automotive		47.35%	26.22%

Tabela 7.2: Resultados da validação dos modelos gerados pelo conjunto de experiências *PG_1000_VF_F_CM*.

Os resultados mostram que, para o conjunto de dados utilizados para validação, 3 dos 9 modelos gerados apresentam erros absolutos médios mais baixos do que o erro do modelo fornecido, o que corresponde a 33.33% dos modelos gerados.

O modelo que apresenta melhores resultados foi gerado pela experiência *PG_1000_VF_F_CM R. Half-and-half 1* com um erro absoluto médio de 31.49%, o que corresponde a um erro

de ± 21.04 litros para um viagem de 100 quilómetros, e com um desvio padrão de 26.59%, semelhante ao do modelo fornecido.

7.3.2 Conclusões

No final da validação é possível concluir que vários dos modelos gerados demonstraram-se melhores na previsão do consumo de combustível dos veículos pesados em comparação com o modelo da empresa.

O melhor modelo foi resultante da experiência *PG_1000_VF_F_CM R. Half-and-half 1* com melhorias de aproximadamente 15% comparativamente ao modelo fornecido, o que corresponde a um erro de previsão inferior em 13.25 litros por 100 quilómetros.

7.4 1ª Fase B

Nesta fase foram incorporadas as variáveis meteorológicas no conjunto de símbolos terminais das experiências em que os modelos gerados são aproximados ao valor real de consumo.

Nesta secção encontram-se os resultados dos conjuntos de experiências realizadas e as conclusões retiradas.

7.4.1 Geração de modelos com os dados reais adicionando fatores meteorológicos

A primeira etapa desta fase tem como objetivo gerar um modelo de previsão do consumo de combustível, utilizando os dados reais em conjunto com os dados meteorológicos.

Nesta etapa não existem configurações adicionais, além das apresentadas em 6.1.

7.4.1.1 Resultados

Os resultados para o primeiro conjunto de experiências encontram-se na Tabela 7.3.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro previsão (Litros)	Nº de nós da solução
<i>PG_1000_DR_CM</i>	Full 1	347.75	± 29.51	874
	Full 2	231.65	± 55.42	1165
	Full 3	268.48	± 34.10	955
	Grow 1	198.48	± 34.25	953
	Grow 2	325.36	± 51.77	734
	Grow 3	314.57	± 41.02	652
	R. Half-and-Half 1	252.82	± 34.30	944
	R. Half-and-Half 2	225.38	± 32.21	696
	R. Half-and-Half 3	218.39	± 40.74	717

Tabela 7.3: Resultados do conjunto de experiências *PG_1000_DR_CM* realizado durante a etapa da geração de modelos com dados reais, incorporando fatores meteorológicos.

Globalmente, os modelos obtidos apresentam um erro de previsão, em média, de 39 litros o que corresponde a aproximadamente 18% da capacidade do depósito.

Dos modelos gerados, aqueles que apresentam os melhores resultados foram gerados com o método de inicialização **Ramped Half-and-half**, apresentando soluções com valores de fitness e erros de previsão, em média, mais baixos. No entanto, além da solução encontrada na experiência **R. Half-and-half 3** é de salientar a experiência **Grow 1** que apresenta um valor de fitness inferior e um erro de previsão semelhante.

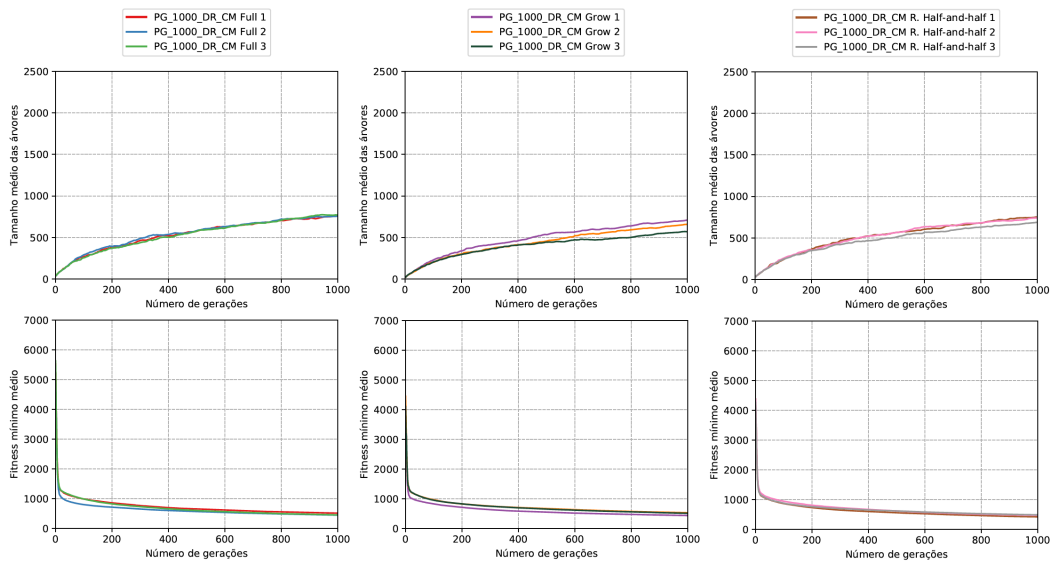


Figura 7.3: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_DR_CM* para os três métodos de inicialização da população, durante 1000 gerações.

À semelhança do que se verificou nas experiências dos capítulos anteriores, é visível um crescimento constante no tamanho das árvores ao longo das gerações sem que se verifique melhorias no valor médio do fitness mínimo. Ainda que este crescimento apresente uma tendência para estagnar durante as últimas gerações, este mantém-se crescente mesmo quando já não são se verificam melhorias nos modelos gerados. Este comportamento é exibido pelos três métodos de inicialização da população, tal como mostram os gráficos da Figura 7.3.

7.4.1.2 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- Com a PG foi possível gerar modelos com erros de previsão de aproximadamente 39 litros, 18% da capacidade do tanque de um veículo pesado do qual foram recolhidos os dados;
- A melhor solução foi gerada pela experiência *PG_1000_DR_CM Grow 1*, com um valor de fitness de 198.48 e um erro de previsão de ± 34.25 litros;

- À semelhança ao que se tem verificado em experiências anteriores, a geração dos modelos tem apresentado uma das limitações da PG, o *bloat*;
- O crescimento das árvores demonstrou uma tendência para vir a estagnar, em que durante as últimas gerações já não existiram quaisquer melhorias nos indivíduos da população.

7.4.2 Eliminação do *bloat*

Nesta etapa foram aplicados diferentes limites dinâmicos ao algoritmo de PG. Ainda que até ao momento não se tenham demonstrado a melhor abordagem na eliminação ou atenuação do *bloat*, permitem gerar soluções com resultados globalmente melhores. Além disso, o interesse da empresa é num modelo que efetue previsões o mais exatas possíveis, não existindo qualquer restrição quanto ao tamanho do modelo.

Nesta etapa não existem configurações adicionais, além das apresentadas em 6.1.

7.4.2.1 Resultados

Os resultados obtidos da aplicação dos dois tipos de limites dinâmicos encontram-se na Tabela 7.4.

Conjunto de experiências	Inicialização	Fitness mínimo	Erro previsão (Litros)	Nº de nós da solução
<i>PG_1000_DR_CM_LD17</i>	Full 1	282.59	±43.47	1131
	Full 2	244.71	±34.42	867
	Full 3	215.41	±38.25	878
	Grow 1	207.45	±36.94	859
	Grow 2	223.65	±27.42	4091
	Grow 3	172.81	±30.28	2434
	R. Half-and-Half 1	277.92	±37.96	813
	R. Half-and-Half 2	294.27	±5138.94	906
	R. Half-and-Half 3	108.05	±33.64	3347
		Full 1	103.88	±35.93
<i>PG_1000_DR_CM_LDTA</i>	Full 2	240.38	±55.68	1253
	Full 3	146.20	±33.36	1194
	Grow 1	225.72	±28.14	1041
	Grow 2	172.01	±39.74	1457
	Grow 3	237.51	±31.48	1561
	R. Half-and-Half 1	245.04	±34.84	799
	R. Half-and-Half 2	201.99	±32.39	1146
	R. Half-and-Half 3	216.88	±37.91	1035

Tabela 7.4: Resultados dos conjuntos de experiências *PG_1000_DR_CM_LD17* e *PG_1000_DR_CM_LDTA* realizado durante a etapa da eliminação do *bloat*.

No conjunto de experiências em que o algoritmo de PG utiliza um limite dinâmico aplicado à profundidade das árvores (*PG_1000_DR_CM_LD17*) o crescimento do tamanho das árvores, ao longo das gerações, demonstrou-se semelhante ao apresentado nas experiências anterior, tal como é notório nos gráficos da Figura 7.4. Contudo, o limite tende a estagnar por volta dos 1000 nós, enquanto que no conjunto de experiências anterior tende a estagnar por volta dos 750.

O erro de previsão obtido para a experiência *PG_1000_DR_CM_LD17 R. Half-and-half*

2 ocorre devido à existência de um valor discrepante gerado pelo modelo para uma das previsões realizadas. Caso esse valor não existisse o erro de previsão seria de 40 litros, um valor semelhantes aos dos restantes modelos.

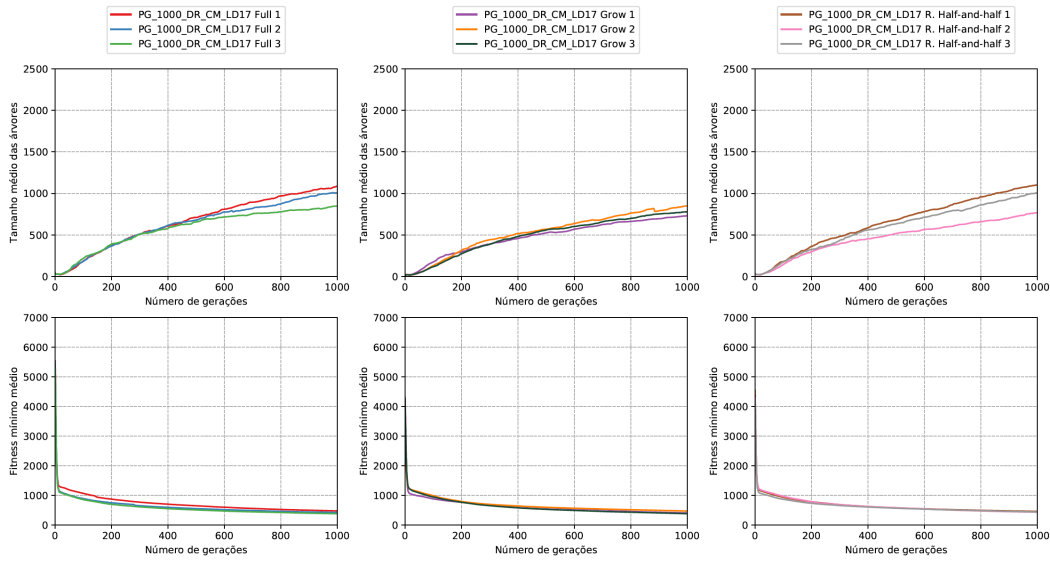


Figura 7.4: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_DR_CM_LD17* para os três métodos de inicialização da população, durante 1000 gerações.

As soluções encontradas com o conjunto de experiências que consideraram um limite dinâmico aplicado ao tamanho das árvores (*PG_1000_DR_CM_LD17A*) apresentam os valores de fitness, em média, mais baixo de todas as experiências realizadas durante esta etapa, ainda que nem sempre apresentem os erros de previsão mais baixos. Além disso, a aplicação deste tipo de limite tornou o crescimento das árvores exponencial, onde apenas uma solução é composta por menos de 1000 nós, porém é aquela com o maior valor de fitness para este conjunto de experiências (Figura 7.5).

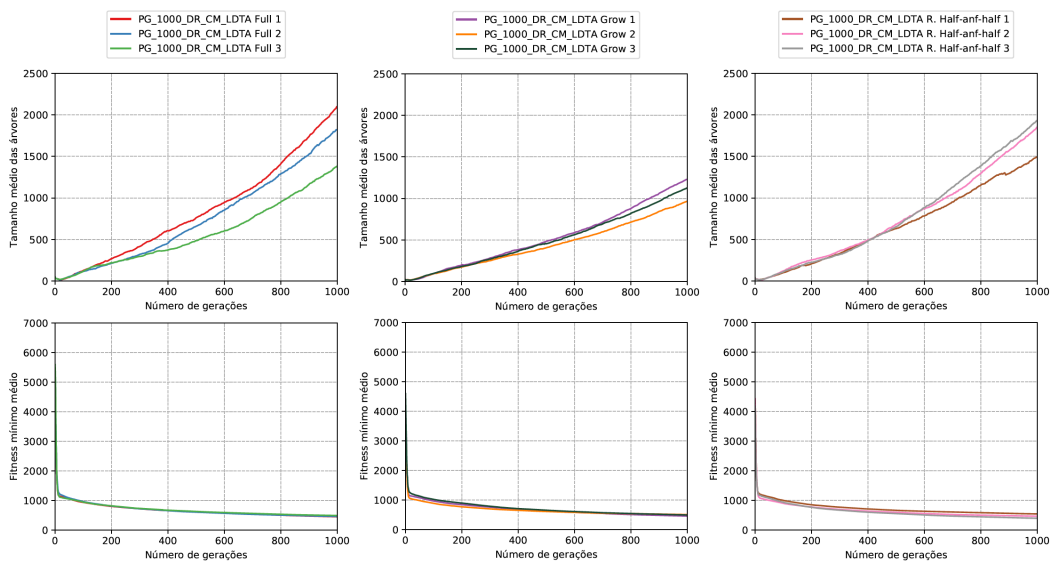


Figura 7.5: Fitness mínimo médio e tamanho médio das árvores do conjunto de experiências *PG_1000_DR_CM_LD17A* para os três métodos de inicialização da população, durante 1000 gerações.

