



Rogério Cortez Brito Leite Póvoa

**Development of unimodal and multimodal
optimization algorithms based on multi-gene
genetic programming**

Tese de Doutorado

Thesis presented to the Programa de Pós-graduação em Ciências em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências em Engenharia Elétrica.

Advisor : Prof. Patricia Lustoza de Souza
Co-advisor: Prof. Bruno Araujo Cautiero Horta

Rio de Janeiro
March 2018



Rogério Cortez Brito Leite Póvoa

Development of unimodal and multimodal optimization algorithms based on multi-gene genetic programming

Thesis presented to the Programa de Pós-Graduação em Engenharia Elétrica of PUC-Rio, in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the undersigned Examination Committee.

Prof.^a Patricia Lustoza de Souza
Advisor

Centro de Estudos e Telecomunicações – PUC-Rio

Prof. Bruno Araujo Cautiero Horta
Co-advisor
UFRJ

Prof.^a Marley Maria Bernardes Rebuzzi Vellasco
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Omar Paranaíba Vilela Neto
UFMG

Prof. Jorge Luís Machado do Amaral
UERJ

Prof. Mauricio Pamplona Pires
UFRJ

Prof. Daniel Neves Micha
CEFET/RJ

Prof. Márcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, March 9th, 2018

All rights reserved.

Rogério Cortez Brito Leite Póvoa

Graduated in computer engineering at PUC-Rio in 2010. Master in electrical engineering with area of concentration in nanotechnology at PUC-Rio in 2013. Researcher in computational intelligence at LabSem/CETUC–PUC-Rio and at MSSM/LabMMol–UFRJ.

Bibliographic data

Cortez Brito Leite Póvoa, Rogério

Development of unimodal and multimodal optimization algorithms based on multi-gene genetic programming / Rogério Cortez Brito Leite Póvoa; advisor: Patricia Lustoza de Souza; co-advisor: Bruno Araujo Cautiero Horta. – Rio de Janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2018.

v., 188 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Otimização Numérica;. 3. Otimização Multimodal;. 4. Computação Evolucionária;. 5. Programação Genética;. 6. Programação Genética Multigênica.. I. Lustoza de Souza, Patricia. II. Araujo Cautiero Horta, Bruno. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

To my parents Rodolpho Antônio and Flávia Regina,
to my siblings Marcos and Sílvia Maria,
and to my wife Franciane.

Acknowledgments

To CAPES and PUC-Rio for the aid granted, without which this work could not be done.

To my advisor Prof. Patricia Lustoza for the support, teaching and orientation at the right times.

To my advisor Prof. Bruno Horta for all his teaching, advice and dedication that made me a better researcher and a better person.

To my great partners Adriano Koshiyama, Douglas Dias, Germano Penello, Guilherme Torelly and Pedro Pereira for teamwork.

To my friends Daniel Valente, Edson Oliveira, Eugenio Furtado, Guilherme da Silva, Luciano da Silva Junior, Mayk Ramos and Yan Gonçalves for all the conversations and discussions on the most varied research topics.

To all other friends of LabSem (PUC-Rio) and LabMMol (UFRJ).

To Alexandre Müller, Leila Meireles and the entire Krav Maga and Bahad Eight families for encouragement and unity. *Kidá!*

To my friends Jeizzon Mendes, Lucas Alves, Ricardo Oliveira Neto, Ronaldo Mello and Thiago Roldão for the encouragement, support and conversations about applications in the most diverse areas.

To my friends André Dumas, Bernardo Braga, Bruno Cardoso, Caio Araújo, Daniel Thomé, Igor Barreto, Marcello Soto, Marcos Vinicius Fernandes, Pedro Saboya, Thiago Vieira, Thiago Morais, Vitor Hugo Conti and Zacarias Dianin Filho who have encouraged and supported me.

To the all other friends and family that motivated and supported me.

To my siblings, Marcos and Sílvia Maria, and my parents, Rodolpho Antônio and Flávia Regina, for unconditional love, for inspiring me on the path of studies and for being my base of values and morals.

To my wife Franciane for her love, companionship and dedication during all the moments in the last 10 years.

Thank you.

Abstract

Cortez Brito Leite Póvoa, Rogério; Lustoza de Souza, Patricia (Advisor); Araujo Cautiero Horta, Bruno (Co-Advisor). **Development of unimodal and multimodal optimization algorithms based on multi-gene genetic programming.** Rio de Janeiro, 2018. 188p. Tese de doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Genetic programming techniques allow flexibility in the optimization process, making it possible to use them in different areas of knowledge and providing new ways for specialists to advance in their areas more quickly and more accurately. Parameter mapping approach is a numerical optimization method that uses genetic programming to find an appropriate mapping scheme among initial guesses to optimal parameters for a system. Although this approach yields good results for problems with trivial solutions, the use of large equations/trees may be required to make this mapping appropriate for more complex systems. In order to increase the flexibility and applicability of the method to systems of different levels of complexity, this thesis introduces a generalization by thus using multi-gene genetic programming to perform a multivariate mapping, avoiding large complex structures. Three sets of benchmark functions, varying in complexity and dimensionality, were considered. Statistical analyses carried out suggest that this new method is more flexible and performs better on average, considering challenging benchmark functions of increasing dimensionality. This thesis also presents an improvement of this new method for multimodal numerical optimization. This second algorithm uses some niching techniques based on the clearing procedure to maintain the population diversity. A multimodal benchmark set with different characteristics and difficulty levels to evaluate this new algorithm is used. Statistical analysis suggested that this new multimodal method using multi-gene genetic programming can be used for problems that requires more than a single solution. As a way of testing real-world problems for these methods, one application in nanotechnology is proposed in this thesis: the structural optimization of quantum well infrared photodetector from a desired energy. The results present new structures better than those known in the literature with improvement of 59.09%.

Keywords

Numerical Optimization; Multimodal Optimization; Evolutionary Computation; Genetic Programming; Multi-Gene Genetic Programming.

Resumo

Cortez Brito Leite Póvoa, Rogério; Lustoza de Souza, Patricia; Araujo Cautiero Horta, Bruno. **Desenvolvimento de algoritmos de otimização unimodal e multimodal com base em programação genética multigênica**. Rio de Janeiro, 2018. 188p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

As técnicas de programação genética permitem flexibilidade no processo de otimização, possibilitando sua aplicação em diferentes áreas do conhecimento e fornecendo novas maneiras para que especialistas avancem em suas áreas com mais rapidez. *Parameter mapping approach* é um método de otimização numérica que utiliza a programação genética para mapear valores iniciais em parâmetros ótimos para um sistema. Embora esta abordagem produza bons resultados para problemas com soluções triviais, o uso de grandes equações/árvores pode ser necessário para tornar este mapeamento apropriado em sistemas mais complexos. A fim de aumentar a flexibilidade e aplicabilidade do método a sistemas de diferentes níveis de complexidade, este trabalho introduz uma generalização utilizando a programação genética multigênica, para realizar um mapeamento multivariado, evitando grandes estruturas complexas. Foram considerados três conjuntos de funções de *benchmark*, variando em complexidade e dimensionalidade. Análises estatísticas foram realizadas, sugerindo que este novo método é mais flexível e mais eficiente (em média), considerando funções de *benchmark* complexas e de grande dimensionalidade. Esta tese também apresenta uma abordagem do novo algoritmo para otimização numérica multimodal. Este segundo algoritmo utiliza algumas técnicas de *niching*, baseadas no procedimento chamado de *clearing*, para manter a diversidade da população. Um conjunto *benchmark* de funções multimodais, com diferentes características e níveis de dificuldade, foi utilizado para avaliar esse novo algoritmo. A análise estatística sugeriu que esse novo método multimodal, que também utiliza programação genética multigênica, pode ser aplicado para problemas que requerem mais do que uma única solução. Como forma de testar esses métodos em problemas do mundo real, uma aplicação em nanotecnologia é proposta nesta tese: a otimização estrutural de fotodetectores de infravermelho de poços quânticos a partir de uma energia desejada. Os resultados apresentam novas estruturas melhores do que as conhecidas na literatura (melhoria de 59,09%).

Palavras-chave

Otimização Numérica; Otimização Multimodal; Computação Evolucionária; Programação Genética; Programação Genética Multigênica.

Table of contents

1	Introduction	26
1.1	Motivation	28
1.2	Objectives	28
1.3	Contributions	28
1.4	Work Description	29
1.5	Work Organization	30
2	Optimization Methods	31
2.1	Evolutionary Computation	31
2.1.1	Genetic Programming	34
2.1.2	Multi-Gene Genetic Programming	38
2.1.3	Optimization by Genetic Programming	42
2.2	Multimodal Optimization	45
2.2.1	Niching Methods	46
2.2.2	Clearing	47
2.2.3	Niching Methods for Metaheuristic Optimization Algorithms	49
3	Multi-Gene Genetic Programming for Numerical Optimization	51
3.1	Multi-Gene Parameter Mapping Approach	51
3.1.1	Multi-Gene Parameter Mapping Approach with Feedback	54
3.2	Niching Multi-Gene Parameter Mapping Approach	56
3.2.1	Niching MG-PMA Procedure	56
3.3	Domain Constraint with Periodic Boundary Conditions	57
4	Analyses of the Multigenic Approach	61
4.1	Analyses of Multigenic Individuals	61
4.1.1	Results of Analyses of Multigenic Individuals	64
4.2	MG-PMA with Feedback	76
4.2.1	Results of MG-PMA with Feedback	76
4.3	Analysis of Tournament Size and Crossover	77
4.3.1	Results of Analysis of Tournament Size and Crossover	78
4.4	Comparison with Known Methods	78
4.4.1	Results of Comparison with Known Methods	79
5	Analyses of Niching MG-PMA	82
5.1	High-Level Crossover Operator	85
5.1.1	Results of High-Level Crossover Operator	85
5.2	Local Optimization Frequency	87
5.2.1	Results of Local Optimization Frequency	88
5.3	Feedback Frequency	89
5.3.1	Results of Feedback Frequency	89
5.4	Comparison with Well-Established Algorithms	90
5.4.1	Results of Comparison with Well-Established Algorithms	91

6	Optimization of Quantum Well Infrared Photodetectors	97
6.1	Representation	99
6.2	Objective Function	100
6.3	Configuration and Parameterization of the Optimization Algorithms	101
6.4	Results	102
7	Conclusions and Future Work	106
7.1	Conclusions	106
7.2	Future Work	108
	Bibliography	110
A	Analyses of the Multigenic Approach – Numerical Results	123
A1	Introduction	123
A2	Numerical Results of the Analysis of Multigenic Individuals	123
A2.1	Set 1	124
A2.2	Set 2	128
A2.3	Set 3	132
A3	Numerical Results for MG-PMA	138
A4	Numerical Results for MG-PMA with Feedback	139
A5	Numerical Results for Analysis of Tournament Size and Crossover	141
A6	Numerical Results for Comparison with Known Methods	151
B	CEC'2015 Benchmark Functions	153
B1	Introduction	153
B1.1	Unimodal Functions	153
B1.2	Multimodal Functions	154
B1.3	Hybrid Functions	156
B1.4	Composition Functions	156
C	Analyses of Niching MG-PMA – Numerical Results	161
C1	Introduction	161
C2	Numerical Results for High-Level Crossover Operator Analysis	161
C3	Numerical Results for Local Optimization Frequency Analysis	166
C4	Numerical Results for Feedback Frequency Analysis	176
C5	Numerical Results for Comparison with Well-Established Algorithms	182
D	CEC'2017 Benchmark Multimodal Functions	185
D1	Introduction	185
D1.1	Test Functions	185
D1.2	Benchmark Functions	188

List of figures

Figure 2.1 Program tree that represents the expression (SIGMA (SET-SV (* (% X J)))) (Koza, 1992).	35
Figure 2.2 Genetic operator of direct reproduction for GP. The individual is copied to the next generation.	37
Figure 2.3 Genetic operator of crossover for GP. The crossing between the individuals P_1 and P_2 generate the individuals O_1 and O_2 .	38
Figure 2.4 Genetic operator of mutation for GP. A random subtree replaces the subtree that is below a selected point.	38
Figure 2.5 Example of a multigenic individual.	39
Figure 2.6 Example of a low-level crossover for multigenic individuals with four equations.	39
Figure 2.7 Example of a mutation for multigenic individuals with four equations..	40
Figure 2.8 Example of a high-level crossover. In this example, the operator exchanges two randomly selected functions (genes) of the individual 1 with two functions of the individual 2 (these functions are highlighted by red arrows). The functions that were not selected, arrows in blue, are concatenated to thereafter receive the genes of another individual.	40
Figure 2.9 Example of a high-level two-point crossover. In this example, the operator combines two parts of the individual 1 with the second part of the individual 2, and the first part of the individual 2 with the second part of the individual 1.	41
Figure 2.10 Example of a high-level single-point crossover. In this example, the operator combines a first part of the individual 1 with the second part of the individual 2, and the first part of the individual 2 with the second part of the individual 1.	42
Figure 2.11 Example of a high-level uniform crossover. In this example, the operator exchanges two randomly selected functions (genes) of the individual 1 with two functions of the individual 2 (these functions are highlighted by red arrows).	42
Figure 2.12 OGP area and its ramifications. The algorithms presented in this thesis are highlighted in green: (i) MG-PMA; (ii) MG-PMA with feedback; and (iii) Niching MG-PMA.	43
Figure 2.13 Transformation performed by typical POT method. (Pujol, 2008)	44