Dos modelos gerados é de salientar o que foram resultantes das experiências *PG_1000_DR_CM_LDTA Full 1* e *PG_1000_DR_CM_LD17 Grow 2* por terem apresentado o melhor valor de fitness e o erro de previsão mais baixo, respetivamente.

7.4.2.2 Pontos-chave

Com base nos resultados obtidos é possível reter os seguintes pontos-chave:

- Os limites dinâmicos permitiram gerar soluções com valores de fitness mais baixos do que aqueles que foram atingidos na etapa anterior;
- No conjunto de experiências *PG_1000_DR_CM_LD17* o limite dinâmico, tal como em experiências anteriores, excluiu alguns indivíduos durante as gerações iniciais, mas acabou por atingir o valor máximo e não voltou a decrescer;
- O conjunto de experiências *PG_1000_DR_CM_LDTA* gerou as soluções com melhores resultados, entre eles o modelo resultante da experiência *PG_1000_DR_CM_LDTA Full 1*, com um valor de fitness de 103.88 e um erro de previsão de ± 35.93 litros.

7.4.3 Conclusões

Mais uma vez foi visível que as experiências realizadas com dados reais tendem a apresentar piores resultados do que quando são realizadas em aproximação a um conjunto de dados, para o qual se sabe que existe um modelo capaz de os mapear. Isto deve-se mais uma vez às características dos dados de treino não se demonstrarem fáceis de mapear, mesmo tendo sido incorporados os fatores meteorológicos na geração do modelo de previsão do consumo de combustível.

Os limites dinâmicos, ainda que não tenham atenuado o *bloat*, apresentaram resultados, em média, melhores que os apresentados pelas experiências em que estes não foram considerados, onde o limite aplicado ao tamanho das soluções foi aquele que gerou modelos com os melhores resultados, destacando o gerado pela experiência *PG_1000_DR_CM_LDTA Full 1*. Contudo, é necessário testar cada um deles perante um conjunto de dados diferentes dos utilizados para treino e teste, para comprovar a validade dos resultados obtidos.

7.5 2ª Fase B

Nesta fase, os modelos gerados na fase anterior foram aplicados a um conjunto de dados diferente daqueles que foram utilizados para treino e teste. O conjunto de dados utilizado foi o mesmo que foi utilizado na validação das experiências do capítulo 6, para que seja possível perceber a influência que os fatores meteorológicos têm na previsão do consumo de combustível, permitindo comparar os resultados desta fase com os resultados da fase de validação do capítulo anterior.

7.5.1 Validação dos modelos gerados

Na Tabela 7.5 encontram-se as médias dos erros absolutos, os desvios padrão e os números de nós que compõem cada uma das soluções encontradas. Os erros absolutos para os 28 períodos de abastecimentos encontram-se no Apêndice D.4.

Conjunto de experiências	Inicialização	Erro absoluto médio (%)	Desvio Padrão	Nº de nós da solução
<i>PG_1000_DR_CM</i>	Full 1	67.67%	44.23%	874
	Full 2	69.84%	33.72%	1165
	Full 3	79.44%	31.26%	955
	Grow 1	69.46%	33.85%	953
	Grow 2	75.65%	39.94%	734
	Grow 3	62.19%	41.21%	652
	R. Half-and-Half 1	68.90%	43.62%	944
	R. Half-and-Half 2	84.81%	37.09%	696
	R. Half-and-Half 3	68.99%	40.80%	717
<i>PG_1000_DR_CM_LD17</i>	Full 1	67.14%	37.05%	1131
	Full 2	90.42%	41.58%	867
	Full 3	75.25%	34.49%	878
	Grow 1	58.40%	34.00%	859
	Grow 2	71.13%	37.12%	4091
	Grow 3	77.16%	38.96%	2434
	R. Half-and-Half 1	77.10%	45.99%	813
	R. Half-and-Half 2	77.28%	40.82%	906
	R. Half-and-Half 3	76.64%	39.40%	3347
<i>PG_1000_DR_CM_LDTA</i>	Full 1	74.42%	44.89%	1774
	Full 2	76.75%	40.16%	1253
	Full 3	68.32%	41.60%	1194
	Grow 1	69.79%	36.31%	1041
	Grow 2	74.17%	41.45%	1457
	Grow 3	74.16%	42.28%	1561
	R. Half-and-Half 1	65.51%	38.90%	799
	R. Half-and-Half 2	67.14%	37.81%	1146
	R. Half-and-Half 3	85.49%	46.14%	1035
Modelo da empresa Stratio Automotive		72.00%	25.22%	-

Tabela 7.5: Resultados da validação dos modelos gerados pelos conjuntos de experiências: *PG_1000_DR_CM*, *PG_1000_DR_CM_LD17* e *PG_1000_DR_CM_LDTA*.

Para os períodos de abastecimento considerados, 12 dos 27 modelos apresentaram erros absolutos médios mais baixos do que o erro do modelo da empresa, o que corresponde a 44.44% dos modelos gerados.

O conjunto de experiências em que não foram utilizados limites dinâmicos (*PG_1000_DR_CM*) gerou o maior número de soluções com erros inferiores ao modelo de referência, onde 5 dos 12 modelos que apresentaram melhorias foram gerados nas experiências deste conjunto. Além disso, é aquele que apresenta os modelos compostos por um menor número de nós. Contudo, foi o conjunto *PG_1000_DR_CM_LD17* que gerou o modelo com o erro absoluto médio mais baixo. Este foi resultante da experiência *PG_1000_DR_CM_LD17 Grow 1* com o erro de 58.40%, o que corresponde a uma melhoria de aproximadamente 15% face ao modelo da empresa, porém apresenta um desvio padrão superior.

7.5.2 Conclusões

Com os dados reais e os fatores meteorológicos foi possível gerar vários modelos que se demonstraram melhores na previsão do consumo de combustível dos veículos pesa-

dos em comparação com o modelo da empresa, onde as experiências em que não foram utilizados limites dinâmicos geraram um maior número de modelos com erros absolutos, em média, mais baixos. Contudo, a melhor solução foi resultante da experiência *PG_1000_DR_CM_LD17 Grow 1*, onde um limite dinâmico foi conjugado com um limite estático de valor 17, apresentado um erro absoluto média de 58.40% o que corresponde a um erro de 93.28 litros para uma viagem de 100 quilómetros.

As soluções que apresentaram os melhores resultados em cada um dos conjuntos de experiências são aquelas que são constituídas por o número mais baixo de nós (soluções mais pequenas têm tendência a generalizar melhor [26]).

7.6 Conclusões gerais

Com a PG foi possível gerar modelos em que foram incorporados fatores meteorológicos que quando comparados aos modelos dos capítulos anteriores permitem determinar se esses fatores têm ou não influência no consumo de combustível. Na geração desses modelos foram utilizadas 21 variáveis, 9 referentes ao funcionamento do veículo e 12 referentes às condições meteorológicas. Mais uma vez, as variáveis consideradas são comuns a todos os veículos o que torna os modelos genéricos, podendo ser aplicados a diferentes veículos pesados.

As experiências em que foram incorporados fatores meteorológicos na geração de modelos baseados no modelo fornecido, foram as únicas que não apresentaram *bloat*. Ainda que o melhor modelo gerado em aproximação ao modelo fornecido e utilizando fatores meteorológicos tenha apresentado um erro absoluto médio mais baixo em comparação com o melhor modelo gerado pelas experiências do capítulo 5, com um erro de previsão inferior em aproximadamente 5 litros para uma viagem de 100 quilómetros, este apresenta um desvio superior. Assim sendo, é preferível o modelo resultante da experiência *PG_1000_VF_F_LD17 Full 3*, uma vez que é dependente de um número inferior de variáveis e é constituído por um menor número de nós.

Os limites dinâmicos, ainda que não tenham atenuado o *bloat*, geraram soluções com valores de fitness, em média, inferiores. Os erros apresentados na fase de validação podem estar associados ao facto de o conjunto de dados utilizado considerar períodos de abastecimento de duas viagens ou ao processo utilizado na recolha de dados.

Ainda que o melhor modelo gerado com base nos valores reais de consumo e utilizando fatores meteorológicos apresente melhores resultados ao nível do valor do erro absoluto médio e do desvio padrão em comparação com o melhor modelo gerado pelas experiências do capítulo 6, as melhorias não são muito significativas, uma vez que o modelo depende de mais do dobro das variáveis. Além disso, a diferente entre os erros de ambos é de 0.23 litros para uma viagem de 100 quilómetros.

Uma vez que não existe nenhum conjunto de dados sem valores em falta e que não tenha sido utilizado na geração dos modelos de todos os capítulos apresentados não foi possível comparar os modelos gerados com base no modelo fornecido e os gerados com base no valor real de consumo.

As pequenas melhorias obtidas durante os conjuntos de experiências presentes neste capítulo demonstram que os fatores meteorológicos influenciam o consumo de combustível, contudo as melhorias da sua incorporação nos modelos gerados não são substanciais, considerando que o número de variáveis utilizadas é o dobro.

Capítulo 8

Conclusão e trabalho futuro

Considerando os objetivos definidos no início desta dissertação, é possível afirmar que os mesmos foram cumpridos na sua totalidade. Assim sendo, os modelos gerados encontram-se prontos a ser entregues à Stratio Automotive para que os seus técnicos consigam analisá-los com mais detalhe e optar pela utilização de algum deles para prever o consumo de combustível dos veículos pesados dos seus clientes.

Ainda que a empresa procure um modelo que consiga prever com a máxima exatidão o consumo de combustível dos veículos pesados, uma vez que seria importante para a maioria dos seus clientes relacionados à indústria de transportes, no limite o fundamental é que o valor da previsão dado pelo modelo seja perto do valor real consumido para que a empresa consiga utilizá-lo para aquela que é a sua principal área de atuação, a previsão atempada de avarias. Assim sendo, um modelo que abranja um maior número de veículos, que não dependa de variáveis dispendiosas e que consiga determinar um valor próximo de consumo real de combustível pode ser utilizado para identificar avarias, pela comparação entre o valor real de consumo e o valor estimado pelo modelo.

Com base nos resultados obtidos, é possível afirmar que a técnica de PG mostrou-se capaz de gerar soluções eficazes quando aplicada a um problema real e complexo de regressão. Isto, à semelhança do que foi demonstrado nos trabalhos relacionados, reforça a sua elevada aplicabilidade, ainda mais dada a tendência atual para o crescimento de grandes volumes de dados (*Big Data*), dos quais é difícil extrair informações relevantes de forma manual ou mesmo com técnicas determinísticas de otimização. Assim sendo, esta tese permitiu mais uma vez reforçar a qualidade das soluções geradas com este tipo de técnicas.

De todo o trabalho desenvolvido, as conclusões-chaves são que com um número de variáveis inferiores à do modelo criado manualmente pela empresa Stratio Automotive, aproximadamente 25% das variáveis fornecidas, foi possível gerar pelo menos um modelo com um erro de previsão mais baixo comparativamente ao modelo inicial e que os fatores meteorológicos considerados, ainda que influenciem o consumo de combustível, as melhorias dos resultados obtidos não são substanciais considerando o aumento do número de variáveis. Por fim, de forma a tentar obter melhores resultados, seria interessante para trabalho futuro explorar dois caminhos. O primeiro seria colmatar a maioria das limitações apon-

tadas ao conjunto de dados. Para tal, era importante testar uma abordagem em que uma viagem seria dividida em várias mais pequenas em que os períodos de funcionamento sem movimento seriam eliminados, avaliando se o erro de excluir esse períodos seria maior ou menor do que os erros obtidos quando estes são considerados. Além disso, seria importante a definição do momento da última recolha ou a substituição dos valores de algumas variáveis por valores médios, com o objetivo de reduzir a variância nos valores de algumas variáveis para a última recolha. O segundo seria a implementação de uma abordagem semelhante à aplicada ao programa Janus Manager (JM) tal como se encontra descrita nos trabalhos relacionados. Para tal, ao fim do dia, durante o período de inatividade dos veículos, o modelo atual seria melhorado com a técnica de PG e com os dados recolhidos nesse mesmo dia.