- Figure 2.14 Schematic representation of the PMA procedure. The aim of PMA is to find a program that maps the initial parameter values (x_{ic}) into the optimal parameter values (x_i^*) for a given system (c is a column of initial parameter values, selected by each tree/program). The current best candidate parameters correspond to adapted values from the mapping of the initial parameters. The fitness value indicates how well a program performs this mapping. As GP, PMA also utilizes random numbers as constants in its individuals. During the evolution process, the initial parameter values, the GP-constants and the objective function are always the same (static) while PMA functions, candidates parameters values and the corresponding fitness values change (dynamic). 45
- Figure 2.15 Snapshots of a simulation run of DE/nrand/1 of the Shubert 2D function, at the 0th, 50th and 100th generations. The graph to the right shows the fitness landscape with multiple pairs of clustered global peaks (optima) of the Shubert 2D function. (Li, 2016) 46
- Figure 2.16 Example of the clearing procedure for a maximization problem with eight individuals in a given generation. (a) Fitness values of the individuals ordered. (b) Individual 3 is selected as a winner and center of the first niche (clearing radius r) and the fitness value of the individuals 1 and 4 are reset to zero. (c) Individual 7 is selected as a winner and center of a new niche and the fitness of the individuals 2 and 8 are reset to zero. Individuals 6 and 7 are selected as winners as well as the first winners. 49
- Figure 3.1 Representation of individuals in PMA and MG-PMA. (a) PMA represents all parameters to be optimized by a single function. (b) MG-PMA partitions the optimization problem into k parts. In this example, for $k = 4$, MG-PMA assigns to each of the four equations the responsibility of optimizing a quarter of the problem. The optimal parameters are represented by points on the graphs. 52
- Figure 3.2 Schematic representation of MG-PMA with feedback procedure. MG-PMA with feedback is a linear combination of programs that maps the initial parameter values (x_{ic}) into the optimal parameter values (x_i^*) for a given system (where i varies from 1 to n and c varies from 1 to m). The current best candidate parameters correspond to adapted values from the mapping of the parameter values. The fitness value indicates how well a program performs this mapping. As GP, MG-PMA with feedback also utilizes random numbers as constants in its individuals. From time to time the parameter values are updated by those of the best candidate. During the evolution process, the GP-constants and the object function are always the same (static) while the parameter values, MG-PMA functions, candidates parameters values and the corresponding fitness values change (dynamic). 55

- Figure 3.3 Example of the normalization based on the PBC applied to an output (green dot) of a GP tree outside of the domain constraints. The linear transformation replicates images of the box and identifies where the outside dot should be (black dot). The domain constraints determine the size of the box. 59
- Figure 4.1 The mean number of evaluations (MNE) on logarithmic scale is plotted as a function of the tree depth d for the 12 functions of test set 1. The corresponding standard errors are represented by vertical bars. Each line color refers to a specific number of trees k (red, $k = 1$; green, $k = n/2$; blue, $k = n$). 65
- Figure 4.2 Box plot graphic showing the mean number of evaluations (MNE) for the test set 1 as a function of the number of trees k considering, for each k , an aggregation of all d values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot. 66
- Figure 4.3 Box plot graphic showing the mean number of the evaluations (MNE) for the test set 1 as a function of the tree depth d considering, for each d , an aggregation of all k values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot. 67
- Figure 4.4 The mean number of evaluations (MNE) on logarithmic scale is plotted as a function of the tree depth d for the 12 functions of test set 2. The corresponding standard errors are represented by vertical bars. Each line color refers to a specific number of trees k (red, $k = 1$; green, $k = n/2$; blue, $k = n$). 70
- Figure 4.5 Box plot graphic showing the mean number of evaluations (MNE) for the test set 2 as a function of the number of trees k considering, for each k , an aggregation of all d values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot. 71
- Figure 4.6 Box plot graphic showing the mean number of the evaluations (MNE) for the test set 2 as a function of the tree depth d considering, for each d , an aggregation of all k values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot. 72
- Figure 4.7 The mean of the best fitness values (MBFV) on logarithmic scale is plotted as a function of the tree depth d for the 15 functions of test set 3. The corresponding standard errors are represented by vertical bars. Each line color refers to a specific number of trees k (red, $k = 1$; green, $k = n/2$; blue, $k = n$). 73

- Figure 4.8 Box plot graphic showing the mean of the best fitness values (MBFV) for the test set 3 as a function of the number of trees k considering, for each k , an aggregation of all d values. This graphic displays the distribution of the MBFV based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot. 74
- Figure 4.9 Box plot graphic showing the mean of the best fitness values (MBFV) for the test set 3 as a function of the tree depth d considering, for each d , an aggregation of all k values. This graphic displays the distribution of the MBFV based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot. 75
- Figure 4.10 Variation of the mean of the best fitness values (MBFV) as a function of the tree depth d . Values were aggregated according to the nature (unimodal, multimodal or hybrid) of each function (c_1-c_{15}) of the CEC benchmark (test set 3). 76
- Figure 5.1 Bar chart with peak ratio (PR) results from Niching MG-PMA using four types of crossover operator for 20 benchmark multimodal functions (p_1-p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. (a) high-level crossover; (b) high-level single-point crossover; (c) high-level two-point crossover; and (d) high-level uniform crossover. 86
- Figure 5.2 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 150, for 20 benchmark multimodal functions (p_1-p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 88
- Figure 5.3 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 10, for 20 benchmark multimodal functions (p_1-p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 90
- Figure 5.4 Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1-p_{20}) with accuracy equal to $1e-01$. 92
- Figure 5.5 Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1-p_{20}) with accuracy equal to $1e-02$. 93
- Figure 5.6 Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1-p_{20}) with accuracy equal to $1e-03$. 94
- Figure 5.7 Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1-p_{20}) with accuracy equal to $1e-04$. 95

Figure 5.8	Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1 - p_{20}) with accuracy equal to $1e-05$.	96
Figure 6.1	Interband and intraband absorption in a quantum well structure (Penello, 2013).	97
Figure 6.2	Asymmetric superlattice with the thickness of the quantum wells and barriers proposed by Penello (Penello, 2018). The red line represents the defective quantum well.	98
Figure 6.3	Potential profile of the structure presented by Penello in 2018 (Penello, 2018), showing the modulus squared wave functions and oscillator strength <i>versus</i> energy.	99
Figure 6.4	Asymmetric superlattice with the thickness of the quantum wells and barriers for the structural optimization. The red line represents the defective quantum well.	100
Figure 6.5	Evolution curve of the mean best values of the oscillator strength in 10 runs performed by MG-PMA (in blue) and CMA-ES (in black). The red line is the oscillator strength value of the structure proposed by Penello (Penello, 2018).	104
Figure 6.6	Potential profile of the best structures found among the solutions obtained, in 10 runs, by: (a) GA; (b) PSO; (c) CMA-ES; and (d) MG-PMA. The potential profile shows the modulus squared wave functions and oscillator strength <i>versus</i> energy.	104
Figure B1	3-D map for 2-D Rotated High Conditioned Elliptic Function (c_1) (Liang, 2014).	153
Figure B2	3-D map for 2-D Rotated Cigar Function (c_2) (Liang, 2014).	154
Figure B3	3-D map for 2-D Shifted and Rotated Ackley's Function (c_3) (Liang, 2014).	154
Figure B4	3-D map for 2-D Shifted and Rotated Rastrigin's Function (c_4) (Liang, 2014).	155
Figure B5	Shifted and Rotated Schwefel's Function (c_5): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).	155
Figure B6	Composition Function I (c_9): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).	156
Figure B7	Composition Function 3 (c_{11}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).	157
Figure B8	Composition Function 4 (c_{12}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).	158
Figure B9	Composition Function 6 (c_{14}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).	159
Figure B10	Composition Function 7 (c_{15}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).	160
Figure C1	Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.	161

- Figure C2 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level single-point crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 162
- Figure C3 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level two-point crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 162
- Figure C4 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level uniform crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 163
- Figure C5 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 10, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 166
- Figure C6 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 10, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 167
- Figure C7 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 50, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 167
- Figure C8 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 50, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 168
- Figure C9 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 100, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 168
- Figure C10 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 100, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 169
- Figure C11 Bar chart with peak ratio (*PR*) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 150, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 169

- Figure C12 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 200, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 170
- Figure C13 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 200, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 170
- Figure C14 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 0, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 176
- Figure C15 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 50, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 177
- Figure C16 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 100, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 177
- Figure C17 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 150, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 178
- Figure C18 Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 200, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. 178
- Figure D1 3-D map for 2-D test functions (F_1 - F_{12}) (Li, 2013). 187

List of tables

Table 2.1	Terminology used by Evolutionary Computation.	32
Table 4.1	Benchmark functions from (Pujol, 2008). The table presents the name of each function, the number of parameters to be optimized and the global minimum. The Schwefel's function was performed with different number of parameters in order to analyze each method according to the dimensionality of the problem.	62
Table 4.2	Benchmark functions proposed for CEC'2015 competition. The table presents the name of each function, the number of the parameters to be optimized and the global minimum.	62
Table 4.3	Parameters used in the test sets 1 and 2.	63
Table 4.4	General configuration of MG-PMA.	64
Table 4.5	Average rankings of Aligned Friedman's and Iman-Davenport's tests for comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 1.	68
Table 4.6	Holm's procedure for pairwise comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 1. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = 1$ and $d = 9$.	68
Table 4.7	Mean of the best fitness values (MBFV) for all functions of the CEC'2015 benchmark found by SPS-L-SHADE-EIG, DEsPA, LSHADE-ND, MVNO-SH, PSO and MG-PMA algorithms. The best results among those found by the algorithms are in bold.	80
Table 5.1	Benchmark functions proposed for 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization. The table presents the abbreviation code, the name of each function, the variable ranges, the number of the parameters to be optimized and the number of global optima.	82
Table 5.2	Maximum number of function evaluations ($MaxFEs$) for each benchmark multimodal function.	83
Table 5.3	General configuration of Niching MG-PMA.	84
Table 6.1	Configuration of GA.	101
Table 6.2	Configuration of PSO.	101
Table 6.3	Configuration of CMA-ES.	102
Table 6.4	Configuration of MG-PMA.	102
Table 6.5	Results of GA, PSO, CMA-ES and MG-PMA for the structural optimization of the QWIP. The table shows the maximum, mean, minimum, and standard deviation values of the oscillator strength of the best individuals in each of the 10 runs for both algorithms. Being the maximum value referring to the best individual and the minimum to the worst individual among the solutions presented in these 10 runs.	103

Table A1	Numerical results for the test set 1 with the tree depth (d) equal to 2.	124
Table A2	Numerical results for the test set 1 with the tree depth (d) equal to 3.	125
Table A3	Numerical results for the test set 1 with the tree depth (d) equal to 5.	125
Table A4	Numerical results for the test set 1 with the tree depth (d) equal to 7.	126
Table A5	Numerical results for the test set 1 with the tree depth (d) equal to 9.	127
Table A6	Numerical results for the test set 2 with the tree depth (d) equal to 2.	128
Table A7	Numerical results for the test set 2 with the tree depth (d) equal to 3.	129
Table A8	Numerical results for the test set 2 with the tree depth (d) equal to 5.	129
Table A9	Numerical results for the test set 2 with the tree depth (d) equal to 7.	130
Table A10	Numerical results for the test set 2 with the tree depth (d) equal to 9.	130
Table A11	Average rankings of Aligned Friedman's and Iman-Davenport's tests for comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 2.	131
Table A12	Holm's procedure for pairwise comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 2. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n$ and $d = 9$.	131
Table A13	Numerical results for the test set 3 with the tree depth (d) equal to 2.	132
Table A14	Numerical results for the test set 3 with the tree depth (d) equal to 3.	133
Table A15	Numerical results for the test set 3 with the tree depth (d) equal to 5.	134
Table A16	Numerical results for the test set 3 with the tree depth (d) equal to 7.	135
Table A17	Numerical results for the test set 3 with the tree depth (d) equal to 9.	136
Table A18	Average rankings of Aligned Friedman's and Iman-Davenport's tests for comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 3.	137
Table A19	Holm's procedure for pairwise comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 3. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n/2$ and $d = 9$.	137
Table A20	Numerical results for MG-PMA with tree depth equal to 7, tournament size equal to 100 and high-level crossover operator.	138