Referências

- [1] Quercus diz que fabricantes automóveis manipulam dados de consumo de combustível. <http://www.jornaldenegocios.pt/empresas/automovel/detalhe/quercus-diz-que-fabricantes-automoveis-manipulam-dados-de-consumo-de-combustivel>. Accessed: 2017-12-23.
- [2] Riccardo Poli, Wb Langdon, and Nf McPhee. *A field guide to genetic programming (With contributions by JR Koza)(2008)*. Number March. 2008.
- [3] Saemundur O. Haraldsson, John R. Woodward, Alexander E.I. Brownlee, and Kristin Siggeirsdottir. Fixing Bugs in Your Sleep: How Genetic Improvement Became an Overnight Success. *Proceedings of the 2017 Conference Companion on Genetic and Evolutionary Computation Companion*, pages 1513–1520, 2017.
- [4] Claire Le Goues, Thanh Vu Nguyen, Stephanie Forrest, and Westley Weimer. GenProg: A generic method for automatic software repair. *IEEE Transactions on Software Engineering*, 38(1):54–72, 2012.
- [5] Coeficientes de correlação. <http://www.abgconsultoria.com.br/blog/coeficientes-de-correlacao/>, 2017. Accessed: 2018-05-05.
- [6] Joe Stam. Stereo imaging with cuda. *OpenVIDIA, january*, (January), 2008.
- [7] Erik Hellström, Jan Åslund, and Lars Nielsen. Design of an efficient algorithm for fuel-optimal look-ahead control. *Control Engineering Practice*, 18(11):1318–1327, 2010.
- [8] A.E. Eiben and James E Smith. *Introduction to Evolutionary Computing*. Springer-Verlag Berlin Heidelberg, 2003.
- [9] Charles Darwin. On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life. *London*, 1:859, 1859.
- [10] Zbigniew Michalewicz. Evolution strategies and other methods. In *Genetic algorithms + data structures= evolution programs*, pages 159–177. Springer, 1996.
- [11] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, 3(2):82–102, 1999.
- [12] John Henry Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

-
- [13] Francesco Archetti, Stefano Lanzeni, Enza Messina, and Leonardo Vanneschi. Genetic programming for human oral bioavailability of drugs. *{GECCO 2006:} Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 1(April 2016):255–262, 2006.
- [14] Francesco Archetti, Stefano Lanzeni, Enza Messina, and Leonardo Vanneschi. Genetic programming for computational pharmacokinetics in drug discovery and development. *Genetic Programming and Evolvable Machines*, 8(4):413–432, 2007.
- [15] W B Langdon and S J Barrett. Genetic Programming in Data Mining for Drug Discovery. *Evolutionary Computing in Data Mining*, 163:211–235, 2004.
- [16] Adnan Parlak, Yasar Islamoglu, Halit Yasar, and Aysun Egrisogut. Application of artificial neural network to predict specific fuel consumption and exhaust temperature for a diesel engine. *Applied Thermal Engineering*, 26(8):824–828, 2006.
- [17] Jeroen Eggermont, Joost N. Kok, and Walter A. Kusters. Genetic programming for data classification: Partitioning the search space. In *Proceedings of the 2004 ACM Symposium on Applied Computing, SAC '04*, pages 1001–1005, New York, NY, USA, 2004. ACM.
- [18] Michael O’Neill, Miguel Nicolau, and Alexandros Agapitos. Experiments in program synthesis with grammatical evolution: A focus on Integer Sorting. *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pages 1504–1511, 2014.
- [19] Stefan Forstenlechner B, David Fagan, Miguel Nicolau, and Michael O Neill. A Grammar Design Pattern for Arbitrary Program Synthesis Problems in Genetic Programming. 10196:262–277, 2017.
- [20] Mark Harman, Yue Jia, and William B. Langdon. Babel Pidgin: SBSE Can Grow and Graft Entirely New Functionality into a Real World System. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8636 LNCS:247–252, 2014.
- [21] W. B. Langdon and M. Harman. Genetically improved CUDA C++ software. *Genetic Programming*, pages 87–99, 2014.
- [22] William B Langdon and Mark Harman. Optimising Existing Software with Genetic Programming. 10196(1):118–135, 2017.
- [23] Westley Weimer, Stephanie Forrest, Claire Le Goues, and ThanhVu Nguyen. Automatic program repair with evolutionary computation. *Communications of the ACM*, 53:109, 2010.
- [24] *The Optimization Problem*, pages 1–26. Springer US, Boston, MA, 2007.
- [25] Transport & Environment. Mind the GAP 2016. Technical report, 2016.

-
- [26] Sara Silva. *Controlling bloat: individual and population based approaches in genetic programming*. PhD thesis, 2008.
- [27] Jason Brownlee. *Master Machine Learning Algorithms: discover how they work and implement them from scratch*. Jason Brownlee, 2016.
- [28] Mark Harman, William B. Langdon, Yue Jia, David R. White, Andrea Arcuri, and John A. Clark. The GISMOE challenge: constructing the pareto program surface using genetic programming to find better programs (keynote paper). *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE'12)*, pages 1–14, 2012.
- [29] David R. White, Andrea Arcuri, and John A. Clark. Evolutionary improvement of programs. *IEEE Transactions on Evolutionary Computation*, 15(4):515–538, 2011.
- [30] Claire Le Goues, Michael Dewey-Vogt, Stephanie Forrest, and Westley Weimer. A systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each BT - 34th International Conference on Software Engineering, ICSE 2012, June 2, 2012 - June 9, 2012. pages 3–13, 2012.
- [31] Andrea Arcuri and Xin Yao. A novel co-evolutionary approach to automatic software bug fixing. *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pages 162–168, 2008.
- [32] Mark Harman, Yue Jia, William B Langdon, Justyna Petke, Iman Hemati Moghadam, Shin Yoo, and Fan Wu. Genetic Improvement for Adaptive Software Engineering. pages 1–4, 2014.
- [33] Steven Gustafson, Anikó Ekárt, Edmund Burke, and Graham Kendall. Problem difficulty and code growth in genetic programming. *Genetic Programming and Evolvable Machines*, 5(3):271–290, 2004.
- [34] Amir H. Gandomi, Amir H. Alavi, and Conor Ryan, editors. *Handbook of Genetic Programming Applications*. Springer International Publishing, 2015.
- [35] W. B. Langdon. A many threaded CUDA interpreter for genetic programming. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6021 LNCS:146–158, 2010.
- [36] Thomas Ackling, Brad Alexander, and Ian Grunert. Evolving Patches for Software Repair. *GECCO'11, 13th annual conference on Genetic and evolutionary computation*, pages 1427–1434, 2011.
- [37] 1000 Genomes Project Consortium et al. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073, 2010.
- [38] Anthony Brabazon, Michael O’Neill, and Seán McGarraghy. *Natural Computing Algorithms*. Springer-Verlag Berlin Heidelberg, 2015.

-
- [39] Robert I. McKay, Nguyen Xuan Hoai, Peter Alexander Whigham, Yin Shan, and Michael O’neill. Grammar-based Genetic programming: A survey. *Genetic Programming and Evolvable Machines*, 11(3-4):365–396, 2010.
- [40] Thomas Helmuth, Lee Spector, and James Matheson. Solving Uncompromising Problems With Lexicase Selection. 19(5):630–643, 2015.
- [41] Thomas Helmuth and Lee Spector. General Program Synthesis Benchmark Suite. *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO ’15*, pages 1039–1046, 2015.
- [42] Valerio Turri, Bart Besselink, and Karl H Johansson. Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning. *IEEE Transactions on Control Systems Technology*, 25(1):12–28, 2017.
- [43] Sina Torabi and Mattias Wahde. Fuel consumption optimization of heavy-duty vehicles using genetic algorithms. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 29–36. IEEE, 2017.
- [44] Luca Caltagirone, Sina Torabi, and Mattias Wahde. Truck platooning based on lead vehicle speed profile optimization and artificial physics. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, pages 394–399. IEEE, 2015.
- [45] Chaozhe R He, Helmut Maurer, and Gábor Orosz. Fuel consumption optimization of heavy-duty vehicles with grade, wind, and traffic information. *Journal of Computational and Nonlinear Dynamics*, 11(6):061011, 2016.
- [46] Necla Togun and Sedat Baysec. Genetic programming approach to predict torque and brake specific fuel consumption of a gasoline engine. *Applied Energy*, 87(11):3401–3408, 2010.
- [47] Necla Kara Togun and Sedat Baysec. Prediction of torque and specific fuel consumption of a gasoline engine by using artificial neural networks. *Applied Energy*, 87(1):349–355, 2010.
- [48] Franz X Moser, Theodor Sams, and Wolfgang Cartellieri. Impact of future exhaust gas emission legislation on the heavy duty truck engine. Technical report, SAE Technical Paper, 2001.
- [49] Jos M Alonso, Fernando Alvarruiz, Jos M Desantes, Leonor Hernández, Vicente Hernández, and Germn Molto. Combining neural networks and genetic algorithms to predict and reduce diesel engine emissions. *IEEE transactions on evolutionary computation*, 11(1):46–55, 2007.

Apêndices

Esta página foi propositadamente deixada em branco.

Apêndice A

Análise correlacional

Na Tabela A.1 encontra-se a totalidade dos resultados obtidos na análise correlacional efetuada ao conjunto de variáveis fornecidas. Cada número entre parênteses corresponde a uma variável com base na legenda seguinte:

Legenda: (1) - tempo (s); (2) - aceleração (m/s^2); (3) - relação de transmissão; (4) - distância (m); (4) - altitude (m); (6) - altura (m); (7) - α ($^\circ$); (8) - raio (m); (9) - velocidade de derrapagem (km/h); (10) - μ ; (11) - velocidade média (m/s); (12) - velocidade média (km/h); (13) - C_x ; (14) - FR (N); (15) - FCTP (N); (16) - FD(N); (17) - FTCD (N); (18) - FP(N); (19) - E(J); (20) - rendimento de transmissão; (21) - Tb (N.m); (22) - número de rotações (rpm); (23) - $\frac{S_p}{S_p(bsfc_{min})}$; (24) - WFNCR (J); (25) - Wb (J); (26) - $\frac{bmep(bsfc_{min})}{bmep_{max}}$; (27) - $\frac{bmep(bsfc_{min})}{bmep}$; (28) - a; (29) - b; (30) - c; (31) - $\frac{bmep(bsfc_{min})}{bmep}$; (32) - $\frac{bsfc}{bsfc_{min}}$; (33) - bsfc (g/kW.h); (34) - mf[$V \geq V_{min1a}$]; (35) - mf[$V < V_{min1a}$]; (36) - mftotal (g); (37) - mfac (g); (38) - dac (m).

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
(1)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(2)	0.06	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(3)	-0.07	0.07	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(4)	0.99	0.03	-0.06	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(5)	-0.46	-0.43	0.06	-0.44	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(6)	-0.46	-0.43	0.06	-0.44	1.00	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(7)	0.05	0.02	0.13	0.05	0.07	0.07	-	-	-	-	-	-	-	-	-	-	-	-	-
(8)	-0.02	-0.01	-0.10	-0.02	-0.02	-0.02	0.03	-	-	-	-	-	-	-	-	-	-	-	-
(9)	0.05	-0.03	-0.15	0.04	-0.06	-0.06	0.08	0.84	-	-	-	-	-	-	-	-	-	-	-
(10)	-0.05	-0.12	0.12	-0.04	0.08	0.08	0.09	-0.12	-0.26	-	-	-	-	-	-	-	-	-	-
(11)	0.15	-0.17	-0.16	0.14	-0.09	-0.09	0.20	0.13	0.40	0.18	-	-	-	-	-	-	-	-	-
(12)	0.15	-0.17	-0.16	0.14	-0.09	-0.09	0.20	0.13	0.40	0.18	1.00	-	-	-	-	-	-	-	-
(13)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-	-	-	-	-	-	-
(14)	0.04	-0.04	-0.22	0.04	-0.04	-0.04	-0.70	-0.03	-0.11	-0.10	-0.24	-0.24	0.00	-	-	-	-	-	-
(15)	-0.07	-0.01	-0.11	-0.07	-0.07	-0.07	-0.99	-0.03	-0.08	-0.09	-0.20	-0.20	0.00	0.63	-	-	-	-	-
(16)	-0.18	0.17	0.27	-0.17	0.13	0.13	-0.15	-0.16	-0.41	-0.12	-0.95	-0.95	0.00	0.16	0.15	-	-	-	-
(17)	0.03	0.11	-0.01	0.02	-0.08	-0.08	-0.05	0.05	0.13	-0.87	-0.12	-0.12	0.00	0.04	0.06	0.10	-	-	-
(18)	0.01	0.01	0.12	0.01	0.07	0.07	0.74	0.03	0.10	0.13	0.24	0.24	0.00	-0.75	-0.75	-0.18	-0.12	-	-
(19)	-0.44	-0.46	0.02	-0.41	0.99	0.99	0.10	0.01	0.01	0.11	0.07	0.07	0.00	-0.07	-0.09	-0.04	-0.09	0.10	-
(20)	0.12	-0.16	-0.32	0.12	-0.07	-0.07	0.13	0.11	0.35	0.12	0.81	0.81	0.00	-0.17	-0.13	-0.71	-0.09	0.17	0.05
(21)	0.10	-0.06	-0.15	0.10	0.01	0.01	0.18	0.03	0.07	0.28	0.26	0.26	0.00	-0.02	-0.21	-0.28	-0.40	0.30	0.06
(22)	0.12	-0.12	0.13	0.11	-0.07	-0.07	0.22	0.08	0.31	0.24	0.87	0.87	0.00	-0.28	-0.22	-0.77	-0.14	0.26	0.06
(23)	0.12	-0.12	0.13	0.11	-0.07	-0.07	0.22	0.08	0.31	0.24	0.87	0.87	0.00	-0.28	-0.22	-0.77	-0.14	0.26	0.06
(24)	-0.03	0.10	0.08	-0.02	0.02	0.02	-0.09	-0.07	-0.07	-0.71	-0.36	-0.36	0.00	0.08	0.10	0.34	0.88	-0.16	-0.04
(25)	0.07	-0.08	-0.09	0.07	-0.04	-0.04	0.15	0.20	0.27	0.27	0.46	0.46	0.00	-0.10	-0.16	-0.46	-0.39	0.26	0.03
(26)	-0.04	0.06	-0.50	-0.04	-0.02	-0.02	-0.24	0.00	-0.17	-0.27	-0.64	-0.64	0.00	0.37	0.23	0.42	0.12	-0.29	-0.09
(27)	-0.07	-0.01	-0.08	-0.08	0.01	0.01	-0.24	-0.13	-0.20	-0.10	-0.33	-0.33	0.00	0.13	0.27	0.28	0.09	-0.27	-0.04
(28)	0.02	0.03	-0.50	0.02	-0.07	-0.07	-0.21	-0.03	-0.13	-0.24	-0.43	-0.43	0.00	0.33	0.20	0.15	0.09	-0.25	-0.10
(29)	0.05	-0.08	0.31	0.05	0.00	0.00	0.24	0.14	0.29	0.27	0.73	0.73	0.00	-0.33	-0.23	-0.52	-0.13	0.28	0.09
(30)	-0.07	0.09	-0.29	-0.06	0.01	0.01	-0.24	-0.16	-0.31	-0.27	-0.77	-0.77	0.00	0.33	0.24	0.58	0.13	-0.28	-0.09