Table A21	Numerical results for MG-PMA with feedback using the test set 3 with the tree depth (d) equal to 7.	139
Table A22	Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between MG-PMA and MG-PMA with feedback. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n$.	140
Table A23	Numerical results for MG-PMA with tournament size equal to 2 and the tree depth (d) equal to 7 using the test set 3.	142
Table A24	Numerical results for MG-PMA with tournament size equal to 25 and the tree depth (d) equal to 7 using the test set 3.	143
Table A25	Numerical results for MG-PMA with tournament size equal to 50 and the tree depth (d) equal to 7 using the test set 3.	144
Table A26	Numerical results for MG-PMA with restricted high-level crossover and the tree depth (d) equal to 7 using the test set 3.	145
Table A27	Numerical results for MG-PMA with high-level single-point crossover and the tree depth (d) equal to 7 using the test set 3.	146
Table A28	Numerical results for MG-PMA with high-level two-point crossover and the tree depth (d) equal to 7 using the test set 3.	147
Table A29	Numerical results for MG-PMA with high-level uniform crossover and the tree depth (d) equal to 7 using the test set 3.	148
Table A30	Average rankings of Aligned Friedman's and Iman-Davenport's tests for comparison between MG-PMAs with different tournament sizes and crossover operators.	149
Table A31	Holm's procedure for pairwise comparison between MG-PMAs with different tournament sizes and crossover operators. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n$, tournament size equals 2 and high-level crossover operator.	150
Table A32	Numerical results for MG-PMA with number of functions (k) equal to n ($n = 10$), tournament size equal to 2, high-level crossover, the tree depth (d) equal to 7 and local minimization using the test set 3.	151
Table A33	Numerical results for PSO with local minimization using the test set 3.	151
Table A34	Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between MG-PMA, PSO, SPS-L-SHADE-EIG, DEsPA, LSHADE-ND and MVMO-SH. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case SPS-L-SHADE-EIG.	152
Table C1	Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-1$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover.	163

- Table C2 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-2$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover. 164
- Table C3 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-3$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover. 164
- Table C4 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-4$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover. 165
- Table C5 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-5$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover. 165
- Table C6 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-1$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L150 sp. 171
- Table C7 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-2$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp. 172
- Table C8 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-3$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp. 173
- Table C9 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-4$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp. 174

- Table C10 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-5$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp. 175
- Table C11 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-1$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1. 179
- Table C12 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-2$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1. 180
- Table C13 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-3$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1. 180
- Table C14 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-4$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1. 181
- Table C15 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-5$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F10. 181
- Table C16 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-01$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case Niching MG-PMA L150. 182

Table C17 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-02$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.	183
Table C18 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-03$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.	183
Table C19 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-04$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.	184
Table C20 Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-05$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.	184
Table D1 Test functions from 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization and their properties	186
Table D2 Benchmark functions from 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization	188

List of Abbreviations

ACO – Ant Colony Optimization
AI – Artificial Intelligence
AIS – Artificial Immune System
CA – Cultural Algorithm
CGP – Cartesian Genetic Programming
CMA-ES – Covariance Matrix Adaptation Evolution Strategy
DE – Differential Evolution
EA – Evolutionary Algorithm
EC – Evolutionary Computation
ECGP – Embedded Cartesian Genetic Programming
EP – Evolutionary Programming
ERC – Ephemeral Random Constant
ES – Evolutionary Strategy
GA – Genetic Algorithm
GE – Grammatical Evolution
GP – Genetic Programming
LGP – Linear Genetic Programming
MA – Memetic Algorithm
MGGP – Multi-Gene Genetic Programming
MMGP – Multimodal Genetic Programming
MG-PMA – Multi-Gene Parameter Mapping Approach
MBFV – Mean of the Best Fitness Values
MMO – Multimodal Optimization
MNE – Mean Number of Evaluations
NM – Niching Method
OGP – Optimization by Genetic Programming
PBC – Periodic Boundary Conditions
PDGP – Parallel Distributed Genetic Programming

PMA – Parameter Mapping Approach
POT – Parameter Optimization and Tuning
PR – Peak Ratio
PSO – Particle Swarm Optimization
QW – Quantum Well
QWIP – Quantum Well Infrared Photodetector
SM – Stochastic Method

*O menor caminho entre dois pontos é uma reta,
mas raramente é o melhor.*

Anônimo, 2018.

1

Introduction

Optimization is the process of minimizing or maximizing a function with one or more variables in a given domain with its set of constraints on variables. There are several optimization algorithms applied in the most diverse areas as mathematics, chemistry, physics, engineering, medicine and economics (Tan, 2017, Ellefsen, 2017, Myszkowski, 2018, Jain, 2018) and they can belong to two classes: deterministic or stochastic.

Deterministic algorithms are those that given a set of input parameters and the same initial conditions will always produce the same output. The conjugate gradient (Fletcher, 1964) and Newton-Raphson (Ben-Israel, 1966) algorithms are classic examples of deterministic algorithms. Deterministic algorithms are well known and can be applied to various problems. However, they are not the best choice for all problems. For example, there is no deterministic algorithm to solve NP-complete problems in polynomial time (Garey, 1979, Aaronson, 2005, Leyton-Brown, 2014) and this kind of problem can be easily solved by stochastic methods (SMs) that make use of randomness in generating solutions and use so-called random number generators.

Stochastic methods (SMs) (Hannah, 2014) refers to a collection of optimization methods that make uses of the randomness. Given the same set of input parameters to a SM, this randomness causes that different outputs are obtained. This implies that the behavior of the SM can only be known by the probability. Metrics such as mean and standard deviations of several simulations are commonly used to validate the methods. SMs are usually faster in locating a global optimum when compared to deterministic methods for two types of functions:(i) black-box; and (ii) extremely ill-behaved (Liberti, 2005).

Genetic programming (GP) (Koza, 1992) is a SM, from computational intelligence, able to create computer programs or mathematical models automatically. Commonly applied for symbolic regression (Liu, 2016), classification (Espejo, 2010) and search based software engineering (Harman, 2013, Harman, 2014, Langdon, 2015, Langdon, 2017, Yeboah-Antwi, 2017), GP have few studies in the field of numerical optimization. However, there are promising GP optimization methods: (i) cartesian genetic programming (CGP) (Walker, 2007, Miller, 2013); (ii) embedded cartesian genetic programming

(ECGP) (Walker, 2007); (iii) parameter mapping approach (PMA) (Pujol, 2008); (iv) optimization by genetic programming (OGP) (Koshiyama, 2013); and (v) multimodal genetic programming (MMGP) (Yoshida, 2017).

Although these optimization algorithms based on GP are promising, few analyzes have been performed on them, using few functions of benchmark and having no applications in real problems (with the exception of OGP that has an application in the optimization of parameters of a spiking neural network for clustering problems (Silva, 2014)). This thesis proposes to carry out the first systematic study using unimodal and multimodal benchmarks and with a real application in optimization by GP.

Computational intelligence provides a variety of nature-inspired techniques employed in the development of intelligent systems. The combination of such techniques with computational nanotechnology is likely to deeply accelerate the development of nanotechnology applications. This area is called intelligent computational nanotechnology (Vilela Neto, 2012). Among the benefits brought by techniques of this area are: (i) automatic design of nanodevices, optimizing the structure, composition and other parameters; (ii) virtual experiments, predicting characteristics of experiments not yet performed; and (iii) optimization of parameters employed in molecular simulators and in the design of more efficient simulation processes. For instance, the optimization of quantum well infrared photodetectors (QWIPs) is an application that are considered in the present thesis.

The operation of a photodetector is triggered with the incidence of light that excites an electron and this electron is collected generating current. An experiment with a photodetector (more specifically the photoelectric effect (Einstein, 1905)) is behind one of the thoughts that revolutionized physics: the quantization of light in energy packets (Clauser, 1974). Photon detectors closely depend on the band structure of the material. By controlling the photodetector materials and their thicknesses, it is possible to manipulate the band structure and control the electronic states. Control of the extended states is a promising step in the search for photodetectors that operate at higher temperatures by reducing the dark current, whereas the electronic confinement above the barrier is intended to increase the selectivity range of the photodetectors (Penello, 2013). The manufacture of a QWIP (Levine, 1993) begins with the theoretical analysis. In 2013, Penello (Penello, 2013) developed an easy-to-use computational program that calculates the states of the heterostructure. With this program, it is possible to calculate the energy levels of the structures, the wave functions, the oscillator strengths between the levels and the absorption spectrum. Using this software it is possible to apply computational intelligence algorithms to control and optimize the photodetector materials and their thicknesses.

1.1

Motivation

PMA and OGP are optimization methods that use GP to map initial parameters into optimal input parameters for an objective function. In 2013, Koshiyama (Koshiyama, 2013) shows a potential of OGP for certain benchmark functions without the need of specify the domain of those functions. This advantage allows the use of OGP in problems that the domain is not known. However the behavior of this algorithm is little known and an analysis of the representation of its individuals and other benchmark functions is necessary. It is also interesting to apply GP to real problems with one (unimodal) or more (multimodal) solutions as a way to validate and improve the use of this technique in optimization problems.

1.2

Objectives

The objective of this research is to know the effects and behavior of OGP and from this method to develop new numerical optimization algorithms that are flexible enough to be applied to real problems with one (unimodal optimization) or more optimal solutions (multimodal optimization – MMO).

1.3

Contributions

According to the objectives presented above, this thesis focused on the development of optimization methods from OGP. The contributions provided by this work are listed below:

- **Multi-Gene Parameter Mapping Approach (MG-PMA):** parameter mapping approach (PMA) (Pujol, 2008) is a numerical optimization method that uses GP to find a mapping among initial guesses to optimal parameters for a system. Although this method yields good results for problems with trivial solutions, the use of a large tree/equation may be required to make this mapping appropriate for more complex systems. In order to increase the flexibility and applicability of the method to systems of different levels of complexity, this thesis introduced MG-PMA, a generalization using multi-gene genetic programming (MGGP) (Hinchliffe, 1996) to perform a multivariate mapping, avoiding large complex structures.
- **MG-PMA with Feedback:** this method aims not only at mapping initial fixed guesses but also at adapting these initial guesses dynamically. For complex problems, MG-PMA with feedback try to avoid getting stuck in a local minimum (or maximum) due to the generation of a new mapping scheme from a new set of parameters.

- **Niching MG-PMA:** a new method using multi-gene genetic programming to multimodal numerical optimization. The new algorithm uses some niching techniques based on the clearing procedure to maintain the population diversity, in order to perform a multivariate mapping among initial guesses to optimal parameters for a system. Statistical analysis suggested that this new multimodal method using multi-gene genetic programming can be used for problems that requires more than a single solution.
- **High-Level Crossover Operators:** two new multigenic crossover operators for MGGP, and its variations, inspired by genetic algorithms (GAs) (Holland, 1975) operators – high-level single-point crossover and high-level uniform crossover.
- **Domain Constraint with Periodic Boundary Conditions:** a new normalization technique to follow domain constraints based on the periodic boundary conditions (PBC) used by the molecular modelling (Leach, 2001). This normalization can be applied to other optimization algorithms such as GA, particle swarm optimization (PSO) (Eberhart, 1995) and differential evolution (DE) (Storn, 1995)).
- **Structural Optimization of QWIPs:** Through trial and error, guided by previous knowledge, specialists try to reproduce experimental results or design new QWIPs with certain energies, which ended up being monotonous and time-consuming. Applying MG-PMA to control and optimize the photodetector thickness, it is possible to find desired structures and tests new ideas in the area.

1.4

Work Description

This research was done as follows:

- **Bibliographic research regarding the topics:** evolutionary computation (EC) and stochastic optimization algorithms such as GA; GP; differential evolution (DE); evolutionary programming (EP); evolution strategies (ES); ant colony optimization (ACO); and particle swarm optimization (PSO).
- **In-depth bibliographic research on GP:** MGGP, linear genetic programming (LGP), grammatical evolution (GE), parallel distributed genetic programming (PDGP) and PMA.
- **In-depth bibliographic research on niching methods for MMO:** MMO, niching methods, clearing procedure and real-world applications.
- **Implementation of the methods:** MG-PMA; MG-PMA with feedback; and Niching MG-PMA.

- **Tests and analysis:** MG-PMA (varying: number of trees, tree depths, tournament size and crossover operators for multigenic individuals), MG-PMA with feedback, Niching MG-PMA (varying: crossover operators for multigenic individuals, local optimization frequency and feedback frequency) and comparisons of the proposed methods with established methods.
- **Application of MG-PMA:** structural optimization of QWIPs and comparison with GA, PSO and covariance matrix adaptation evolution strategy (CMA-ES) (Hansen, 2001, Hansen, 2006).

1.5

Work Organization

This thesis is organized as follows:

The next Chapter presents theoretical foundations necessary for understanding this work, which will consider subjects as EC, GP, MGPP, the area of the optimization by genetic programming, MMO, niching methods and recent applications in the MMO area. Chapter 2 also presents two new multigenic crossover operators for MGPP and its variations.

Chapter 3 presents the methods developed in this thesis (MG-PMA, MG-PMA with feedback and Niching MG-PMA) and the domain constraint with PBC.

Chapter 4 presents the analysis performed with MG-PMA: (i) the benchmark-based sensibility analysis and its results of the number of trees as well as the variations of the tree depth, allowing the creation of simple or more complex solutions; (ii) an approach to increase the dynamics of MG-PMA and the results of the comparison with MG-PMA; (iii) an analysis varying the tournament size and the types of crossover operators for multigenic individuals; and (iv) a comparison of the MG-PMA with established methods.

Chapter 5 presents the analysis performed with Niching MG-PMA: (i) crossover operator for multigenic individuals; (ii) local optimization frequency; (iii) feedback frequency; and (iv) comparison with well-established multimodal methods.

Chapters 6 presents the applications of MG-PMA for optimization of QWIPs comparing the results with GA, PSO and covariance matrix adaptation evolution strategy (CMA-ES) (Hansen, 2001, Hansen, 2006) results.

Finally, Chapter 7 presents the closing remarks of this thesis and suggestions for future works.

2

Optimization Methods

This chapter presents algorithms classified as SMs. Section 2.1 presents the evolutionary computation (EC) and genetic programming (GP) and multi-gene genetic programming (MGGP) techniques. These methods are traditionally used in problems of prediction, classification, or even the discovery of knowledge. However, in this work, these algorithms are used as numerical optimization methods. The field of numerical optimization by GP is also presented in this section. Finally, Section 2.2 presents the main concepts of multimodal optimization (MMO), niching methods (NMs), clearing and NMs for metaheuristic optimization algorithms.

2.1

Evolutionary Computation

Evolutionary computation (EC) consists of the design and analysis of algorithms inspired by Darwin's natural selection principles and their variations (Spears, 1993). In turn, the algorithms that make up the EC are called evolutionary algorithms (EAs).

According to (Back, 1997), EC originated in the late 1950s and remained relatively unknown to the scientific community for approximately three decades, mainly due to the lack of efficient computers at the time, but also due to the methodology not developed during the first surveys. During the sixties and early seventies, three independent proposals were fundamental to modify the image of EC that, from then on, began to be widely developed: (i) evolutionary programming (EP), proposed by Lawrence Fogel, Alvin Owens and Michael Walsh, whilst examining the use of simulated evolution as an approach for developing artificial intelligence (Fogel, 1966); (ii) evolution strategies (ES), created by Ingo Rechenberg and Hans-Paul Schwefel at the Technical University of Berlin, as experimental optimum-parameter seeking procedures and numerical optimization algorithms (Rechenberg, 1973, Schwefel, 1995); and (iii) genetic algorithms (GAs), introduced by John Holland at the University of Michigan, whilst working on the use of evolutionary techniques for adaptive systems (Holland, 1975).

These three areas were developed for about fifteen years as different representatives of EC. It was also in the early nineties that a fourth stream

following general ideas has emerged – Genetic Programming (GP) (Koza, 1992, Banzhaf, 1998). The contemporary terminology denotes the entire field by EC and considers EP, ES, GA and GP as its subareas.

As EC is based on evolutionary mechanisms found in nature, many terms adopted by EAs are based on biology. Table 2.1 presents a brief overview of some terminology used by EC.

Table 2.1: Terminology used by Evolutionary Computation.

Term	Meaning
chromosome	string of symbols
fitness function	the objective function
gene	basic unity of the chromosome
generation	the evolutionary order of different populations
genetic operations	a set of operations that the EA* performs on each of the chromosomes
genotype	structure
individual	a candidate solution
phenotype	a set of parameters, an alternative solution or a decoded structure
population	a set of individuals

* EA – Evolutionary Algorithm

The common idea behind all these techniques from EC is the same (Eiben, 2002): given a population of individuals, the environmental pressure causes natural selection (survival of the fittest) and thereby the fitness of the population improves. Therefore, EA as an optimization process (maximizing or minimizing an objective function) randomly creates a set of candidate solutions (called the initial population) and uses the objective function as an abstract fitness measure. Based on this fitness, some of the best candidates are selected to seed the next generation by applying recombination (also called crossover) and/or mutation. Often these candidates are called parents as they are used for creating new individuals (often called children). Recombination is usually applied to two selected parents and results in one or two children. Mutation is applied to any individual and it means making a random alteration to the chromosome. Applying genetic operators (recombination and mutation) leads to a set of new candidates, the offspring. The best parents can be promoted to the next generation without any change by the elitism concept. Thus, the new population is a combination of parents, offspring and their mutated counterparts. This process can be iterated until a solution is found (reaching a predefined convergence criteria) or a previously set time limit is reached. It is possible to see that many components of such an evolutionary process are stochastic.

According to Darwin, the emergence of new species, adapted to their environment, is a consequence of the interaction between the survival of the fittest mechanism and undirected variations. Variation operators must be stochastic, the

choice on which pieces of information will be exchanged during recombination, as well as the changes in a candidate solution during mutation, are random. On the other hand, selection operators can be either deterministic, or stochastic. In the latter case fitter individuals have a higher chance to be selected than less fit ones, but typically even the weak individuals have a chance to become a parent or to survive. The Algorithm 1 presents the general scheme of an EA.

```

1:  Generate initial population of size  $p$ . Set number of generation,  $g = 0$ 
2:  repeat
3:    Calculate the fitness of each member of the population
4:    Select a number of parents according to quality
5:    Recombine some, if not all, parents to create offspring chromosomes
6:    Mutate some parents and offspring
7:    Form new population from mutated parents and offspring
8:    Optional: promote a number of unaltered parents from step 4 to the new population
9:    Increment the number of generations  $g \leftarrow g + 1$ 
10: until ( $g$  equals number of generations required) OR (fitness is acceptable)

```

Algorithm 1: General scheme of an Evolutionary Algorithm (Miller, 2011).

This algorithm does not fall in the category of generate-and-test, also known as trial-and-error, algorithms. The fitness function represents a heuristic estimation of solution quality and the search process is driven by the variation operators (recombination and mutation creating new candidate solutions) and the selection operators. EA is distinguished within the family of generate-and-test methods since it relies on the population concept, i.e., process an entire set of candidate solutions and by the use of recombination to mix information of two candidate solutions. The population-based characteristic also distinguishes EA from the most common deterministic methods. In addition, EA uses probabilistic transition rules and it does not require additional information (e.g., derivatives) on the function to be optimized and the constraints.

Thus, a search for solutions can take place in non-convex sets with functions that are also non-convex and non-differentiable, being able to work simultaneously with real, logical and integer variables. It is also worth mentioning that EAs are not easily trapped to local minima as in the case of the usual deterministic methods. By using an EA, these characteristics may lead to the discovery of unconventional solutions that are not seen for being counter-intuitive. It is a paradigm that does not require prior knowledge of a way to find the solution.

According to Cortes (Cortes, 2004), EC has been successfully used for solving complex optimization problems. Its main obstacle is the accuracy of the solution to be found, because the closer to the optimum solution, the more computational power and processing time are required, especially when multimodal functions are used.

2.1.1

Genetic Programming

Genetic Programming (GP) is an EC technique that relies on the automatic creation of programs or mathematical equations to solve a wide range of academic and industrial problems (Koza, 1992, Banzhaf, 1998, Langdon, 2002, Poli, 2008, Miller, 2011). The origins of GP, and EC, go back to the origins of EAs (Fogel, 1998). As early as 1958, Friedberg devised an algorithm that could evaluate the quality of a computer program, make some random changes to it and then test it again to check for improvements, and so on (Friedberg, 1958, Friedberg, 1959). In the eighties, some approaches used GP for solving artificial intelligence (AI) problems (Smith, 1980, Forsyth, 1981, Cramer, 1985, Schmidhuber, 1987). However, GP became more widely known after the publication of John Koza's book in 1992 (Koza, 1992).

Considering a large, but restricted set of computer programs (also known as individuals), GP performs a search for the best individuals to solve a particular problem, synthesizing and selecting programs based on genetic and population operators. One of the obvious difficulties in evolving computer programs is caused by the fact that computer programs are highly constrained and must obey a specific grammar in order to be compiled.

Representation

Originally, the representation of computational programs used by Koza (Koza, 1992) employed the tree data structures and it is the most commonly used. These programs need to be run to get the corresponding candidate to a certain problem. This tree representation addresses the three aspects of problem representation:

- closure property (boundary conditions for the functions) – all functions must accept as argument any value and data type that can be returned by a function of the set of functions, or by a terminal (variables or constants) of the set of terminals. It is also possible to define special handling routines for specific exceptions;
- sufficiency property – the definition of the functions and terminals ends up specifying the search space of the possible programs. This space should be large enough to contain the desired solution. In addition, the defined functions and terminals must be appropriate for the problem domain. However, the larger this space, the less chance of finding the solution;
- the impossibility of maintaining syntactically incorrect and/or semantically invalid solutions can produce problems of low population diversity.

In 1958, John MacCarthy invented one of the oldest high-level computer languages that is widely used by researchers of the AI community: LISP (McCarthy, 1960). In LISP, all programs consist of expressions of lists of symbols enclosed in parentheses. For instance, calls to functions are written as a list with the function name first, followed by its arguments. For example, a function f that takes four arguments would be written in LISP as $(f \text{ arg1 arg2 arg3 arg4})$. All LISP programs can be written in the form of data structures known as trees (Koza, 1992).

As an example, Equation 2-1 has as analytical solution e^x .

$$\frac{dy}{dx} - y = 0 \quad (2-1)$$

The solution is equivalent to the Taylor power-series expansion (Equation 2-2).

$$\sum_{j=0}^{\infty} \frac{x^j}{j!} \quad (2-2)$$

Figure 2.1 presents an evolved expression that computes an approximate solution $(e^x - 1)$ to Equation 2-1 in a tree representation. The corresponding LISP expression is $(\text{SIGMA } (\text{SET-SV } (* (\% X J))))$ (Koza, 1992).

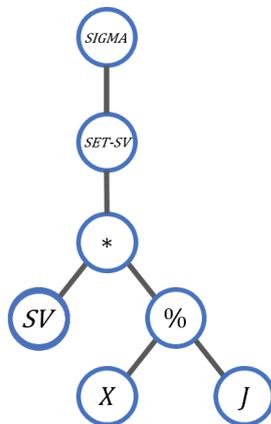


Figure 2.1: Program tree that represents the expression $(\text{SIGMA } (\text{SET-SV } (* (\% X J))))$ (Koza, 1992).

The function SIGMA is a one-argument function that adds its argument to the value stored in a register called SV. It is defined so as to increment an indexing variable J each time it does this. The function % is a two-argument function that implements protected division (it returns 1 if the denominator is close to zero). The function SET-SV assigns its single argument to the register. SV and J are defined to have initial values of 1. The program sums the successive arguments $x, x^2/2!$,

$x^3/3!$, $x^4/4!$ and so on, which is an approximate solution to Equation 2-1 and equal to $e^x - 1$.

The use of LISP ensure that GP can generate valid random trees during the evolution process. In this representation scheme, GP terminals are represented by input variables and constants (real or binary numbers, according to the problem) and appear at the extreme nodes of the tree (leaves). The GP functions appear in the internal nodes of the tree and can be arithmetic operations (“sum”, “subtraction”, “multiplication”, “division” etc.), mathematical functions (“sine”, “cosine”, “tangent” etc.), Boolean operations (“and”, “or”, “not” etc.), conditional operators (“if”, “else” etc.) and iteration functions (“for”, “while” etc.), to recursion or specific functions of the problem domain.

The synthesis of programs occurs from the sets of terminals and functions. For the programs created by GP to be valid, it is necessary that the sets of functions and terminals meet the closing property. This property determines that each function must accept as its arguments any value that can be returned by any function or terminal, ensuring that any generated tree can be evaluated correctly.

In order to apply the GP to a given problem, one must:

- determine the sets of terminals and primitive functions that are used;
- define a fitness measurement;
- establish a parameter to control an execution;
- define a method for determining the output;
- define a criterion to end an execution.

It is important to note that not all GP approaches use trees as a representation of their individuals. There are representations in the literature such as: (i) bitstrings (similar to programs written in machine code) used in Linear Genetic Programming (LGP) (Banzhaf, 1993, Miller, 2011); (ii) variable-length binary-string (a set of rules maps binary numbers into functions and terminals) used in Grammatical Evolution (GE) (O’Neill, 2003, Miller, 2011); (iii) a stack-based computer language used in PushGP (Spector, 2002); and (iv) graph representation (that allows more than one path between any pair of nodes) used in parallel distributed genetic programming (PDGP) (Poli, 1996) and in Cartesian Genetic Programming (CGP) (Banzhaf, 1998, Miller, 2000, Miller, 2011).

Evolution Process

In this algorithm, an evaluation function is used to assess how well a program performs a given task. For instance, in function approximation or prediction the

evaluation can be cast through the difference between the output generated by a GP individual and a target (error) – a well fitted individual yielding a negligible error.

The evolution process of the GP begins with the random creation of individuals which will form the initial population. This random creation can be done by some techniques (grow, full or ramped-half-and-half (Koza, 1992)) and the individuals have their sizes (tree depth) limited by one parameter. Each individual of the initial population is evaluated, and those of higher fitness are selected by some heuristics (roulette method, tournament etc. (Poli, 2008)) for the application of genetic operators. After the application of genetic operators, the resulting programs pass through the stopping criterion, which can be a maximum number of generations or the arrival at a satisfactory point of the problem. If the stopping criterion is not attended, the programs pass to the population of the new generation and the steps are repeated.

Genetic Operators

This Section presents some genetic operators used in GP (Koza, 1992, Vilela Neto, 2012):

- Direct reproduction – the individual of the population is selected according to some method based on fitness and is copied, without any change, to the next generation (Figure 2.2);

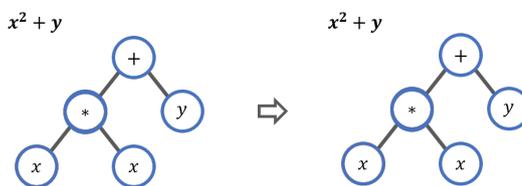


Figure 2.2: Genetic operator of direct reproduction for GP. The individual is copied to the next generation.