(31)	-0.07	-0.01	-0.08	-0.08	0.01	0.01	-0.24	-0.13	-0.20	-0.10	-0.33	-0.33	0.00	0.13	0.27	0.28	0.09	-0.27	-0.04
(32)	0.10	0.04	0.16	0.10	-0.05	-0.05	0.27	0.13	0.20	0.08	0.31	0.31	0.00	-0.15	-0.30	-0.25	-0.07	0.29	0.00
(33)	0.10	0.04	0.16	0.10	-0.05	-0.05	0.27	0.13	0.20	0.08	0.31	0.31	0.00	-0.15	-0.30	-0.25	-0.07	0.29	0.00
(34)	0.08	-0.07	-0.09	0.07	-0.05	-0.05	0.15	0.21	0.28	0.26	0.46	0.46	0.00	-0.10	-0.16	-0.46	-0.37	0.27	0.03
(35)	0.01	0.09	-0.28	0.00	-0.07	-0.07	0.01	-0.05	-0.08	-0.13	-0.29	-0.29	0.00	-0.02	-0.01	0.19	0.06	0.00	-0.10
(36)	0.08	-0.02	-0.23	0.07	-0.08	-0.08	0.15	0.17	0.22	0.17	0.28	0.28	0.00	-0.11	-0.16	-0.33	-0.32	0.25	-0.03
(37)	0.99	0.01	-0.07	0.99	-0.38	-0.38	0.05	-0.02	0.04	-0.04	0.15	0.15	0.00	0.04	-0.06	-0.17	0.02	0.01	-0.36
(38)	0.99	0.03	-0.06	1.00	-0.44	-0.44	0.05	-0.02	0.04	-0.04	0.14	0.14	0.00	0.04	-0.07	-0.17	0.02	0.01	-0.41
	(20)	(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)	(31)	(32)	(33)	(34)	(35)	(36)	(37)	(38)
(21)	0.19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(22)	0.59	0.21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(23)	0.59	0.21	1.00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(24)	-0.29	-0.43	-0.31	-0.31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(25)	0.35	0.77	0.40	0.40	-0.52	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(26)	-0.48	-0.07	-0.78	-0.78	0.20	-0.25	-	-	-	-	-	-	-	-	-	-	-	-	-
(27)	-0.26	-0.55	-0.34	-0.34	0.16	-0.54	0.32	-	-	-	-	-	-	-	-	-	-	-	-
(28)	-0.39	-0.01	-0.57	-0.57	0.12	-0.14	0.93	0.26	-	-	-	-	-	-	-	-	-	-	-
(29)	0.57	0.13	0.89	0.89	-0.24	0.30	-0.94	-0.34	-0.87	-	-	-	-	-	-	-	-	-	-
(30)	-0.58	-0.15	-0.92	-0.92	0.26	-0.32	0.93	0.34	0.83	-0.99	-	-	-	-	-	-	-	-	-
(31)	-0.26	-0.55	-0.34	-0.34	0.16	-0.54	0.32	0.99	0.26	-0.34	0.34	-	-	-	-	-	-	-	-
(32)	0.22	0.56	0.35	0.35	-0.16	0.52	-0.35	-0.84	-0.29	0.36	-0.36	-0.84	-	-	-	-	-	-	-
(33)	0.22	0.56	0.35	0.35	-0.16	0.52	-0.35	-0.84	-0.29	0.36	-0.36	-0.84	1.00	-	-	-	-	-	-
(34)	0.35	0.77	0.40	0.40	-0.52	0.99	-0.26	-0.56	-0.15	0.31	-0.33	-0.56	0.56	0.56	-	-	-	-	-
(35)	-0.20	-0.02	-0.32	-0.32	0.06	-0.14	0.44	0.23	0.40	-0.39	0.39	0.23	-0.23	-0.23	-0.15	-	-	-	-
(36)	0.23	0.71	0.21	0.21	-0.46	0.86	-0.01	-0.41	0.07	0.09	-0.11	-0.41	0.40	0.40	0.86	0.38	-	-	-
(37)	0.12	0.10	0.12	0.12	-0.02	0.07	-0.04	-0.07	0.01	0.06	-0.07	-0.07	0.09	0.09	0.07	0.00	0.06	-	-
(38)	0.12	0.10	0.11	0.11	-0.02	0.07	-0.04	-0.08	0.02	0.05	-0.06	-0.08	0.10	0.10	0.07	0.00	0.07	0.99	-

Tabela A.1: Tabela dos resultados completos da análise correlacional

Esta página foi propositadamente deixada em branco.

Apêndice B

Resultados das experiências realizadas

Nas tabelas seguintes encontram-se os resultados para as experiências realizadas com PG. Na Tabela B.1 encontram-se os resultados das experiências realizadas com o 1º conjunto de dados, baseadas no modelo fornecido. Na Tabela B.2 estão os resultados das experiências realizadas com o 2º conjunto de dados, baseadas nos dados reais. Por fim, na Tabela B.3 estão os resultados das experiências realizadas com o 2º conjunto de dados, baseadas no modelo fornecido pela empresa.

B.1 Resultados das experiências realizadas com o 1º conjunto de dados baseadas no modelo fornecido

Conjunto de experiências	Experiência	Fitness	Erro modelo (%)	Erro real (%)	Erro previsão (L/100km)	Nº de nós da solução
PG_500	Full	2.98	0.82%	5.90%	-	845
	Grow	5.08	6.49%	0.61%	-	391
	R. Half-and-half	3.06	0.20%	6.86%	-	362
PG_1000	Full	1.09	1.65%	5.13%	-	665
	Grow	2.74	2.74%	4.11%	-	606
	R. Half-and-half	3.35	4.79%	2.20%	-	586
PG_1000_NU	Full	2.91	0.76%	5.96%	-	727
	Grow	1.95	0.39%	7.03%	-	961
	R. Half-and-half	2.30	2.38%	4.45%	-	726
PG_1000_CORR	Full	3.94	1.28%	7.86%	-	663
	Grow	5.82	14.99%	7.32%	-	1002
	R. Half-and-half	2.28	4.20%	2.75%	-	525
PG_1000_NU_CORR	Full	3.31	4.97%	2.03%	-	677
	Grow	2.29	2.50%	9.01%	-	943
	R. Half-and-half	4.57	2.48%	4.35%	-	420
PG_1000_CORR_WITH_DAC	Full	1.80	0.13%	6.54%	-	723
	R. Half-and-half	5.35	3.82%	3.10%	-	773
PG_1000_NU_CORR_WITH_DAC	Full	4.26	2.66%	2.31%	-	769
	R. Half-and-half	3.03	2.31%	4.51%	-	891
PG_1000_E2 (PG_500)	Full 1	2.18	4.43%	2.53%	1.21	327
	Full 2	5.10	1.96%	4.84%	2.01	243
	Full 3	4.36	6.44%	0.66%	1.66	445
	R. Half-and-half 1	6.07	2.48%	8.98%	2.17	335
	R. Half-and-half 2	2.88	2.74%	4.11%	1.32	380
	R. Half-and-half 3	3.81	2.17%	8.70%	1.85	390
PG_1000_E2 (PG_1000)	Full 1	1.83	2.60%	9.10%	1.60	746

PG_1000_E2 (PG_1000)	Full 2	1.28	1.75%	5.03%	1.26	597
	Full 3	1.28	5.99%	12.26%	1.28	1027
	Full 1	1.57	2.90%	9.38%	1.62	821
PG_1000_E2 (PG_1000_NU)	Full 2	2.93	2.29%	8.81%	1.57	709
	Full 3	1.94	1.39%	7.96%	1.41	967
	Grow 1	0.98	2.04%	4.77%	1.14	609
	Grow 2	2.16	1.49%	5.28%	1.52	561
	Grow 3	1.14	2.60%	4.24%	0.91	453
PG_1000_E2 (PG_1000_CORR)	Full 1	6.51	0.26%	6.43%	2.24	900
	Full 2	3.78	6.57%	0.54%	1.65	671
	Full 3	1.73	0.43%	7.07%	1.73	1240
PG_1000_E2 (PG_1000_CORR_WITH_DAC)	Full 1	3.81	8.89%	1.63%	1.94	755
	Full 2	4.12	1.66%	8.22%	1.65	855
	Full 3	2.31	4.79%	11.14%	1.36	980
PG_1000_VF	Full 1	7.37	1.80%	8.35%	2.55	658
	Full 2	5.79	7.34%	13.52%	2.54	742
	Full 3	7.13	14.03%	19.76%	2.94	1012
	Grow 1	6.88	10.45%	16.42%	2.32	923
	Grow 2	6.80	1.32%	7.90%	2.57	999
	Grow 3	6.32	1.17%	7.76%	1.90	682
	R. Half-and-half 1	6.30	2.35%	4.48%	2.45	737
	R. Half-and-half 2	6.84	7.66%	13.82%	2.29	832
	R. Half-and-half 3	5.54	3.42%	9.86%	2.13	727
PG_1000_VF_F	Full 1	1.89	0.73%	7.35%	1.38	1180
	Full 2	3.12	2.06%	4.74%	1.44	585
	Full 3	2.23	0.57%	6.14%	1.70	574
	Grow 1	1.57	2.78%	9.26%	1.49	604
	Grow 2	2.67	1.91%	8.45%	1.78	907
	Grow 3	1.79	2.64%	4.20%	1.06	1018
	R. Half-and-half 1	2.85	5.07%	11.40%	2.51	624
	R. Half-and-half 2	2.14	0.34%	6.35%	1.15	1070
	R. Half-and-half 3	2.88	0.20%	6.49%	1.33	875

PG_1000_VF_F_LD17 (sem truncação 0-100)	Full 1	2.49	0.67%	6.04%	1.57	703
	Full 2	2.22	0.44%	7.08%	1.69	1199
	Full 3	1.83	6.96%	0.17%	1.26	886
	Grow 1	1.77	0.22%	6.87%	2.50	672
	Grow 2	2.04	1.51%	5.26%	1.24	904
	Grow 3	2.93	4.37%	2.59%	2.35	494
	R. Half-and-half 1	2.97	0.76%	5.96%	1.62	1003
	R. Half-and-half 2	2.76	0.51%	0.51%	2.13	1085
	R. Half-and-half 3	2.90	48.06%	51.53%	2.00	358
PG_1000_VF_F_LD17	Full 1	2.23	1.74%	5.04%	1.64	637
	Full 2	2.76	0.01%	6.66%	1.67	735
	Full 3	2.98	0.37%	7.01%	1.46	552
	Grow 1	2.60	3.35%	3.54%	2.15	634
	Grow 2	1.69	0.46%	6.24%	8.44	963
	Grow 3	2.51	0.01%	6.66%	1.38	569
	R. Half-and-half 1	1.07	1.31%	7.90%	1.33	779
	R. Half-and-half 2	2.03	1.68%	8.23%	1.72	461
	R. Half-and-half 3	2.20	3.51%	3.39%	1.43	1030
PG_1000_VF_F_LD30	Grow 1	1.24	1.34%	5.42%	2.33	1517
	Grow 2	1.98	4.70%	2.28%	1.75	2362
	Grow 3	2.09	1.97%	8.51%	1.34	1150
PG_1000_VF_F_LD35	Grow 1	1.91	0.46%	6.24%	1.98	1983
PG_1000_VF_F_LDTA	Full 1	1.88	2.77%	4.09%	1.39	706
	Full 2	1.46	1.78%	5.01%	1.08	701
	Full 3	2.21	1.13%	5.62%	4.00	593
	Grow 1	1.90	0.90%	5.83%	1.48	689
	Grow 2	2.82	3.32%	9.77%	2.08	558
	Grow 3	2.64	6.75%	12.97%	3.27	596
	R. Half-and-half 1	1.29	2.42%	4.41%	1.79	662
R. Half-and-half 2	2.11	4.38%	2.58%	1.19	402	
R. Half-and-half 3	2.76	8.10%	0.89%	1.23	388	
PG_1000_VF_F	Full 1	6.85	2.49%	4.35%	2.40	729

B.2 Resultados das experiências realizadas com o 2º conjunto de dados baseadas nos dados reais

PG_1000_VF_F	Full 2	3.49	0.16%	6.82%	1.91	871
	Full 3	4.32	1.60%	8.16%	1.87	715
	Full 1	0.88	0.18%	6.51%	0.79	572
PG_1000_VF_F_CM	Full 2	0.53	0.59%	6.12%	0.59	804
	Full 3	0.48	0.12%	6.78%	1.42	991
	Grow 1	0.49	1.53%	8.09%	0.85	726
	Grow 2	0.54	0.40%	7.04%	0.68	705
	Grow 3	0.60	0.25%	6.90%	1.29	438
	R. Half-and-half 1	0.52	1.68%	8.23%	0.89	679
	R. Half-and-half 2	0.47	0.03%	6.70%	0.69	479
	R. Half-and-half 3	0.67	2.05%	8.59%	2.26	670

Tabela B.1: Resultados das experiências realizadas com o 1º conjunto de dado baseadas no modelo fornecido.

Conjunto de experiências	Experiência	Fitness	Erro previsão (Litros)	Nº de nós da solução
PG_1000_DR	Full 1	284.35	36.70	1129
	Full 2	307.59	25.50	1128
	Full 3	273.49	44.99	816
	Grow 1	317.06	96.79	1264
	Grow 2	210.92	32.02	837
	Grow 3	249.94	30.18	846
	R. Half-and-half 1	249.57	84.07	1232
	R. Half-and-half 2	312.40	268.94	2045
	R. Half-and-half 3	274.23	33.61	1314
PG_1000_DR_LD17	Full 1	249.02	35.22	888
	Full 2	272.66	27.45	989
	Full 3	232.91	38.81	1382
	Grow 1	202.56	67.76	1063

	Grow 2	155.57	29.90	3120
	Grow 3	159.90	33.55	1488
PG_1000_DR_LD17	R. Half-and-half 1	199.91	33.87	2001
	R. Half-and-half 2	257.24	87.38	1146
	R. Half-and-half 3	104.30	30.02	2000
	Full 1	295.45	27.19	3395
	Full 2	91.73	32.88	2689
	Full 3	117.11	33.49	3530
	Grow 1	164.45	35.25	869
	Grow 2	222.04	26.59	2339
PG_1000_DR_LDTA	Grow 2	222.04	26.59	2339
	Grow 3	242.23	39.37	2133
	R. Half-and-half 1	176.71	29.90	3003
	R. Half-and-half 2	184.88	38.36	1869
	R. Half-and-half 3	262.32	26.60	2738
	Full 1	347.75	29.51	874
	Full 2	231.65	55.42	1165
	Full 3	268.48	34.10	955
	Grow 1	198.48	34.25	953
PG_1000_DR_CM	Grow 2	325.36	51.77	734
	Grow 3	314.57	41.02	652
	R. Half-and-half 1	252.82	34.30	944
	R. Half-and-half 2	225.38	32.21	696
	R. Half-and-half 3	218.39	40.74	717
	Full 1	282.59	43.47	1131
	Full 2	244.71	34.42	867
	Full 3	215.41	38.25	878
	Grow 1	207.45	36.94	859
PG_1000_DR_CM_LD17	Grow 2	223.65	27.42	4091
	Grow 3	172.81	30.28	2434
	R. Half-and-half 1	277.92	37.96	813
	R. Half-and-half 2	294.27	5138.94	906

PG_1000_DR_CM_LD17	R. Half-and-half 3	108.05	33.64	3347
PG_1000_DR_CM_LD17	Full 1	103.88	35.93	1774
	Full 2	240.38	55.68	1253
	Full 3	146.20	33.36	1194
PG_1000_DR_CM_LD17	Grow 1	225.72	28.14	1041
	Grow 2	172.01	39.74	1457
	Grow 3	237.51	31.48	1561
	R. Half-and-half 1	245.04	34.84	799
	R. Half-and-half 2	201.99	32.39	1146
	R. Half-and-half 3	216.88	37.91	1035

Tabela B.2: Resultados das experiências realizadas com o 2º conjunto de dados baseadas nos dados reais.