- Crossover – two individuals are selected, and a random cut-off point is chosen from each of the parents. The resulting sub-trees are exchanged, forming new programs for the next generation (Figure 2.3);
- Mutation – any tree edge is selected randomly. Then, this operation removes the subtree that is at this point and inserts a new, randomly generated subtree (Figure 2.4);

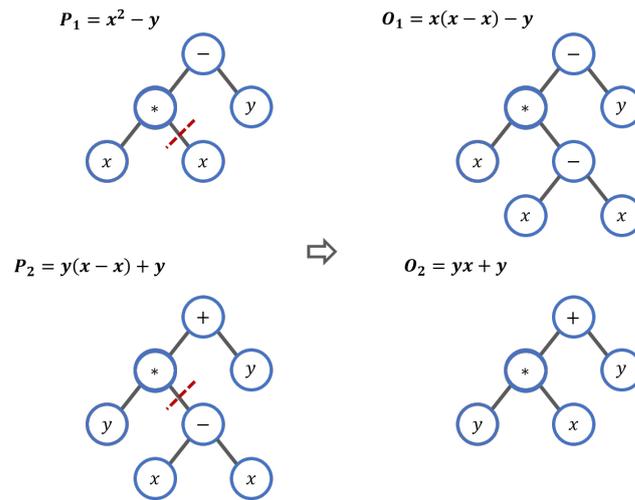


Figure 2.3: Genetic operator of crossover for GP. The crossing between the individuals P_1 and P_2 generate the individuals O_1 and O_2 .

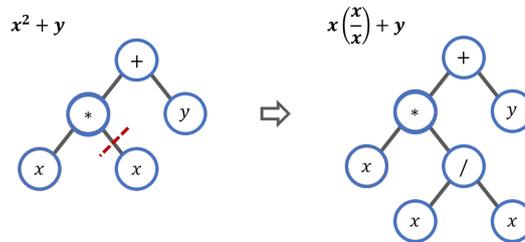


Figure 2.4: Genetic operator of mutation for GP. A random subtree replaces the subtree that is below a selected point.

2.1.2 Multi-Genetic Programming

Multi-gene genetic programming (MGGP) (Hinchliffe, 1996) is a generalization of the canonical GP, in which most components are similar, but MGGP denotes an individual as a complex tree structure, also called genes (G). Figure 2.5 shows an example of a multigenic individual with D genes (or number of trees).

Each tree structure can be considered as a partial solution to the problem, and the final output results from the linear combination of them (Equation 2-3).

$$\hat{Y}_i = \beta_0 + \sum_{d=1}^D \beta_d G_d \quad (2-3)$$

\hat{Y}_i is the resulting output of any individual of the population, β_0 is the linear coefficient (or intercept), β_d are linear coefficient and G_d (where $d = 1, 2, 3, \dots, D$, and D is the number of trees) are outputs of each decoded equation in the

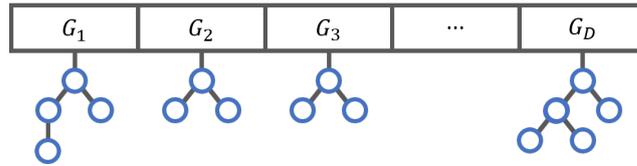


Figure 2.5: Example of a multigenic individual.

individual. The β_d of this method can be estimated by least squares, considering that the calculation of the error $\varepsilon_i = Y_i - \hat{Y}_i$ is feasible. From this error metric, it is possible to evaluate the overall quality of the individual. It is easy to see that when $D = 1$, the MGGP is reduced to the solution obtained by a classical GP.

Genetic Operators

Regarding the genetic operators, the mutation operation in the MGGP is similar to that performed in the classical GP. In the case of the crossover operation, it is necessary to distinguish the level at which the operation is performed as it is possible to apply this operation at the low and at the high level. Figure 2.6 shows an example of low-level crossover for multigenic individuals with four equations ($D = 4$). This operator manipulates the structures (terminals and functions) of the equations present in the individuals. In this example the operator exchanges random structures of individual 1 with random structures of individual 2.

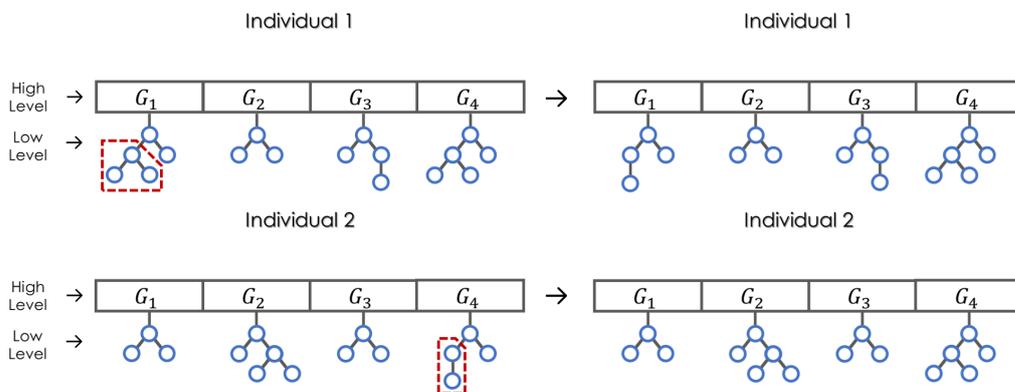


Figure 2.6: Example of a low-level crossover for multigenic individuals with four equations.

Figure 2.7 shows an example of mutation procedure for a multigenic individual with four equations ($D = 4$). This operator also manipulates the structures (terminals and functions) of the equations present in an individual. In this example the operator changes a random part of an equation of individual 1 by a random element in this equation.

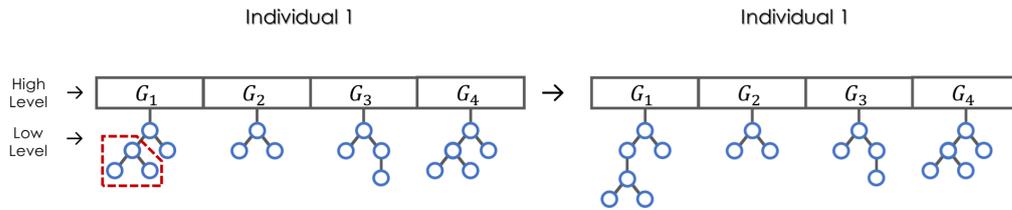


Figure 2.7: Example of a mutation for multigenic individuals with four equations..

Figures 2.6 and 2.8 show the two levels of a MGGP individual: low and high. The low level is characterized by the presence of four functions, arranged in tree structures. The high level represents the outputs of each function, similarly to a GA chromosome. A closer look at Figures 2.6 and 2.7 reveals that the results achieved by the two operations are quite similar, although the operations are conceptually different.

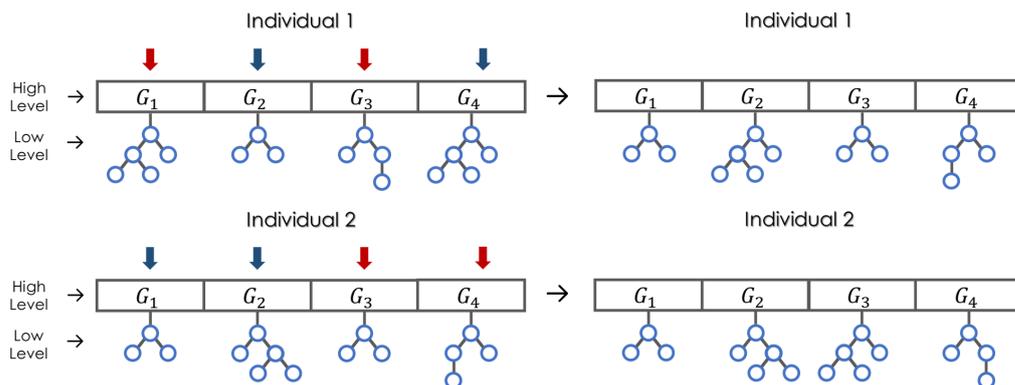


Figure 2.8: Example of a high-level crossover. In this example, the operator exchanges two randomly selected functions (genes) of the individual 1 with two functions of the individual 2 (these functions are highlighted by red arrows). The functions that were not selected, arrows in blue, are concatenated to thereafter receive the genes of another individual.

Crossover can also be performed at the high level (Figure 2.8). In this case, the operation consists of selecting random functions (genes) from one individual to be exchanged by functions (genes) also randomly selected from another individual. The new individuals will be formed by the genes that were not exchanged, as well as the genes that came from another individual. This high-level crossover was provided by MGGP GPTIPS 2 library (Searson, 2015). In comparison to the low-level crossover, it is clear that the high-level crossover implies changes in the macro level of the individual, whereas the low-level changes occurs in micro level.

Therefore, any changes in the high level tends to affect more an individual fitness than changes perpetrated on its low level.

The first version of the MGGP GPTIPS library (Searson, 2010) provides another high-level crossover operator, referred in this thesis of high-level two-point crossover. This operator selects two different individuals for the crossover and two random cut-off points are determined so that the first and third part of an individual is combined with the second part of the other (Figure 2.9).

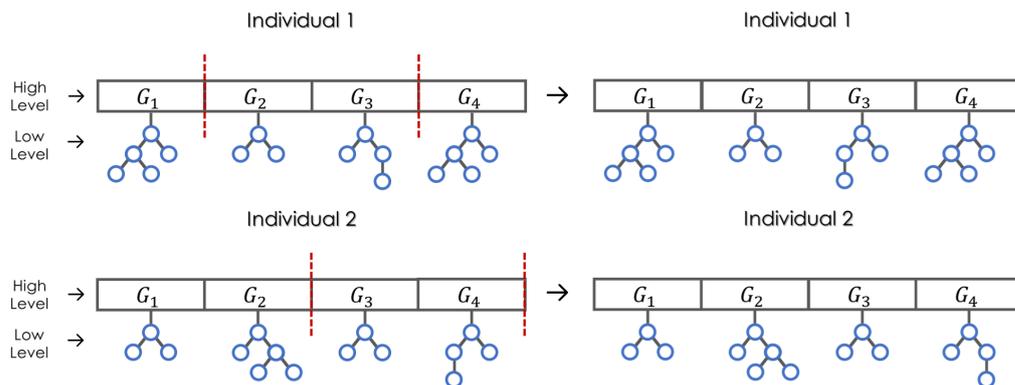


Figure 2.9: Example of a high-level two-point crossover. In this example, the operator combines two parts of the individual 1 with the second part of the individual 2, and the first part of the individual 2 with the second part of the individual 1.

New High-Level Crossover Operators

This section introduces two new multigenic crossover operators: (i) high-level single-point crossover; and (ii) high-level uniform crossover. These operators were created in this thesis from two GA crossover operators already well established in the literature (Kora, 2017). The high-level single-point crossover can be seen as an analogy to GA single-point crossover. This operator selects two different individuals for the crossover and a random cut-off point is determined so that the first part of an individual is combined with the second part of the other (Figure 2.10).

High-level uniform crossover can be seen as an analogy to GA uniform crossover. This operator selects two different individuals for the crossover and random genes from one individual are exchanged by random genes from the other individual (Figure 2.11).

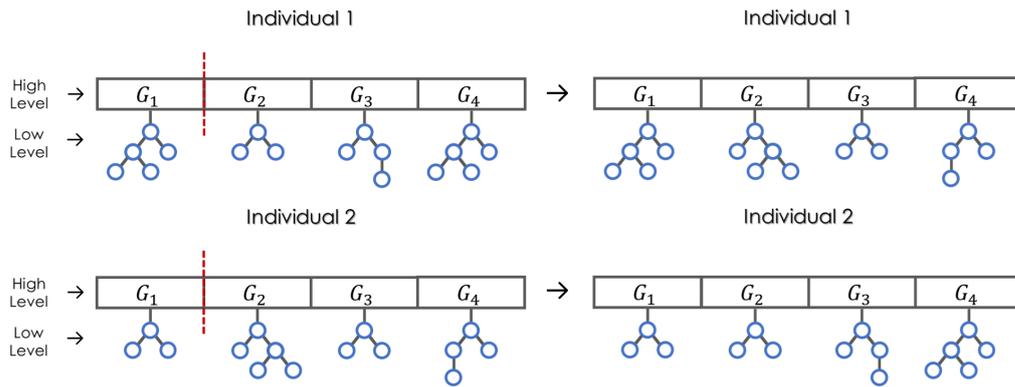


Figure 2.10: Example of a high-level single-point crossover. In this example, the operator combines a first part of the individual 1 with the second part of the individual 2, and the first part of the individual 2 with the second part of the individual 1.

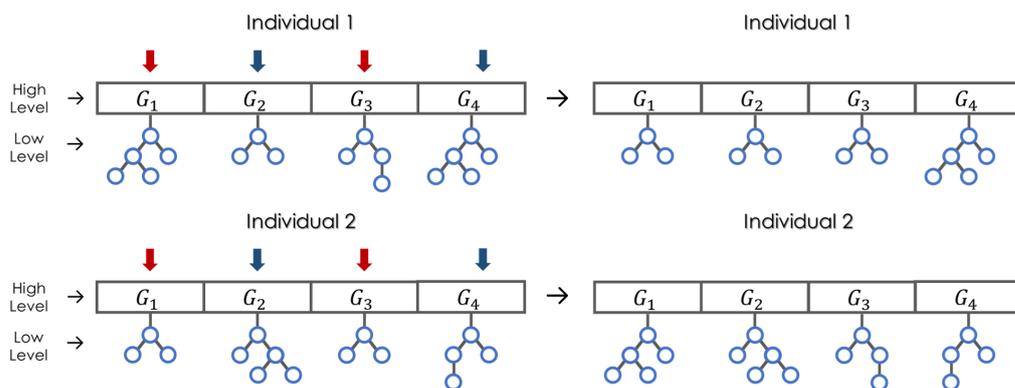


Figure 2.11: Example of a high-level uniform crossover. In this example, the operator exchanges two randomly selected functions (genes) of the individual 1 with two functions of the individual 2 (these functions are highlighted by red arrows).

2.1.3 Optimization by Genetic Programming

The importance of applying GP in the field of numerical optimization is the several applications in areas such as Machine Learning, Electrical Circuits and Financial Mathematics (Changra, 2009). In 2007, Walker and Miller used the cartesian genetic programming (CGP) and a extended form of CGP called Embedded CGP (ECGP) to solve real-valued optimization problems (Walker, 2007). The following year, Pujol and Poli introduced the parameter mapping approach (PMA) (Pujol, 2008), using the canonical GP. In 2013, Miller and Mohid proposed a new CGP approach to real-valued

optimization (Miller, 2013). In 2017, Yoshida, Harada and Thawonmas proposed a multimodal genetic programming using the canonical GP, called multimodal genetic programming (MMGP) (Yoshida, 2017).

In view of further improving GP-based optimization algorithms, the present work introduces a generalization of the PMA method. From this point on, the generalization of PMA introduced in the present work will be referred to as multi-gene parameter mapping approach (MG-PMA). Preliminary studies with early implementations of the multi-gene approach to PMA were performed.

The results were compared to those obtained by GA, PSO, DE and PMA (Koshiyama, 2013) and applied to optimize the parameters of a spiking neural network for clustering problems (Silva, 2014). There in, the method was referred to as optimization by genetic programming (OGP). However, since it was in fact a generalization of PMA, the acronym MG-PMA seems more appropriate as it acknowledges the previous PMA method. This thesis recommends the usage of OGP for referring to the emerging field of GP-based optimization methods.

In this work, MG-PMA was implemented with improvements and fixes code using the latest version of GPTIPS library (GPTIPS 2 (Searson, 2015)) with the aim of improving the original algorithm (Koshiyama, 2013). Chapter 3 presents the main MG-PMA concepts as well as two new methods developed in this thesis based on this algorithm: (i) multi-gene parameter mapping approach with feedback (MG-PMA with feedback); and (ii) niching multi-gene parameter mapping approach (Niching MG-PMA). Figure 2.12 shows the ramifications of the OGP area.

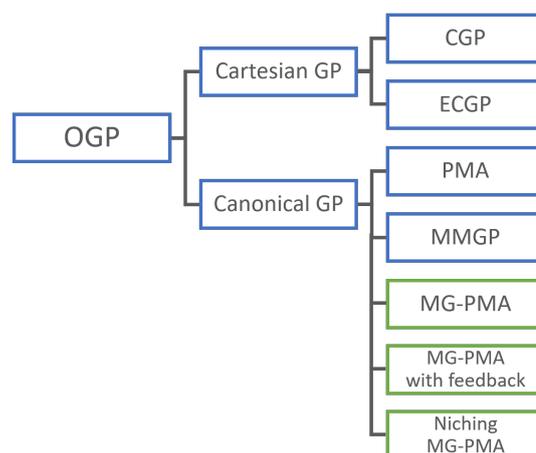


Figure 2.12: OGP area and its ramifications. The algorithms presented in this thesis are highlighted in green: (i) MG-PMA; (ii) MG-PMA with feedback; and (iii) Niching MG-PMA.

Parameter Mapping Approach

Parameter mapping approach (PMA) (Pujol, 2008) is a method based in GP to solve parameter optimization and tuning (POT) problems, where the maximum performance of a system is sought from the setting of the system parameters. POT methods are used to perform the tuning or the optimization of these parameters, being necessary at least one initial set of parameter values to start with. These methods then produce a set of adapted parameter values (see Figure 2.13).

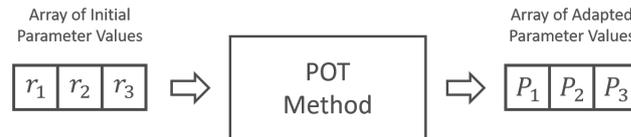


Figure 2.13: Transformation performed by typical POT method. (Pujol, 2008)

In PMA, a population of GP programs represents a set of possible mapping functions that transforms initial parameter values into adapted ones. These adapted values are submitted to an objective (evaluation) function, resulting in a value that defines the quality of the solution. GP iteratively improves such functions by mapping initial parameters until (sub)optimal values are found.

The variables of these mapping functions are represented by the initial parameters. As variables of a mathematical function can assume more than one value, the representation of these parameters can not be only an array as in Figure 2.13. The initial parameters are represented in the PMA by a matrix of n rows and m columns (where n is the dimensionality of the objective function and m is a user-configured number at the beginning of the optimization). Each mapping function, generated by GP, uses a random column as its variables and its constants are made available by GP constants. Usually random numbers are used as initial input values at generation zero, and remain unaltered during the evolutionary process. Figure 2.14 illustrates this process. Results found by Pujol and Poli (Pujol, 2008) suggested that PMA is as effective as well established methods in the literature such as GA (Holland, 1975), PSO (Eberhart, 1995) and DE (Storn, 1995).

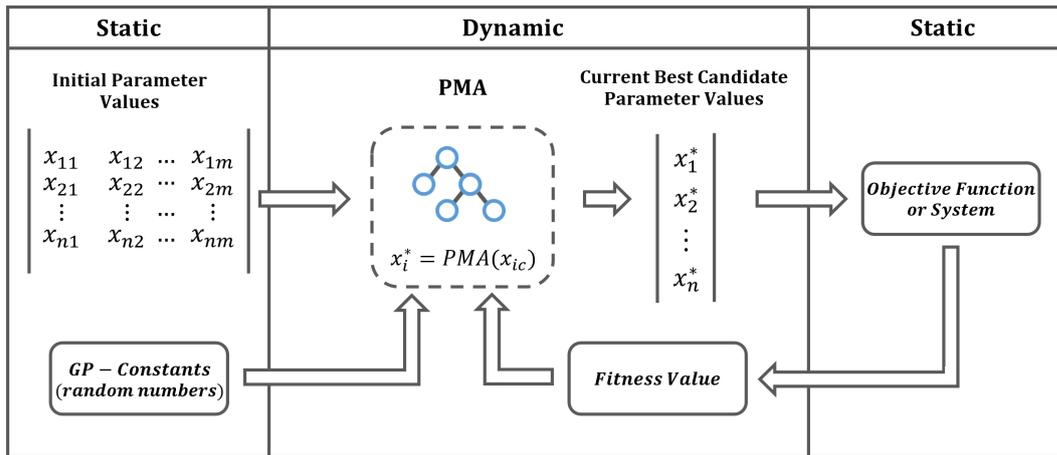


Figure 2.14: Schematic representation of the PMA procedure. The aim of PMA is to find a program that maps the initial parameter values (x_{ic}) into the optimal parameter values (x_i^*) for a given system (c is a column of initial parameter values, selected by each tree/program). The current best candidate parameters correspond to adapted values from the mapping of the initial parameters. The fitness value indicates how well a program performs this mapping. As GP, PMA also utilizes random numbers as constants in its individuals. During the evolution process, the initial parameter values, the GP-constants and the objective function are always the same (static) while PMA functions, candidates parameters values and the corresponding fitness values change (dynamic).

2.2 Multimodal Optimization

Multimodal optimization (MMO) aims finding multiple optimal and sub-optimal solutions in the search space in a single run (Deb, 2012). In the real-world, many problems have more than one satisfactory solution and, for some of them, it may be necessary to locate all global optima and/or some local optima. In other cases, finding more than one solution may improve knowledge about the problem or alternative solutions.

Methods developed especially for solving MMO problems are known as niching methods (NMs) and are predominantly population-based optimization methods such as evolutionary and swarm intelligence algorithms (Preuss, 2015). Sometimes “multimodal optimization” also refers to seeking a single optimum on a multimodal fitness landscape. To make this clearer denomination, NMs refer to “multi-solution” methods according to the work of Li, Epitropakis, Deb and Engelbrecht (Li, 2016).

2.2.1 Niching Methods

The first NMs appeared more than thirty years ago (De Jong, 1975, Goldberg, 1987) and currently several researches are being conducted applying them in several areas such as Engineering (Luh, 2011, Islam, 2017), Electromagnetics (Sareni, 1998), Medicine (Delibasis, 2010), Pharmaceuticals (Kruisselbrink, 2009), Management (Pérez, 2012, Pérez, 2016) and Games (Preuss, 2012).

Making a parallel between the areas of ecology and MMO (Glibovets, 2013), while a niche is a complex of specific living conditions or a subset of environmental resources in the first area, the term niche is used in the second area to name the subspace search space. Similarly, while in ecology species are a part of the set of all possible individuals, in MMO species represent the set of individuals with similar characteristics. Figure 2.15 shows the clustering tendency of the distributed individuals in the immediate vicinity of the global optima during the simulation run of a niching method (DE/nrand/1 – a DE method with a mutation strategy that incorporate spatial information about the neighborhood of each potential solution and exhibit a niching formation (Epitropakis, 2011)) on the Shubert 2D function, at the 0th, 50th and 100th generations (Li, 2016).

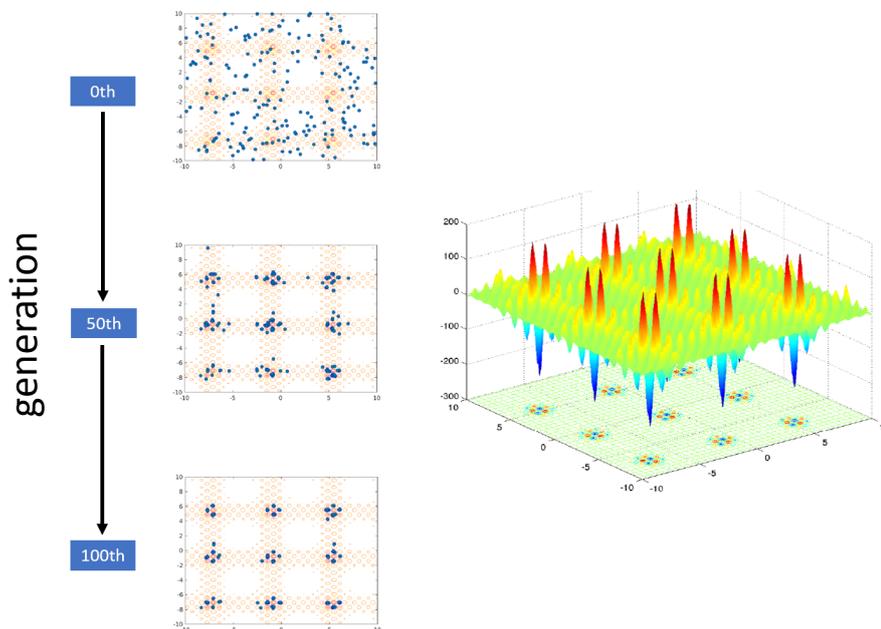


Figure 2.15: Snapshots of a simulation run of DE/nrand/1 of the Shubert 2D function, at the 0th, 50th and 100th generations. The graph to the right shows the fitness landscape with multiple pairs of clustered global peaks (optima) of the Shubert 2D function. (Li, 2016)

NMs are generally designed to solve MMO problems. A typical MMO problem can be expressed as:

Given a search domain χ , and an objective function f that maps elements of χ into a real domain \mathfrak{R} , assuming maximization of $f(\vec{x})$, where $\vec{x} \in \chi$ and \vec{x} is a n -dimensional vector (x_1, \dots, x_n) . In MMO an NM aims to locate as many as possible $\vec{x}^* \in \chi$ (not just a single \vec{x}^*) such that $f(\vec{x}^*) \geq f(\vec{x}), \forall \vec{x} \in \chi$. The mapped f values in the immediate vicinity of an \vec{x}^* should be all lower than $f(\vec{x}^*)$, which maximize the possible function response. In contrast to local optima, which although they are surrounded in their immediate vicinity by inferior solutions, the fitness values of local optima do not exceed the highest possible value (Fieldsend, 2014, Li, 2016).

Relaxing the MMO definition, it is possible to allow locating global optimal solutions, as well as “sufficiently” good sub-optimal solutions (Liu, 2016).

There are many NMs such as fitness sharing (Goldberg, 1987), crowding methods (De Jong, 1975, Mahfoud, 1992), derating (Beasley, 1993), restricted tournament selection (Harik, 1995), parallelization (Bessaou, 2000), clustering (Yin, 1993), stretching and deflation (Parsopoulos, 2001, Parsopoulos, 2004), speciation (Li, 2002) and clearing (Petrowski, 1996).

2.2.2 Clearing

Pétrowski proposed the clearing procedure (Petrowski, 1996) for GAs inspired by the principle of sharing of limited resources (Holland, 1975) within subpopulations of individuals characterized by some similarities. This procedure only supplies the resources to the best individuals of each subpopulation and it is naturally adapted to elitist strategies.