B.3 Resultados das experiências realizadas com o 2º conjunto de dados baseadas no modelo fornecido

Conjunto de experiências	Experiência	Fitness	Erro previsão (Litros)	Nº de nós da solução
PG_1000_VF_F_DL17	Full 1	13.37	42.68	1185
	Full 2	12.27	10.90	1072
	Full 3	8.73	52.32	1286
	Grow 1	17.45	8.64	1249
	Grow 2	11.93	13.51	1047
	Grow 3	20.64	35.76	761
PG_1000_VF_F_DL30	Full 1	11.72	11.01	3792
	Full 2	9.64	12.80	2053
	Full 3	8.56	12.43	3604
PG_1000_VF_F_DLTA	Full 1	12.05	10.94	674
	Full 2	18.71	9.50	867
	Full 3	11.98	13.80	701

PG_1000_VF_F_DLTA	Grow 1	7.75	12.10	684
	Grow 2	8.74	16.57	506
	Grow 3	9.21	19.69	571
PG_1000_VF_F_CM_DL17	Full 1	7.09	50.70	946
	Full 2	7.49	26.73	1127
	Full 3	8.86	4.89	974
PG_1000_VF_F_CM_DL30	Grow 1	7.75	12.10	880
	Grow 2	8.74	16.57	684
	Grow 3	9.21	19.69	775
PG_1000_VF_F_CM_DLTA	Full 1	6.62	8.82	2008
	Full 2	7.39	13.71	2250
	Full 3	6.52	16.73	1915
PG_1000_VF_F_CM_DLTA	Full 1	8.86	20.97	508
	Full 2	21.64	10.79	2578
	Full 3	9.32	12.72	1345
PG_1000_VF_F_CM_DLTA	Grow 1	15.53	12.84	515
	Grow 2	19.57	13.40	557
	Grow 3	16.38	17.73	499

Tabela B.3: Resultados das experiências realizadas com o 2º conjunto de dados baseadas no modelo fornecido.

Apêndice C

Expressão resultante da experiência PG_1000_NU Grow 1

$$\begin{aligned} & 2 * altitude - 2 * altura - bmep_{bmep_{max}} * vel_{derrapagem} / ((cos(bsfc_{min} + fr) + \\ & (distancia - t) / fr) * (2 * bsfc + bsfc_{min} - cx * mfac - fd + fr - mftotal * veloc_{media} * \\ & (bsfc_{min} + cx) + tb / vel_{derrapagem} + cos(bmep_{bmep_{max}}) - (bsfc - tb) / vel_{derrapagem} \\ & - raio / mftotal + wfncr / fd) + cos(cos(acceleracao * fp / cos(bmep_{bmep_{max}}))) * \\ & cos(bmep_{bmep_{max}} + c + fp))) + 3 * bsfc_{min} + 2 * cos(bmep_{bmep_{max}}) - 4 * cos(mfac / fr) \\ & - cos((distancia - t) / fr) - cos((distancia - fd - n_{motor}) / fr) - cos(mfac / n_{motor}) - cos(((- altitude \\ & + dac) * cos(bmep_{bmep_{max}}) + (acceleracao * (veloc_{media} + wb + mfac / cx) * (-2 * b + \\ & bmep_{menor_{zero}} + distancia - t - vel_{derrapagem} * (rendtrans + wb) / tb) / (c - \\ & (-vel_{derrapagem} / fr + (fd + n_{motor}) / altura) / (-fr + mfac + vel_{derrapagem} * (rendtrans \\ & + wb) / tb) + (-altitude + c + fd + raio - tb) / cos(cos(bsfc_{bsfc_{bsfc_{min}}})) + altura + bsfc \\ & + fctp / veloc_{media} + 2 * fr + mfac / (rendtrans * (fr + cos(mfac / fr)) * (a + bsfc_{min} * cos(altura) / t \\ & - dac + mu + raio + cos(cos(tb) / (-bmep_{menor_{zero}} + t))) * (altitude + bmep_{bmep_{max}} - c / t - cx \\ & + mfac + mu) * (bmep_{bmep_{max}} + bsfc_{min} + fr + mu + tb + (distancia + mfac) * (raio \\ & + veloc_{media}))) - mftotal * veloc_{media} * (bmep_{bmep_{max}} + bsfc_{min}) + raio - raio / (altitude \\ & + dac + distancia) + tb / vel_{derrapagem} - (bmep_{menor_{zero}} - cos(bmep_{menor_{zero}} \\ & - n_{motor})) * (altitude + dac + distancia + cos(t - (fd + vel_{derrapagem}) / (bsfc_{min} + fr))) \\ & / ((cos((distancia - t) / fr) + cos(fctp / veloc_{media} + vel_{derrapagem} / mfac + mfac / cx) + \\ & cos(cos(bsfc_{min}))) * (cos((-fd + fr + t) * (fctp / veloc_{media} + vel_{derrapagem} / mfac + mfac / cx)) \\ & - cos(cos(-mu + rendtrans + cos(mfac / fr))) - mfac / fr) * cos(cx - mu) - cos(vel_{derrapagem}) \\ & + cos(veloc_{media} / fr) - cos(mftotal - n_{motor} + raio) + 1 - mfac / fr - fr / altitude) / (altitude + fr + \\ & mftotal - mfac * (-bsfc_{min} * c * cos(bmep_{menor_{zero}}) + cx * (altitude + vel_{derrapagem}) / mfac) \\ & * (bsfc_{min} * (bsfc_{bsfc_{bsfc_{min}}} - tb) * (rendtrans + wb) - cos(n_{motor} + 1)) * (mftotal + raio + \\ & cos(acceleracao)) * (-altura + distancia - mftotal * raio + n_{motor} + wb - cos(mfac)) * cos(a + rendtrans) \\ & / (bmep_{bmep_{max}} * bmep_{menor_{zero}} * c * fr) - mfac * veloc_{media} * (-c * cos(distancia) + (altitude \\ & + vel_{derrapagem}) * (distancia + wb) / mftotal) * (bsfc_{bsfc_{bsfc_{min}}} * c * raio * (rendtrans + wb) \\ & - cos(n_{motor} + 1)) * (cos((bsfc - vel_{derrapagem}) / (vel_{derrapagem} * veloc_{media}))) \end{aligned}$$

$$\begin{aligned}
& + \cos(\text{vel_derapagem}) / (\text{fp} - \text{mftotal}) * (\text{bmep}_{\text{menorzero}} - \text{bsfc}_{\text{min}} - t) * (\text{mftotal} + \text{raio} \\
& + \cos(\text{aceleracao}) * (\text{distancia} - \text{mftotal} * \text{raio} + \text{mu} + \text{n}_{\text{motor}} - \cos(\text{altura}) - \cos(\text{fr})) \\
& / (\text{bmep}_{\text{bmepmax}} * \text{bmep}_{\text{menorzero}} * c * \text{fctp} * \text{fr} * \text{n}_{\text{motor}} * (a + \text{fctp} * (\text{altitude} + \text{bmep}_{\text{bmepmax}} \\
& + \text{bsfc}_{\text{min}}) * \cos(\text{fr}/t) / (\text{fr} * \text{mu} * \text{veloc}_{\text{media}} + t))) / (\text{fr} + \cos((\text{distancia} - t) / \text{fr})) + \cos(\text{altitude} \\
& - \text{altura} + 3 * \text{bsfc}_{\text{min}} - \cos((-bsfc + \text{distancia} - \text{n}_{\text{motor}} - t) / \text{fr})) + (\text{distancia} - \text{fd} + \text{fr} + \text{n}_{\text{motor}} \\
& + (\text{altitude} - (\text{altura} - \text{bsfc}_{\text{min}} * c * \cos(\text{mftotal} + \text{raio} + \cos(\text{aceleracao})) + \text{cx} * \text{fr} / \text{mfac} + 2 * \text{distancia} \\
& - \text{mfac} + \text{raio} + (-bsfc + (-b - \text{bmep}_{\text{bmepmax}} + \text{bmep}_{\text{menorzero}} - \text{bsfc}_{\text{min}}) / (\text{altitude} * \cos(\text{mfac} \\
& / \text{bmep}_{\text{bmepmax}}))) * (\text{altura} - \text{mu} + \text{rendtrans} + \text{veloc}_{\text{media}} / (\text{bmep}_{\text{menorzero}} * (\text{aceleracao} + \text{cx}))) \\
& + (-\text{cx} + 2 * \text{distancia}) * \cos(\text{raio}) / \text{n}_{\text{motor}}) * (\text{altura} + \cos((\text{fr} + (-bsfc + (-2 * b + \text{bmep}_{\text{menorzero}}) \\
& / (\text{altitude} * \cos(\text{mfac} / \text{bmep}_{\text{bmepmax}}))) * (\text{altura} - \text{mu} + \text{rendtrans} + \text{veloc}_{\text{media}} / ((\text{aceleracao} + \text{cx}) \\
& * \cos(\text{bmep}_{\text{bmepmax}})))) * (\text{altura} - \text{bsfc}_{\text{min}} - \text{bsfc}_{\text{min}} * \text{mftotal} / \text{cx} + \text{mu}))) / \text{n}_{\text{motor}}
\end{aligned}$$

Apêndice D

Resultados completos da validação dos modelos

Nas tabelas D.1, D.2, D.3 e D.4 estão os erros absolutos entre os modelos gerados durante as experiências realizadas e o consumo real de combustível, para os diferentes conjuntos de validação utilizados.

D.1 Validação dos modelos das experiências da geração de modelo baseado no modelo da empresa

PG_1000_VF_F						PG_1000_VF_F_LD17							
Full 1	Full 2	Full 3	Grow 1	Grow 2	Grow 3	Half 1	Half 2	Half 3	Full 1	Full 2	Full 3	Grow 1	Grow 2
106.30%	106.30%	23.32%	106.30%	106.30%	106.30%	25.41%	20.04%	106.30%	105.51%	41.29%	20.28%	106.30%	106.30%
2.30%	2.30%	0.33%	2.30%	2.30%	2.30%	32.57%	41.54%	2.30%	100.00%	2.30%	42.73%	2.30%	2.30%
22.96%	22.96%	30.43%	22.96%	22.96%	22.96%	29.50%	25.04%	22.96%	22.96%	55.31%	32.81%	100.00%	22.96%
21.99%	21.99%	27.27%	21.99%	21.99%	21.99%	22.29%	30.11%	21.99%	20.66%	66.67%	33.34%	9.20%	21.99%
6.89%	6.89%	36.23%	6.89%	6.89%	6.89%	34.34%	38.63%	6.89%	6.23%	70.77%	43.42%	11.48%	6.89%
29.44%	29.44%	45.53%	29.44%	29.44%	29.44%	15.18%	26.42%	29.44%	100.00%	29.44%	26.64%	29.44%	29.44%
9.11%	9.11%	9.44%	9.11%	9.11%	9.11%	49.15%	36.94%	9.11%	6.80%	71.93%	41.57%	29.70%	3.99%
100.00%	3.26%	100.00%	100.00%	3.26%	49.72%	95.91%	100.00%	3.26%	63.77%	100.00%	61.28%	100.00%	3.26%
21.57%	21.57%	29.28%	21.57%	21.57%	21.57%	21.57%	29.58%	21.57%	1.96%	74.08%	31.54%	21.57%	21.57%
25.66%	35.14%	14.95%	28.33%	35.14%	35.14%	29.88%	21.06%	35.14%	3.19%	65.26%	30.74%	35.14%	35.14%
17.73%	19.65%	17.18%	20.88%	19.65%	2.50%	38.28%	29.32%	19.65%	19.44%	66.40%	35.31%	11.17%	3.41%
100.00%	68.42%	100.00%	100.00%	68.42%	15.63%	95.69%	100.00%	68.42%	40.91%	68.42%	36.76%	100.00%	68.42%
23.91%	23.91%	26.42%	23.91%	23.91%	23.91%	21.87%	26.84%	23.91%	20.98%	67.14%	34.18%	15.78%	23.91%
54.60%	54.60%	11.29%	54.60%	54.60%	54.60%	9.58%	12.36%	54.60%	54.60%	53.51%	16.63%	54.60%	54.60%
40.55%	40.55%	25.49%	20.97%	40.55%	40.55%	4.18%	20.42%	40.55%	48.87%	40.55%	19.49%	100.00%	40.55%
27.41%	27.41%	57.07%	28.98%	27.41%	27.41%	66.19%	59.19%	27.41%	46.72%	27.41%	58.89%	27.41%	27.41%
0.58%	0.58%	41.46%	0.58%	0.58%	0.58%	0.58%	41.17%	0.58%	21.91%	0.58%	42.87%	0.58%	17.22%
100.00%	36.60%	100.00%	100.00%	100.00%	24.21%	96.02%	100.00%	36.60%	52.08%	100.00%	48.67%	100.00%	36.60%
0.85%	0.85%	41.33%	0.85%	0.85%	0.85%	35.96%	41.99%	0.85%	34.45%	0.85%	42.08%	0.85%	0.57%
57.12%	57.12%	11.93%	24.78%	57.12%	57.12%	28.11%	10.94%	57.12%	21.15%	57.12%	12.47%	57.12%	57.12%
100.00%	5.17%	100.00%	100.00%	5.17%	51.16%	96.43%	100.00%	5.17%	66.76%	100.00%	64.33%	100.00%	5.17%
22.19%	22.19%	20.22%	22.19%	22.19%	22.19%	32.10%	28.48%	22.19%	100.00%	22.19%	28.80%	22.19%	22.19%
28.17%	28.17%	23.59%	28.17%	28.17%	28.17%	25.86%	23.62%	28.17%	100.00%	28.17%	4.18%	28.17%	28.17%
100.00%	6.35%	100.00%	100.00%	100.00%	46.24%	96.38%	100.00%	6.35%	62.71%	100.00%	60.16%	100.00%	6.35%
100.00%	33.63%	100.00%	100.00%	100.00%	33.08%	96.39%	100.00%	33.63%	53.10%	100.00%	51.02%	100.00%	33.63%
100.00%	77.26%	100.00%	100.00%	77.26%	6.90%	93.91%	100.00%	77.26%	37.76%	77.26%	36.36%	100.00%	77.26%
100.00%	40.36%	100.00%	100.00%	100.00%	28.12%	93.73%	100.00%	40.36%	50.69%	100.00%	47.62%	100.00%	40.36%