Algorithm 2 presents a simplified version of the clearing procedure. Considering a maximization problem, the current population P , with n individuals, is sorted by decreasing fitness values of individuals. Thus, the first individual is selected as a dominant individual, also called a winner, of a subpopulation associated with a current niche. All other individuals of P that have dissimilarity less than a certain threshold σ (the clearing radius) with respect to the winner, belong to this same subpopulation. This measure of dissimilarity, used to identify whether or not individuals belong to the same subpopulation, could be the Hamming distance for binary coded genotypes, the Euclidian distance for “real coded” genotypes or it could be defined at the phenotype level (Petrowski, 1996). The best individual of P who is not in this subpopulation becomes the winner in the next subpopulation. The next subpopulations are formed in the same way as the first subpopulation with the other individuals of the P .

```

Input :  $\Sigma$  and  $\kappa$ 
SortFitness( $P$ );
for  $i \leftarrow 1$  to  $n$  do
  if Fitness( $P[i]$ ) > 0 then
    nbWinners  $\leftarrow$  1;
    for  $j \leftarrow (i+1)$  to  $n$  do
      if Fitness( $P[j]$ ) > 0 AND Distance( $P[i], P[j]$ ) <  $\Sigma$  then
        if nbWinners <  $\kappa$  then
          nbWinners  $\leftarrow$  nbWinners + 1;
        else
          Fitness( $P[j]$ )  $\leftarrow$  0;
        end
      end
    end
  end
end

```

Algorithm 2: A simplified scheme of the clearing procedure (Petrowski, 1996).

All available resources in each niche are assigned to its dominant individual, this is done by preserving the fitness of the winners and resetting the fitness of the others to zero. The winners are then copied from the population, and placed on a list of winners, and the optimization process continues with the selection and genetic (e.g., crossover and mutation) operations. The clearing procedure is applied in the same way with the new populations. Thus, the list of all the winners is produced after a certain number of steps.

Niches can also accept more than one winner, and the maximum number of winners these niches can accept is called the capacity of a niche. In Algorithm 2, κ is the capacity of each niche, $nbWinners$ indicates the number of winners of the subpopulation associated with the current niche and P can be considered as an array of n individuals. The function $SortFitness(P)$ sorts the population P by decreasing fitness values of individuals, the function $Fitness(P[i])$ returns the fitness value of the i -th individual of the population P and the function $Distance(P[i], P[j])$ returns the distance between the individuals i and j of population P .

Figure 2.16 presents an example of the clearing procedure (with $nbWinners = 1$) for a maximization problem with eight individuals in a given generation. Figure 2.16.a shows a table with the fitness values of the individuals ordered, where individual 3 is the fittest and individual 5 is the least fit. In Figure 2.16.b, individual 3 is selected as a winner and center of the first niche with a pre-set clearing radius r . The fitness value of the individuals within this niche (individuals 1 and 4) are reset to zero. Then (Figure 2.16.c), the second best individual is selected (individual 7) as a winner and center of a new niche. As in the previous step, the fitness of the individuals within this second niche (individuals 2 and 8) are reset to zero. Individuals 6 and 7 are selected as winners as well as the first winners. Since there are no individuals within the niches of these two winners, no other individual has his fitness reset to zero. Then the optimization process continues with the selection

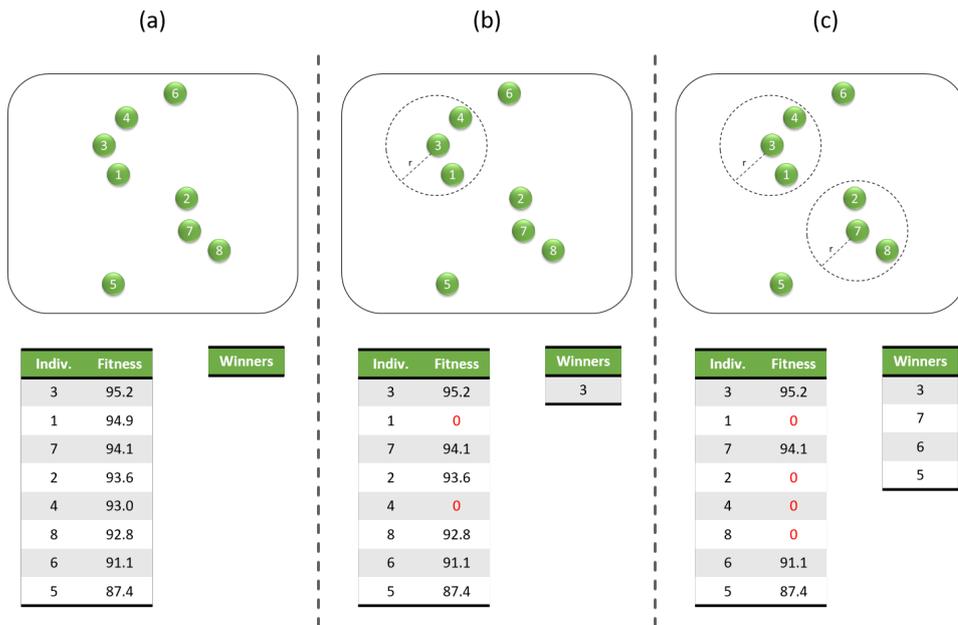


Figure 2.16: Example of the clearing procedure for a maximization problem with eight individuals in a given generation. (a) Fitness values of the individuals ordered. (b) Individual 3 is selected as a winner and center of the first niche (clearing radius r) and the fitness value of the individuals 1 and 4 are reset to zero. (c) Individual 7 is selected as a winner and center of a new niche and the fitness of the individuals 2 and 8 are reset to zero. Individuals 6 and 7 are selected as winners as well as the first winners.

of individuals for the application of the genetic operators, without giving chance to the individuals who were not selected as winners (individuals with fitness equal to zero) to be selected.

2.2.3

Niching Methods for Metaheuristic Optimization Algorithms

The main objective of the early NMs was to preserve population diversity due to loss of population diversity in standard EAs. Finding multiple optima was merely a byproduct of the use of NMs (Shir, 2008) and subsequent to early research, the main goal of NMs is to find multiple optimal solutions (Liu, 2016). NMs were also developed for other metaheuristic optimization algorithms (Engelbrecht, 2002), such as ES, PSO and DE and can be introduced to other meta-heuristics as well, such as Artificial Immune Systems (AIS) (Forrest, 1993, de Castro, 2002), Ant Colony Optimization (ACO) (Guntsch, 2002, Angus, 2009, Yang, 2017), and Cultural Algorithms (CA) (Ali, 2014).

Hybrid methods combining meta-heuristics with local search are commonly referred to as memetic algorithms (MAs) (Chen, 2011). This combination has presented great promise for global optimization, adding the refining capacity of the

local algorithms to the exploratory capacity of the global optimization algorithms (Chelouah, 2003). There was also success in the attempts (Peng, 2002, Wei, 2005, Ono, 2007, Bird, 2010) to hybridize NMs with local optimization algorithms, increasing the distributed convergence to multiple optima.

In 2017, Yoshida, Harada and Thawonmas proposed the first GP application for MMO, called multimodal GP (MMGP) (Yoshida, 2017) which separates the population into several clusters based on the tree structure similarity measurement (using the method tree structure similarity proposed by Yang (Yang, 2005)) and optimizes each cluster. MMGP was compared with a GP without clustering in a simple MMO problem with one global optimal solution and one local solution. MMGP was able to find the global and the local optimal solution simultaneously, in contrast to simple GP which was only able to find the global optimum solution or the local optimal solution in a single trial. Although MMGP was tested only on a simple multimodal problem, the method was able to maintain population diversity during optimization.

3

Multi-Gene Genetic Programming for Numerical Optimization

3.1

Multi-Gene Parameter Mapping Approach

In PMA, the parameters to be optimized are not individually encoded, but mapped by a single function. The method works fairly well for trivial problems, as shown by Pujol and Poli (Pujol, 2008). For example, consider $X_i^* = 0$ for all i , the optimal parameters for $f(\cdot)$. Thus, if an individual encodes the function $\text{PMA}(x_i) = x_i - x_i$, the optimization problem is solved. However, when the set of optimal solutions is not trivial, PMA tends to face more difficulties. Consider a second example in which $X_i^* = 1 + r_i$, where r_i is a random number generated from a uniform distribution on the interval $[-1, 1]$. In this case, it is more difficult to find a $\text{PMA}(X_i)$ function that solves the optimization problem. This is discussed in details by Pujol and Poli (Pujol, 2008).

As described in Chapter 2, this thesis presents MG-PMA, a generalization of the PMA method where a set of equations is responsible for certain parameters rather than a single one responsible for all. This is achieved through MGPP. Figure 3.1 depicts the main differences between a single tree and a multi-tree approach. While PMA represents all parameters to be optimized by a single function (Figure 3.1.a), MG-PMA separates the optimization problem into k parts, where k is the number of equations of MGPP. For example, for $k = 4$, MG-PMA assigns to each equation the responsibility of searching for a quarter of the solution. Figure 3.1.b shows the function found by MG-PMA to solve a hypothetical optimization problem. In the limiting case, when k equals the number n of parameters to be optimized ($k = n$), MG-PMA sets to each parameter an equation that receives an arbitrary value and returns it transformed by the decoded equation in the individual. By contrast, when $k = 1$, MG-PMA becomes PMA. Note that, in the limiting case ($k = n$), it is not mandatory to include initial parameters values (x_{ij} , as in Figure 2.13) in the terminal set of MGPP, but the ephemeral random constants (ERCs) are still required. Since each function is responsible for only one parameter of the objective function, this can be represented by a result of arithmetical combinations of constants. In this case, MG-PMA searches a function

of constants for each parameter.

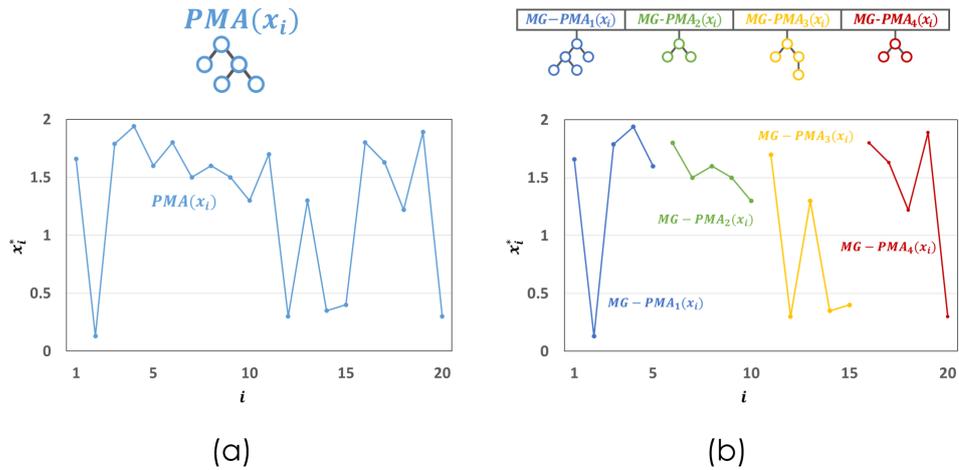


Figure 3.1: Representation of individuals in PMA and MG-PMA. (a) PMA represents all parameters to be optimized by a single function. (b) MG-PMA partitions the optimization problem into k parts. In this example, for $k = 4$, MG-PMA assigns to each of the four equations the responsibility of optimizing a quarter of the problem. The optimal parameters are represented by points on the graphs.

In general, the pseudo-code of MG-PMA, for any objective function, is described as in Algorithm 3. The `SplitInitialParameters` and `ApplyOperators` functions used in Algorithm 3 are presented by Algorithms 4 and 5, respectively.

```

Data: Objective Function
Output: Optimal Parameters
- InitialParameterValues  $\leftarrow$  Initialize();
- GPConstants  $\leftarrow$  Initialize();
- Set parameters for MG-PMA;
- Set  $k$ ;
-  $t \leftarrow 0$ ;
- SplitInitialParameters(InitialParameters,  $k$ );
- Start Population( $t$ );
- Fit  $\leftarrow$  Evaluate(Population( $t$ ));
while Stop criterion  $\neq$  FALSE do
     $t \leftarrow t + 1$ ;
    Population( $t$ )  $\leftarrow$  Select(Population( $t - 1$ ));
    Population( $t$ )  $\leftarrow$  ApplyOperators(Population( $t$ ));
    Fit  $\leftarrow$  Evaluate(Population( $t$ ));
end
    
```

Algorithm 3: Pseudocode of MG-PMA for a generic objective function.