58.14%	58.14%	5.51%	58.14%	58.14%	58.14%	5.95%	7.97%	58.14%	100.00%	58.14%	9.76%	58.14%	58.14%	
100.00%	131.49%	100.00%	100.00%	100.00%	23.31%	92.64%	100.00%	131.49%	18.80%	100.00%	12.95%	96.76%	131.49%	
100.00%	67.56%	100.00%	100.00%	100.00%	12.48%	95.47%	100.00%	67.56%	41.28%	100.00%	36.98%	100.00%	67.56%	
100.00%	36.03%	100.00%	100.00%	36.03%	18.16%	90.29%	100.00%	7.29%	52.32%	36.03%	48.87%	75.89%	17.26%	
100.00%	116.25%	100.00%	100.00%	116.25%	26.11%	100.00%	100.00%	116.25%	24.04%	116.25%	18.67%	72.28%	116.25%	
100.00%	42.44%	100.00%	100.00%	100.00%	23.32%	96.21%	100.00%	42.44%	50.05%	100.00%	46.46%	100.00%	42.44%	
131.27%	131.27%	41.89%	131.27%	131.27%	131.27%	131.27%	45.14%	131.27%	131.27%	21.32%	24.67%	131.27%	131.27%	
42.02%	42.02%	13.92%	42.02%	42.02%	42.02%	11.45%	18.01%	42.02%	42.02%	58.52%	20.50%	42.02%	42.02%	
12.31%	12.31%	31.10%	12.31%	12.31%	12.31%	12.31%	33.33%	12.31%	100.00%	12.31%	34.73%	12.31%	12.31%	
154.79%	154.79%	49.53%	154.79%	154.79%	154.79%	47.96%	49.72%	154.79%	10.23%	154.79%	45.55%	154.79%	154.79%	
13.50%	6.35%	48.70%	0.47%	6.35%	6.35%	43.40%	32.22%	6.35%	6.38%	72.67%	43.58%	10.37%	4.76%	
6.39%	6.39%	35.78%	6.39%	6.39%	6.39%	6.39%	37.76%	6.39%	10.24%	6.39%	40.02%	6.39%	6.39%	
107.45%	107.45%	29.57%	107.45%	107.45%	107.45%	1.03%	25.21%	107.45%	107.45%	40.95%	10.86%	107.45%	107.45%	
72.62%	72.62%	0.90%	72.62%	72.62%	72.62%	6.53%	6.17%	72.62%	72.62%	61.68%	1.86%	72.62%	72.62%	
118.91%	118.91%	23.70%	118.91%	118.91%	118.91%	23.52%	26.97%	118.91%	118.91%	33.97%	12.07%	118.91%	118.91%	
119.52%	119.52%	22.98%	119.52%	119.52%	119.52%	23.13%	27.19%	119.52%	119.52%	26.95%	12.27%	119.52%	119.52%	
69.89%	69.89%	4.15%	69.89%	69.89%	69.89%	90.70%	4.30%	69.89%	52.90%	57.71%	9.13%	56.05%	69.89%	
93.19%	93.19%	21.45%	93.19%	93.19%	93.19%	18.79%	16.24%	93.19%	59.63%	56.59%	8.28%	93.19%	93.19%	
PG_1000_VF_F_LD17 (cont.)			PG_1000_VF_F_LD30				PG_LD35			PG_1000_VF_F_LDTA				
Grow 3	Half 1	Half 2	Half 3	Grow 1	Grow 2	Grow 3	Grow 1	Full 1	Full 2	Full 3	Grow 1	Grow 2	Grow 3	
106.30%	106.30%	4.77%	33.83%	11.78%	106.30%	48.11%	16.98%	102.50%	44.05%	16.44%	8.71%	16.34%	16.17%	
2.30%	2.30%	42.77%	100.00%	45.84%	13.22%	31.31%	2.30%	0.25%	34.86%	45.00%	2.30%	41.76%	100.00%	
22.96%	22.96%	22.96%	60.90%	34.86%	22.96%	38.87%	22.96%	22.96%	25.74%	32.51%	43.18%	12.00%	30.18%	
21.99%	15.40%	21.99%	62.29%	35.32%	0.07%	22.67%	19.20%	19.04%	28.29%	30.19%	46.32%	22.01%	30.73%	
6.89%	0.38%	6.89%	67.11%	43.33%	12.58%	27.99%	6.89%	6.12%	31.90%	38.83%	54.34%	30.70%	39.30%	
7.11%	100.00%	66.53%	29.44%	31.53%	0.79%	7.78%	29.44%	26.90%	22.99%	26.28%	29.44%	27.38%	100.00%	
9.11%	10.63%	0.29%	71.54%	42.13%	23.05%	45.28%	7.46%	5.65%	29.30%	37.75%	51.68%	31.27%	38.05%	
100.00%	61.57%	11.24%	100.00%	83.74%	100.00%	66.58%	84.74%	100.00%	53.47%	61.74%	48.79%	100.00%	100.00%	
21.57%	100.00%	47.68%	62.01%	35.04%	10.19%	21.53%	33.45%	12.96%	3.56%	33.30%	48.67%	31.54%	32.61%	
22.22%	2.71%	6.06%	59.29%	28.29%	14.51%	2.26%	27.00%	2.91%	32.36%	22.71%	42.33%	16.36%	23.26%	
19.65%	19.65%	41.01%	71.98%	36.48%	19.65%	34.51%	4.74%	23.16%	52.27%	35.99%	49.95%	32.54%	35.25%	
100.00%	37.50%	44.78%	100.00%	73.47%	64.65%	57.70%	75.12%	100.00%	68.42%	38.21%	12.89%	100.00%	100.00%	

23.91%	16.98%	23.91%	61.39%	34.31%	1.53%	28.44%	22.50%	21.29%	27.95%	29.07%	45.87%	20.30%	29.64%
54.60%	54.60%	54.60%	50.68%	18.06%	54.60%	5.46%	54.60%	54.60%	11.74%	11.43%	34.55%	3.99%	12.22%
6.36%	100.00%	39.87%	40.55%	25.73%	3.57%	9.83%	40.55%	35.97%	24.70%	23.28%	40.55%	21.77%	100.00%
27.41%	27.41%	60.61%	27.41%	62.11%	46.39%	64.28%	27.41%	39.59%	53.01%	59.78%	27.41%	60.14%	84.68%
0.58%	0.58%	46.06%	0.58%	47.44%	18.31%	32.81%	0.58%	6.80%	45.33%	45.06%	0.58%	41.74%	77.96%
36.60%	48.60%	17.42%	100.00%	78.49%	51.94%	52.73%	79.82%	100.00%	63.19%	48.54%	30.08%	100.00%	100.00%
13.53%	100.00%	55.18%	0.85%	46.72%	23.79%	34.08%	0.85%	3.03%	43.03%	43.90%	0.85%	44.18%	76.37%
57.12%	57.12%	17.06%	57.12%	17.43%	21.44%	2.28%	57.12%	37.37%	18.16%	13.11%	57.12%	9.01%	98.23%
100.00%	66.21%	18.49%	100.00%	85.06%	100.00%	69.37%	85.99%	100.00%	57.10%	65.35%	52.05%	100.00%	100.00%
0.26%	100.00%	71.74%	100.00%	35.95%	8.23%	31.02%	22.19%	22.19%	23.97%	30.45%	22.19%	31.59%	100.00%
5.64%	100.00%	100.00%	100.00%	32.15%	9.07%	31.82%	28.17%	28.17%	21.90%	26.30%	28.17%	27.84%	100.00%
6.35%	60.01%	8.58%	100.00%	83.25%	62.58%	72.70%	84.29%	100.00%	73.34%	61.74%	46.92%	100.00%	100.00%
33.63%	52.43%	14.86%	100.00%	78.87%	33.63%	70.02%	80.26%	100.00%	45.13%	48.98%	33.06%	100.00%	100.00%
100.00%	24.48%	52.37%	100.00%	72.08%	92.57%	49.68%	73.81%	100.00%	17.41%	35.49%	10.93%	100.00%	100.00%
40.36%	51.00%	20.65%	100.00%	77.80%	40.36%	57.44%	79.26%	100.00%	39.68%	49.89%	28.30%	100.00%	100.00%
58.14%	58.14%	16.95%	100.00%	14.91%	58.14%	14.56%	58.14%	58.14%	4.23%	10.69%	58.14%	11.04%	100.00%
131.49%	7.61%	98.99%	100.00%	63.55%	18.55%	47.32%	65.80%	100.00%	39.88%	15.18%	14.80%	100.00%	100.00%
67.56%	38.56%	44.04%	100.00%	73.60%	41.06%	39.97%	75.24%	100.00%	55.44%	39.40%	13.67%	100.00%	100.00%
100.00%	47.75%	9.24%	36.03%	78.63%	62.27%	59.63%	84.43%	100.00%	94.06%	51.61%	36.03%	100.00%	36.03%
100.00%	17.83%	73.66%	116.25%	65.98%	39.36%	50.11%	75.25%	100.00%	84.00%	19.92%	116.25%	100.00%	116.25%
42.44%	44.67%	22.44%	100.00%	77.57%	49.86%	49.73%	78.96%	100.00%	61.74%	47.97%	26.31%	100.00%	100.00%
131.27%	131.27%	17.98%	26.46%	26.30%	131.27%	45.74%	1.06%	114.37%	5.91%	34.38%	1.62%	35.69%	27.24%
42.02%	42.02%	42.02%	55.44%	22.48%	23.93%	29.89%	100.00%	42.02%	15.00%	17.89%	37.29%	3.44%	17.58%
100.00%	100.00%	44.11%	100.00%	38.80%	40.18%	30.55%	12.31%	6.05%	72.72%	36.76%	12.31%	33.95%	100.00%
145.40%	100.00%	15.23%	154.79%	38.62%	154.79%	80.22%	154.79%	154.79%	113.96%	42.63%	154.79%	47.24%	100.00%
8.26%	25.54%	20.23%	69.10%	41.86%	35.23%	45.67%	100.00%	21.84%	36.34%	38.67%	50.50%	33.37%	38.28%
6.39%	6.39%	38.63%	6.39%	42.07%	13.02%	26.93%	27.32%	1.98%	40.33%	40.04%	6.39%	37.20%	61.03%
107.45%	72.94%	94.03%	33.87%	13.29%	48.14%	33.16%	100.00%	81.62%	28.83%	19.78%	14.69%	37.92%	20.39%
43.48%	100.00%	19.76%	47.36%	6.23%	46.14%	0.43%	21.39%	72.62%	72.62%	0.63%	22.37%	0.54%	3.86%
118.91%	118.91%	118.91%	31.48%	13.64%	118.08%	5.34%	118.91%	118.91%	30.53%	23.17%	8.27%	59.28%	22.45%
119.52%	119.52%	119.52%	30.74%	13.95%	119.52%	56.41%	119.52%	119.52%	31.87%	23.49%	5.73%	65.92%	22.79%
69.89%	46.90%	64.81%	53.67%	11.76%	28.50%	20.87%	69.89%	53.88%	5.16%	4.47%	25.13%	4.59%	4.98%

27.01%	100.00%	17.58%	49.13%	0.41%	30.65%	1.11%	74.03%	69.49%	35.48%	1.23%	10.14%	9.18%	5.55%
PG_1000_VF_F_LDFTA (cont.)				Modelo da									
Half 1	Half 2	Half 3	Empresa										
39.98%	41.13%	8.89%	8.67%										
63.46%	29.02%	2.30%	42.54%										
75.65%	29.10%	37.33%	36.90%										
56.42%	14.38%	41.34%	33.15%										
61.55%	24.18%	46.27%	46.39%										
56.06%	10.17%	29.44%	35.45%										
53.68%	28.32%	44.84%	42.70%										
100.00%	3.26%	100.00%	43.49%										
60.56%	29.87%	47.92%	42.88%										
35.25%	16.90%	42.85%	50.03%										
50.32%	30.98%	49.50%	42.25%										
100.00%	68.42%	100.00%	19.15%										
53.85%	12.82%	43.07%	37.42%										
60.45%	20.93%	23.99%	27.57%										
49.41%	1.64%	40.55%	37.83%										
70.61%	52.97%	27.41%	23.76%										
63.15%	30.74%	0.58%	49.14%										
75.76%	36.60%	100.00%	75.27%										
64.61%	28.41%	0.85%	41.33%										
36.15%	4.40%	57.12%	13.99%										
100.00%	5.17%	100.00%	37.80%										
66.73%	8.62%	22.19%	23.19%										
60.94%	11.65%	28.17%	28.34%										
82.36%	6.35%	100.00%	94.02%										
33.63%	50.53%	100.00%	94.94%										
100.00%	77.26%	100.00%	95.12%										
40.36%	48.04%	100.00%	93.21%										
100.00%	17.58%	58.14%	25.96%										
48.87%	101.32%	100.00%	82.47%										

95.03%	53.50%	100.00%	86.86%
63.39%	73.58%	100.00%	86.83%
62.90%	116.25%	100.00%	81.02%
74.54%	42.44%	100.00%	31.35%
100.00%	37.41%	2.27%	16.14%
54.51%	5.42%	30.34%	25.06%
58.87%	21.79%	12.31%	51.75%
61.31%	115.47%	154.79%	29.42%
45.76%	34.30%	57.82%	37.85%
60.76%	24.77%	6.39%	47.23%
9.30%	40.42%	14.47%	12.05%
56.59%	37.02%	17.00%	1.45%
43.41%	69.63%	4.00%	73.95%
52.90%	77.39%	11.36%	74.61%
35.91%	12.33%	15.46%	76.47%
24.79%	9.01%	11.28%	73.70%

Tabela D.1: Erro absoluto entre os modelos gerados na 2ª fase capítulo 5 e o consumo real de combustível.

D.2 Validação dos modelos das experiências da geração de modelo baseado em dados reais

PG_1000_DR									PG_1000_DR_LD17				
Full 1	Full 2	Full 3	Grow 1	Grow 2	Grow 3	Half 1	Half 2	Half 3	Full 1	Full 2	Full 3	Grow 1	Grow 2
72.29%	98.33%	91.36%	114.21%	117.18%	90.47%	179.13%	81.38%	103.32%	58.29%	78.37%	65.15%	69.58%	75.53%
30.41%	55.85%	50.38%	63.57%	49.69%	118.68%	35.67%	29.98%	27.74%	71.27%	33.68%	53.27%	24.92%	19.70%
93.34%	108.03%	64.04%	99.30%	7.53%	99.38%	81.34%	115.68%	132.75%	84.48%	192.85%	103.96%	119.59%	111.05%
40.94%	8.06%	45.99%	4.30%	25.90%	119.23%	33.71%	16.39%	15.58%	81.69%	19.68%	32.07%	4.93%	7.11%
19.80%	30.75%	41.15%	29.90%	12.14%	15.31%	23.47%	25.89%	23.51%	17.30%	54.45%	1.39%	37.22%	31.56%
23.29%	66.62%	71.53%	111.93%	39.05%	137.90%	64.10%	49.34%	58.13%	45.92%	28.71%	136.23%	57.54%	36.80%
77.89%	48.20%	92.01%	111.71%	83.98%	65.75%	81.72%	59.35%	84.27%	60.70%	20.73%	128.37%	72.09%	62.04%

71.32%	118.33%	127.48%	61.81%	170.31%	144.20%	87.43%	92.37%	83.80%	118.47%	98.18%	143.78%	5.37%	119.93%	
63.91%	50.11%	57.06%	78.23%	81.18%	53.16%	52.40%	74.71%	68.84%	43.93%	31.42%	75.11%	81.02%	60.05%	
102.14%	131.30%	88.83%	136.89%	108.62%	159.41%	98.87%	109.44%	126.47%	34.62%	82.89%	121.11%	77.32%	174.64%	
78.32%	97.60%	60.87%	40.75%	54.49%	101.78%	71.58%	105.26%	42.99%	52.86%	69.97%	77.17%	70.96%	84.53%	
14.30%	6.48%	63.56%	6.21%	3.85%	20.83%	74.27%	1.39%	21.93%	3.62%	7.48%	17.61%	2.28%	1.63%	
44.71%	71.84%	34.49%	17.34%	33.74%	66.90%	50.62%	49.85%	25.11%	36.48%	54.25%	65.52%	68.47%	47.42%	
44.55%	44.25%	60.99%	10.78%	67.34%	80.15%	75.08%	54.55%	67.32%	49.49%	36.02%	82.06%	70.43%	81.42%	
88.50%	43.22%	117.28%	65.47%	0.05%	71.51%	56.10%	111.97%	71.17%	59.94%	25.94%	90.24%	89.60%	115.16%	
133.05%	136.09%	149.65%	124.64%	159.49%	194.88%	151.70%	147.77%	109.24%	154.77%	174.98%	149.18%	175.05%	150.02%	
82.05%	77.04%	76.34%	76.12%	149.39%	100.22%	83.30%	66.02%	126.47%	58.93%	60.97%	118.91%	91.27%	69.06%	
67.10%	121.84%	102.92%	112.69%	124.86%	77.35%	104.04%	118.32%	98.49%	120.09%	95.02%	5.86%	19.47%	25.55%	
124.11%	93.55%	68.77%	105.47%	102.87%	134.82%	88.13%	118.47%	99.62%	93.28%	118.19%	131.58%	112.44%	130.28%	
57.69%	71.87%	53.83%	49.45%	70.55%	26.46%	68.67%	27.26%	45.38%	144.03%	54.43%	64.80%	62.58%	62.15%	
101.90%	149.82%	124.55%	162.72%	135.20%	143.31%	95.92%	158.17%	154.91%	110.91%	109.05%	139.74%	82.15%	104.62%	
81.82%	31.48%	54.74%	42.90%	119.71%	83.31%	51.08%	7.42%	58.58%	53.46%	66.62%	199.39%	61.32%	9.39%	
78.36%	74.06%	41.58%	69.09%	73.47%	17.43%	75.87%	68.30%	57.95%	63.03%	61.16%	58.24%	40.01%	56.95%	
56.73%	64.84%	86.21%	63.99%	66.84%	90.32%	71.64%	38.34%	83.62%	116.19%	195.43%	85.00%	65.14%	61.32%	
137.16%	103.71%	131.43%	122.60%	81.64%	81.18%	117.47%	96.07%	91.25%	79.94%	145.53%	82.27%	65.75%	124.64%	
19.81%	85.98%	38.05%	62.71%	88.84%	89.71%	81.80%	56.04%	35.79%	52.24%	33.70%	76.67%	92.27%	74.87%	
69.68%	75.28%	37.99%	56.79%	113.59%	113.04%	81.74%	91.59%	53.26%	49.18%	95.63%	68.34%	77.39%	51.80%	
62.07%	91.21%	64.99%	74.67%	121.55%	67.77%	90.62%	67.05%	29.31%	15.76%	71.40%	80.46%	97.16%	53.57%	
PG_1000_DR_LD17 (cont.)							PG_1000_DR_LD1A						Modelo da	
Grow 3	Half 1	Half 2	Half 3	Full 1	Full 2	Full 3	Grow 1	Grow 2	Grow 3	Half 1	Half 2	Half 3	Empresa	
78.16%	72.48%	123.23%	65.97%	13.20%	104.28%	99.80%	144.56%	81.84%	95.31%	115.21%	20.44%	118.88%	60.18%	
56.61%	70.97%	48.11%	65.76%	86.31%	6.17%	20.43%	61.02%	84.26%	41.42%	41.43%	42.76%	59.70%	99.16%	
168.79%	50.16%	83.11%	115.47%	95.00%	31.82%	78.25%	108.42%	115.92%	68.77%	97.44%	169.94%	84.79%	40.08%	
28.09%	21.80%	13.02%	17.91%	7.75%	22.28%	34.75%	10.78%	34.93%	17.25%	0.82%	3.65%	68.33%	81.13%	
4.34%	13.69%	13.98%	40.97%	30.92%	37.65%	41.19%	0.63%	40.37%	19.25%	24.75%	2.14%	7.64%	73.75%	
96.48%	45.29%	54.14%	57.02%	175.76%	43.74%	76.17%	51.56%	79.75%	107.58%	84.11%	58.25%	81.57%	82.16%	
66.26%	57.37%	107.30%	64.34%	57.64%	92.90%	106.71%	120.87%	78.25%	59.30%	46.99%	36.29%	112.58%	52.75%	
99.23%	113.67%	117.45%	94.44%	139.05%	64.03%	117.07%	151.02%	112.02%	122.00%	116.30%	99.80%	128.95%	63.46%	
58.57%	49.17%	62.72%	74.08%	54.11%	53.50%	61.28%	66.02%	47.32%	84.66%	50.39%	62.83%	44.64%	80.10%	

113.62%	87.35%	101.46%	118.26%	102.09%	124.03%	119.68%	125.16%	133.60%	100.89%	3.09%	80.66%	67.14%	29.58%
57.38%	117.52%	113.97%	56.24%	74.24%	45.90%	97.54%	121.99%	35.90%	55.31%	107.36%	78.02%	91.70%	67.64%
68.04%	0.90%	20.68%	6.00%	16.76%	20.45%	6.99%	6.27%	7.85%	0.35%	13.14%	17.07%	5.70%	24.76%
58.21%	58.61%	65.53%	36.36%	45.88%	81.11%	59.12%	80.23%	98.81%	45.91%	94.11%	37.57%	35.78%	80.25%
46.46%	60.56%	49.41%	66.62%	75.87%	81.56%	84.11%	111.00%	106.65%	77.35%	97.09%	59.46%	105.08%	49.34%
10.93%	98.10%	41.50%	86.39%	138.55%	61.08%	47.60%	84.51%	93.42%	51.00%	23.81%	12.45%	92.56%	73.39%
134.25%	174.60%	142.21%	165.00%	158.84%	191.00%	172.00%	185.52%	99.82%	166.73%	143.82%	37.35%	192.49%	13.66%
15.75%	114.07%	82.23%	38.24%	82.28%	76.44%	100.45%	37.69%	67.06%	76.90%	53.82%	19.45%	108.64%	56.98%
121.97%	72.34%	90.09%	72.75%	88.08%	104.32%	134.07%	81.08%	97.84%	133.15%	124.73%	126.19%	14.17%	96.29%
135.65%	111.27%	110.21%	106.76%	113.50%	82.43%	106.27%	134.98%	113.17%	114.12%	114.42%	119.09%	93.85%	98.06%
79.12%	68.76%	17.56%	66.22%	64.16%	48.38%	87.42%	28.84%	41.12%	8.75%	5.26%	83.32%	65.96%	95.16%
154.65%	98.26%	146.71%	166.75%	112.05%	168.14%	126.58%	89.90%	105.64%	122.66%	169.83%	18.39%	151.79%	41.02%
76.29%	60.25%	62.95%	47.67%	60.09%	59.17%	31.94%	8.09%	44.13%	56.03%	40.80%	25.20%	60.02%	99.11%
33.00%	78.51%	70.04%	53.41%	55.90%	58.70%	51.85%	65.98%	65.94%	128.52%	102.16%	75.44%	63.41%	99.48%
78.73%	58.10%	61.44%	50.94%	77.17%	63.20%	79.86%	92.42%	54.91%	28.00%	68.74%	56.44%	38.38%	88.15%
116.09%	71.68%	95.97%	95.74%	108.89%	99.78%	110.19%	110.75%	124.08%	126.91%	89.01%	93.64%	43.32%	99.01%
84.50%	92.72%	48.92%	35.19%	96.94%	68.59%	34.92%	69.58%	72.94%	75.42%	61.00%	64.64%	75.31%	95.30%
79.33%	107.89%	50.59%	22.50%	80.34%	19.21%	89.01%	47.16%	37.01%	49.37%	60.02%	65.15%	132.95%	84.93%
156.70%	56.92%	94.12%	41.84%	71.73%	72.97%	2.85%	95.71%	65.21%	31.85%	42.38%	73.29%	76.47%	91.10%

Tabela D.2: Erro absoluto entre os modelos gerados no capítulo 6 e o consumo real de combustível.

D.3 Validação dos modelos das experiências da geração de modelo baseado no modelo fornecido adicionando fatores meteorológicos

PG_1000_VF_F_CM									Modelo da
Full 1	Full 2	Full 3	Grow 1	Grow 2	Grow 3	Half 1	Half 2	Half 3	Empresa
99.27%	82.46%	99.95%	46.67%	100.00%	38.89%	55.85%	81.25%	106.30%	8.67%
2.30%	2.30%	82.52%	2.30%	100.00%	0.25%	2.30%	2.30%	2.30%	42.54%
51.33%	13.38%	74.82%	80.32%	22.96%	64.65%	2.09%	22.96%	71.90%	36.90%

21.99%	21.99%	80.79%	21.99%	100.00%	19.58%	21.99%	21.99%	21.99%	33.15%
6.89%	17.40%	68.29%	17.72%	6.89%	25.07%	6.89%	34.18%	6.89%	46.39%
29.44%	100.00%	79.72%	29.44%	100.00%	26.84%	29.44%	29.44%	89.51%	35.45%
9.11%	3.81%	58.06%	76.99%	100.00%	47.50%	9.11%	82.97%	9.11%	42.70%
3.26%	20.75%	67.77%	20.45%	3.26%	27.63%	3.26%	36.42%	3.26%	43.49%
21.57%	21.57%	65.37%	21.57%	100.00%	8.81%	21.57%	21.57%	40.18%	42.88%
35.14%	34.32%	49.94%	71.43%	100.00%	34.88%	35.14%	78.87%	35.14%	50.03%
19.65%	41.44%	98.71%	74.89%	98.80%	89.03%	19.65%	19.65%	100.00%	42.25%
68.42%	29.95%	71.15%	11.18%	68.42%	15.48%	30.17%	68.42%	24.44%	19.15%
23.91%	23.91%	78.07%	23.91%	100.00%	21.47%	23.91%	23.91%	23.91%	37.42%
90.39%	18.47%	97.75%	100.00%	99.35%	74.12%	21.03%	90.24%	67.05%	27.57%
40.55%	7.60%	57.25%	8.25%	40.55%	1.59%	40.55%	11.65%	100.00%	37.83%
27.41%	27.41%	100.00%	88.38%	100.00%	84.56%	62.09%	93.31%	100.00%	23.76%
0.58%	0.58%	71.74%	0.58%	100.00%	26.62%	2.85%	0.58%	100.00%	49.14%
36.60%	4.80%	59.01%	5.13%	36.60%	4.33%	36.60%	15.88%	36.60%	75.27%
0.85%	100.00%	82.29%	0.85%	100.00%	1.18%	0.85%	0.85%	100.00%	41.33%
57.12%	52.93%	98.84%	37.24%	100.00%	57.59%	57.12%	83.92%	100.00%	13.99%
5.17%	25.94%	98.36%	72.31%	5.17%	74.31%	49.94%	90.38%	94.57%	37.80%
22.19%	18.01%	57.22%	74.35%	100.00%	41.08%	22.19%	80.99%	22.19%	23.19%
28.17%	29.79%	100.00%	71.88%	98.72%	88.20%	32.97%	19.87%	82.88%	28.34%
6.35%	18.33%	66.46%	18.04%	6.35%	25.46%	6.35%	34.50%	6.35%	94.02%
33.63%	41.82%	74.98%	11.82%	33.63%	32.85%	27.63%	33.63%	39.69%	94.94%
77.26%	47.28%	96.60%	26.80%	100.00%	51.94%	32.64%	82.16%	74.32%	95.12%
100.00%	40.36%	97.44%	28.44%	100.00%	45.69%	40.36%	81.98%	58.92%	93.21%
100.00%	58.14%	97.02%	43.95%	47.68%	33.58%	58.14%	80.31%	58.14%	25.96%
131.49%	31.03%	99.65%	9.57%	100.00%	37.48%	39.60%	76.00%	63.93%	82.47%
67.56%	23.91%	99.09%	55.17%	100.00%	59.63%	4.97%	84.81%	64.40%	86.86%
36.03%	34.49%	71.62%	11.90%	36.03%	32.02%	36.03%	36.03%	33.46%	86.83%
61.94%	26.28%	74.97%	48.12%	68.82%	57.76%	34.95%	116.25%	116.25%	81.02%
91.13%	24.72%	97.21%	81.39%	99.43%	75.53%	27.10%	90.49%	64.53%	31.35%
131.27%	67.64%	98.44%	6.04%	100.00%	37.43%	2.29%	76.73%	67.44%	16.14%
42.02%	40.79%	72.87%	6.25%	42.02%	28.81%	19.47%	42.02%	38.22%	25.06%

12.31%	12.31%	70.11%	12.31%	100.00%	15.78%	12.31%	12.31%	1.30%	51.75%
154.79%	154.79%	154.79%	154.79%	100.00%	52.89%	154.79%	51.39%	100.00%	29.42%
6.35%	15.22%	98.34%	78.27%	100.00%	77.38%	39.18%	91.28%	74.01%	37.85%
6.39%	6.39%	79.96%	6.39%	100.00%	20.15%	6.39%	6.39%	100.00%	47.23%
107.45%	33.37%	72.93%	107.45%	97.93%	74.96%	48.42%	100.00%	100.00%	12.05%
14.16%	72.62%	64.66%	72.38%	72.62%	52.88%	19.20%	72.62%	55.70%	1.45%
118.91%	40.39%	69.35%	118.91%	97.81%	73.55%	51.59%	100.00%	100.00%	73.95%
119.52%	30.70%	73.62%	119.52%	97.80%	73.47%	52.76%	100.00%	100.00%	74.61%
69.89%	49.63%	97.49%	32.55%	100.00%	54.05%	69.89%	82.92%	76.37%	76.47%
99.09%	70.53%	98.88%	43.17%	100.00%	50.11%	45.50%	82.34%	93.19%	73.70%

Tabela D.3: Erro absoluto entre os modelos gerados na 1ª fase A capítulo 7 e o consumo real de combustível.

D.4 Validação dos modelos das experiências da geração de modelo baseado nos dados reais adicionando fatores meteorológicos

PG_1000_DR_CM									PG_1000_DR_CM_LD17				
Full 1	Full 2	Full 3	Grow 1	Grow 2	Grow 3	Half 1	Half 2	Half 3	Full 1	Full 2	Full 3	Grow 1	Grow 2
88.56%	5.72%	76.82%	64.50%	120.38%	109.49%	0.45%	90.56%	93.64%	80.89%	109.50%	76.15%	62.80%	93.46%
36.12%	45.11%	79.60%	50.88%	84.57%	41.31%	31.56%	67.98%	39.02%	28.35%	48.82%	49.13%	36.92%	56.78%
62.62%	72.66%	78.93%	74.55%	104.74%	79.17%	92.82%	98.69%	117.24%	67.83%	185.56%	109.09%	44.06%	118.41%
32.07%	18.46%	38.15%	18.92%	21.17%	28.69%	6.04%	13.92%	30.64%	6.65%	49.48%	15.62%	8.95%	33.45%
28.17%	23.28%	30.51%	0.48%	31.31%	43.69%	31.52%	13.04%	44.33%	24.15%	35.19%	41.57%	33.50%	9.65%
33.54%	65.87%	66.67%	83.82%	71.74%	19.15%	82.00%	93.40%	55.36%	17.86%	104.69%	72.54%	67.45%	59.76%
64.09%	60.79%	91.09%	70.40%	38.70%	109.16%	63.16%	120.34%	50.12%	67.41%	84.59%	86.82%	79.30%	34.56%
98.47%	98.36%	97.45%	73.31%	121.48%	105.56%	99.80%	73.52%	67.48%	121.90%	104.05%	119.63%	38.12%	116.08%
19.59%	60.92%	87.94%	47.37%	74.94%	59.75%	60.06%	82.37%	54.03%	76.66%	82.21%	57.56%	11.42%	91.65%
67.80%	54.26%	113.84%	128.30%	99.79%	119.53%	84.84%	89.76%	166.16%	85.31%	115.46%	149.15%	5.60%	123.13%
96.68%	101.87%	121.11%	102.76%	77.66%	126.55%	185.49%	69.39%	69.98%	12.39%	128.23%	54.86%	81.83%	44.74%
51.07%	5.75%	15.15%	29.63%	1.35%	49.17%	49.00%	172.92%	15.27%	42.90%	29.64%	30.90%	28.68%	8.78%

79.56%	49.91%	49.85%	31.97%	29.64%	58.54%	1.73%	124.04%	74.00%	42.97%	86.64%	66.04%	65.06%	36.74%	
5.39%	71.53%	43.09%	80.62%	99.56%	93.37%	80.33%	66.75%	4.70%	58.66%	72.06%	68.06%	67.87%	75.53%	
114.93%	89.51%	119.25%	80.99%	87.74%	54.90%	76.19%	67.54%	88.66%	79.61%	73.09%	85.20%	88.01%	46.37%	
167.44%	103.56%	130.00%	102.80%	175.78%	120.52%	146.50%	97.02%	175.74%	164.46%	155.93%	5.41%	138.43%	128.97%	
16.55%	78.07%	47.28%	79.82%	33.73%	8.83%	87.31%	112.27%	13.80%	24.53%	96.18%	28.34%	20.77%	46.92%	
97.07%	80.91%	35.23%	110.79%	73.05%	14.89%	112.74%	102.11%	105.31%	110.32%	104.62%	104.73%	72.12%	62.74%	
36.72%	93.36%	84.86%	109.07%	63.97%	1.93%	80.77%	115.97%	74.62%	97.04%	97.53%	60.45%	110.27%	118.31%	
45.16%	45.12%	53.93%	57.08%	23.37%	33.24%	32.42%	33.20%	48.07%	71.21%	76.93%	66.38%	68.74%	35.49%	
191.79%	155.77%	110.77%	126.22%	145.21%	122.73%	147.94%	62.08%	112.19%	93.95%	182.98%	142.67%	18.85%	140.93%	
44.20%	75.90%	72.46%	113.31%	83.07%	5.93%	37.94%	81.52%	93.98%	70.32%	51.72%	71.71%	81.93%	41.62%	
27.92%	70.25%	106.68%	29.44%	65.00%	75.74%	67.62%	47.10%	25.73%	16.84%	18.92%	82.07%	59.41%	70.14%	
55.72%	53.51%	80.97%	22.21%	65.89%	74.24%	68.92%	96.09%	59.46%	92.20%	77.12%	80.71%	101.33%	52.65%	
80.55%	119.66%	81.03%	76.59%	116.73%	9.25%	25.50%	146.09%	80.92%	77.28%	143.23%	100.00%	4.87%	118.59%	
75.57%	92.03%	81.50%	81.20%	60.67%	68.73%	63.46%	104.01%	66.22%	59.31%	66.45%	75.36%	71.32%	66.01%	
127.69%	98.67%	125.13%	53.35%	96.88%	96.81%	58.09%	105.88%	51.10%	94.91%	74.35%	102.84%	81.69%	84.44%	
49.69%	64.62%	105.10%	44.37%	50.09%	10.38%	54.96%	27.15%	53.83%	94.11%	76.58%	104.14%	85.84%	75.82%	
PG_1000_DR_CM.LD17 (cont.)							PG_1000_DR_CM.LDTA						Modelo da	
Grow 3	Half 1	Half 2	Half 3	Full 1	Full 2	Full 3	Grow 1	Grow 2	Grow 3	Half 1	Half 2	Half 3	Empresa	
138.99%	90.13%	104.55%	137.76%	99.60%	74.37%	56.08%	66.13%	52.72%	145.27%	71.06%	131.12%	112.63%	60.18%	
23.10%	81.82%	51.40%	39.95%	52.77%	34.08%	53.76%	16.07%	44.07%	83.69%	20.88%	34.34%	0.14%	99.16%	
108.80%	79.50%	66.16%	93.48%	123.37%	84.17%	78.13%	102.05%	69.53%	61.77%	30.47%	66.79%	72.05%	40.08%	
12.16%	61.54%	20.80%	22.29%	39.51%	70.02%	24.93%	14.64%	14.65%	47.22%	1.06%	36.56%	64.06%	81.13%	
28.56%	43.72%	37.81%	12.97%	0.71%	16.46%	27.59%	36.64%	32.77%	25.03%	13.64%	33.29%	58.90%	73.75%	
69.79%	26.77%	75.33%	81.72%	6.22%	73.12%	35.88%	35.53%	40.93%	77.87%	18.96%	106.00%	88.77%	82.16%	
87.95%	144.46%	105.00%	71.19%	38.11%	64.45%	95.00%	77.77%	63.36%	84.21%	54.77%	90.79%	87.13%	52.75%	
46.18%	65.52%	94.19%	36.75%	81.55%	85.07%	73.94%	113.52%	39.46%	70.60%	97.70%	124.75%	137.81%	63.46%	
67.26%	74.49%	84.91%	59.52%	99.43%	63.60%	26.85%	71.79%	68.58%	70.84%	50.35%	52.44%	108.40%	80.10%	
87.96%	167.82%	62.26%	138.17%	137.54%	162.58%	58.59%	107.01%	148.61%	91.29%	116.75%	95.44%	197.75%	29.58%	
105.30%	90.79%	101.38%	100.42%	131.16%	100.23%	64.47%	113.78%	102.48%	96.01%	69.02%	107.75%	107.95%	67.64%	
7.50%	11.26%	10.89%	13.70%	29.85%	50.87%	17.68%	100.00%	13.77%	12.35%	2.34%	3.00%	12.94%	24.76%	
73.97%	28.25%	47.57%	52.14%	66.01%	56.55%	39.35%	30.43%	44.98%	82.32%	35.10%	85.04%	55.41%	80.25%	
103.68%	134.96%	115.67%	64.62%	88.45%	35.57%	106.65%	38.45%	84.53%	66.81%	43.06%	60.40%	160.56%	49.34%	

85.03%	150.72%	81.14%	96.90%	98.25%	85.80%	83.04%	81.21%	111.56%	47.01%	69.68%	42.23%	37.29%	73.39%
147.92%	86.49%	186.47%	122.68%	90.37%	146.47%	158.41%	134.98%	155.08%	157.24%	148.79%	32.04%	121.20%	13.66%
108.08%	42.43%	71.82%	91.10%	6.96%	68.91%	32.83%	65.42%	32.92%	74.84%	88.93%	10.49%	48.04%	56.98%
62.93%	111.18%	116.60%	162.94%	71.48%	110.89%	118.81%	2.26%	133.85%	106.90%	92.48%	2.86%	102.63%	96.29%
57.07%	139.95%	87.69%	113.87%	105.71%	126.58%	190.68%	96.24%	169.45%	30.33%	63.99%	138.90%	60.81%	98.06%
55.11%	88.52%	1.55%	87.28%	55.69%	1.73%	31.60%	55.82%	59.92%	59.94%	61.67%	73.74%	42.54%	95.16%
149.71%	19.12%	168.45%	100.44%	199.81%	131.17%	64.80%	125.09%	95.79%	168.01%	110.40%	73.26%	172.66%	41.02%
48.00%	52.71%	79.83%	70.45%	69.87%	108.95%	51.87%	67.34%	68.51%	8.57%	28.97%	88.08%	75.28%	99.11%
79.06%	97.07%	65.29%	4.56%	63.48%	116.47%	45.23%	43.09%	73.34%	128.98%	78.41%	64.17%	87.87%	99.48%
76.04%	118.25%	49.37%	80.26%	81.48%	51.94%	77.58%	45.65%	67.08%	101.81%	81.38%	67.00%	97.09%	88.15%
137.33%	10.02%	68.86%	95.35%	110.99%	105.95%	79.50%	91.11%	31.29%	91.05%	92.55%	21.62%	114.14%	99.01%
37.17%	36.54%	44.95%	58.30%	60.74%	48.36%	118.72%	43.00%	84.38%	46.30%	91.57%	62.14%	55.69%	95.30%
93.65%	99.72%	93.39%	79.50%	62.40%	8.57%	78.37%	116.40%	63.57%	6.54%	141.01%	65.78%	34.74%	84.93%
62.12%	5.05%	70.49%	57.63%	12.21%	66.09%	22.63%	62.82%	109.50%	33.79%	59.15%	110.02%	79.19%	91.10%

Tabela D.4: Erro absoluto entre os modelos gerados na 1ª fase B capítulo 7 e o consumo real de combustível.