Algorithm 4 shows how the initial parameters are split to the number of trees (k) in MG-PMA. If the number of initial parameters (n) is multiple of k , each tree is responsible for k/n consecutive parameters. For example, for $k = 4$ and $n = 20$, each tree will be responsible for mapping 5 initial parameters. If n is

```

Input : Initial Parameters and Parameter  $k$ 
-  $MaxParametersPerTree \leftarrow \text{ceil}(\text{CountRows}(\text{InitialParameters})/k)$ ;
-  $Parts[MaxParametersPerTree, k] \leftarrow$  Create segments with  $k$  parts;
-  $NumberParametersParts[k] \leftarrow$  All elements equal to 0;
-  $CountParameters \leftarrow 0$ ;
-  $IndexParameters \leftarrow 1$ ;
while  $CountParameters < \text{CountRows}(\text{InitialParameters})$  do
  for  $CurrentPart \leftarrow 1$  to  $k$  do
    if  $CountParameters < \text{CountRows}(\text{InitialParameters})$  then
       $NumberParametersParts[CurrentPart] \leftarrow \text{NumberParametersParts}[CurrentPart] + 1$ ;
       $CountParameters \leftarrow \text{CountParameters} + 1$ ;
    else
      break;
    end
  end
end
for  $CurrentPart \leftarrow 1$  to  $k$  do
  for  $Index \leftarrow 1$  to  $NumberParametersParts[CurrentPart]$  do
     $Parts[Index, CurrentPart] \leftarrow \text{InitialParameters}(IndexParameters)$ ;
     $IndexParameters \leftarrow \text{IndexParameters} + 1$ ;
  end
end

```

Algorithm 4: Pseudocode of the `SplitInitialParameters` algorithm: division of the initial parameters into k trees.

not a multiple of k , each tree will be responsible for mapping the initial parameters in the most distributed way, being the first trees responsible for mapping the largest number of parameters. For example, for $k = 8$ and $n = 20$, each of the first four trees is responsible for mapping three consecutive initial parameters (the first tree responsible for the initial parameters 1-3, the second tree for the parameters 4-6 and so on), while the other four are mapping two consecutive initial parameters.

Algorithm 5 shows how the genetic operators are used.

```

Input : Population
Output: New Population
-  $NewPopulation \leftarrow \text{ApplyElitism}(\text{Population})$ ;
while  $\text{Size}(NewPopulation) \neq \text{Size}(\text{Population})$  do
   $Parent1 \leftarrow \text{SelectOneIndividual}(\text{Population})$ ;
   $Flip \leftarrow$  Random Number Between 0 and 1;
  if  $Flip < \text{Mutation Probability}$  then
     $NewPopulation \leftarrow \text{ApplyMutation}(Parent1)$ ;
  else if  $Flip < \text{Sum of Mutation and Direct Reproduction Probabilities}$  then
     $NewPopulation \leftarrow \text{ApplyDirectReproduction}(Parent1)$ ;
  else
     $Parent2 \leftarrow \text{SelectOneIndividual}(\text{Population})$ ;
     $FlipCrossover \leftarrow$  Random Number Between 0 and 1;
    if  $FlipCrossover < \text{High Level Crossover Probability}$  then
       $NewPopulation \leftarrow \text{ApplyHighLevelCrossover}(Parent1, Parent2)$ ;
    else
       $NewPopulation \leftarrow \text{ApplyLowLevelCrossover}(Parent1, Parent2)$ ;
    end
  end
end

```

Algorithm 5: Pseudocode of `ApplyOperators` algorithm: MG-PMA genetic operators.

MG-PMA starts with the initialization of the initial parameters. The initial parameters are represented by an array with n rows and m columns, where n is equal to the dimension of the objective function and m is determined by the user at the beginning of the optimization¹. The reason for this matrix representation for the initial parameters is that they represent variables in the programs that are being evolved. Then, MG-PMA specifies the population parameters (e.g., population size and number of generations) and genetic operators (e.g., crossover rate and mutation rate), which are commonly defined in a GP execution. Once this is established, the user defines the number of functions encoded in an individual (represented by the natural number k). A randomly created population, consisting of individuals with k functions each. The individuals can select a random column from initial parameters by simulating its variables from the mathematical equations which, in turn, are evaluated through the objective function.

After the evaluation of the entire population, it is verified whether the stopping criterion was satisfied. If so, then the current population is returned, otherwise the algorithm enters in a loop that is only interrupted when the stop criterion is reached. During the loop, three operations are performed: selection, application of operators and evaluation. The first one chooses, based on some heuristic methods (e.g., roulette and tournament) (Poli, 2008), the entities of the following population. Subsequently, the crossover at low and high level, direct reproduction and mutation operators are applied in this new population. Finally, this new population is re-evaluated and again it is verified whether the stopping criterion is reached. If yes, then the current population is returned, otherwise the whole process is repeated.

Since MG-PMA is a numerical optimization method that uses MGGP to find a mapping among initial guesses to optimal parameters for a system, it is important to emphasize that this algorithm and its variations leave the advantage of the interpretability of the GP algorithm aside to focus on improving the accuracy of their solutions.

3.1.1

Multi-Gene Parameter Mapping Approach with Feedback

One of the most damaging problems in overall optimization is the premature convergence of the objective function to a local minimum (or maximum) (Michalewicz, 1996). Several factors can cause this problem: lack of genetic diversity, poorly adjusted control parameters, poorly established stopping criterion and complexity of a problem for a particular optimization method. Since MG-PMA works by mapping initial values into adapted ones, it is natural that for some problems, this mapping may be very complex or even impossible. The MG-PMA

¹In all the simulations of this thesis the value of m was equal to 100.

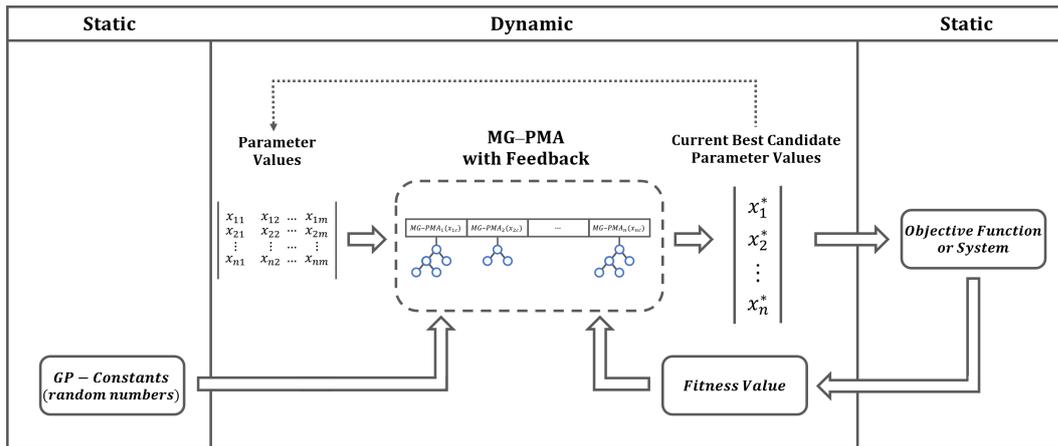


Figure 3.2: Schematic representation of MG-PMA with feedback procedure. MG-PMA with feedback is a linear combination of programs that maps the initial parameter values (x_{ic}) into the optimal parameter values (x_i^*) for a given system (where i varies from 1 to n and c varies from 1 to m). The current best candidate parameters correspond to adapted values from the mapping of the parameter values. The fitness value indicates how well a program performs this mapping. As GP, MG-PMA with feedback also utilizes random numbers as constants in its individuals. From time to time the parameter values are updated by those of the best candidate. During the evolution process, the GP-constants and the object function are always the same (static) while the parameter values, MG-PMA functions, candidates parameters values and the corresponding fitness values change (dynamic).

approach with feedback aims not only to map initial fixed values but also at adapting these initial values dynamically. It is then expected that it would avoid getting stuck in a local minimum (or maximum) due to the generation of a new mapping scheme from a new set of parameters.

This update of the initial values is made from time to time during the evolution process, through a control parameter, or when the optimization converges to a certain value for many generations (determined by the user also by a control parameter). When one of these criteria is met, one column of the initial values is replaced by the adjusted values, generated by the best individual of the current generation. Figure 3.2 illustrates the MG-PMA with feedback process.

This replacement of the parameter values by adapted values is done from the first column ($c = 1$ in Figure 3.2) to the last column ($c = m$ in Figure 3.2). After replacing all the columns, the MG-PMA with feedback repeats the process of choosing to update the initial values from the first to the last column.

3.2

Niching Multi-Gene Parameter Mapping Approach

Niching multi-gene parameter mapping approach (Niching MG-PMA) is an MMO method that aims to explore the search space while maintaining population diversity, using MG-PMA with feedback, clearing procedure and local optimization.

Following the concepts of the clearing procedure, Niching MG-PMA uses the idea of dominant individuals in subpopulations. Each subpopulation is formed from individuals that are within the clearing radius and only winners are selected for the application of the genetic operators. In Niching MG-PMA, all individuals who are not winners are removed from the optimization and new individuals are generated from the winners.

Niching MG-PMA is a MA, combining global optimization with local optimization. As explained in the previous chapter, local optimization refines the global search. Just as the update of the initial parameters is done from time to time, the local optimization is also done for each determined number of generations.

3.2.1

Niching MG-PMA Procedure

Algorithm 6 presents the pseudocode of the Niching MG-PMA. As in MG-PMA, Niching MG-PMA begins with the initialization of the initial parameters and GP constants and the configuration of the Niching MG-PMA parameters, the number of trees (k), the frequency parameters for updating the initial parameters (*feedbackFrequency*) and the use of local optimization (*localOptimizationFrequency*).

The division of the initial parameters by the trees is done in the same way as in MG-PMA (function `SplitInitialParameters` presented in Algorithm 4). The population is initialized and the individuals are selected by function `Select` for application of the genetic operators (function `ApplyOperators` in the Algorithm 5) generating new individuals who are evaluated.

While the stopping criterion is not satisfied, the function `Niching` identifies the winners and removes individuals that are within a distance smaller than the clearing radius (relative to dominant individuals) from the population.

If the current generation is multiple of the *localOptimizationFrequency* parameter, the function `LocalOpt` locally optimizes the winners. Then, the algorithm performs the selection and the application of the genetic operators, generating new individuals for the next population. Niching MG-PMA evaluates these individuals and if the current generation is multiple of the *feedbackFrequency* parameter, the function `Feedback` updates the initial parameters. In this update, the

best individual of the current generation replaces one of the columns of the initial parameters.

After the stop criterion is satisfied, the function `CheckSolutions` searches for optimal and sub-optimal solutions among all the winners found during the optimization.

```

Data: Objective Function
Output: Multiple Optimal/Sub-Optimal Solutions
- InitialParameterValues ← Initialize();
- GPConstants ← Initialize();
- Set parameters for Niching MG-PMA;
- Set k, feedbackFrequency, localOptimizationFrequency;
- t ← 0;
- AllIndividuals ← an empty vector;
- SplitInitialParameters(InitialParameters, k);
- Start Population(t);
- Fit ← Evaluate(Population(t));
while Stop criterion ≠ FALSE do
    | t ← t + 1;
    | Population(t) ← Niching(Population(t - 1));
    | if mod(t, localOptimizationFrequency) then
    | | Population(t) ← LocalOptimization(Population(t));
    | end
    | Population(t) ← Select(Population(t));
    | AllIndividuals and Population(t) ← ApplyOperators(Population(t));
    | Fit ← Evaluate(Population(t));
    | if mod(t, feedbackFrequency) then
    | | InitialParameterValues ← Feedback(Population(t));
    | end
end
- MultipleSolutions ← CheckSolutions(AllIndividuals);

```

Algorithm 6: Pseudocode of Niching MG-PMA for a generic objective function.

3.3

Domain Constraint with Periodic Boundary Conditions

Metaheuristics such as GA, PSO and DE naturally follow the domain constraints of the objective function. GP and its variations lose this control, because the input values of the objective function are generated by computer programs. In the PMA, Pujol and Poli use a linear transformation, Equation 3-1, that works for the benchmarks functions tested by them (Pujol, 2008). But this transformation does not work for every type of function, for example when one of the upper or lower limits is equal to zero or in a periodic function with more than one solution.

$$ADAPT = LOWER + OUT \times (UPPER - LOWER) \quad (3-1)$$

where *ADAPT* is the adapted value of the parameter, *LOWER* is the lower limit of the search range, *UPPER* is the upper limit of the search range and *OUT* is the

output of the GP tree (Pujol, 2008).

This thesis proposes another way to normalize the outputs of the GP trees based on the periodic boundary conditions (PBC) used by the molecular modelling (Leach, 2001). This normalization considers that the objective function is inside a box that is replicated in all dimensions. The values generated by the trees that are in an image of the box are put back into the box through multiple sums or subtractions of the box size. Equation 3-2 presents transformation for a value smaller than the lower limit of the objective function and Equation 3-3 for a value greater than the upper limit.

$$N_{BOX} = \frac{|OUT - LOWER|}{SIZE_{BOX}} \quad (3-2)$$

$$ADAPT = UPPER - (N_{BOX} - \lfloor N_{BOX} \rfloor) \times SIZE_{BOX}$$

$$N_{BOX} = \frac{|OUT - UPPER|}{SIZE_{BOX}} \quad (3-3)$$

$$ADAPT = LOWER + (N_{BOX} - \lfloor N_{BOX} \rfloor) \times SIZE_{BOX}$$

where *ADAPT* is the adapted value of the parameter, *LOWER* is the lower limit of the search range, *UPPER* is the upper limit of the search range, *OUT* is the output of the GP tree, *N_{BOX}* is the number of boxes away from the domain and *SIZE_{BOX}* is the box size.

Equations 3-4 and 3-5 present a normalization by PBC for two individuals, $(-15, 0.3)$ and $(-3.2, 107)$ respectively, for a two-dimensional objective function with domain constraint equals $[-5, 5]^2$.

Individual : $(-15, 0.3)$

$$\begin{cases} OUT = -15 \\ LOWER = -5 \\ UPPER = 5 \\ SIZE_{BOX} = 10 \end{cases} \quad (3-4)$$

OUT < *LOWER* :

$$N_{BOX} = \frac{|OUT - LOWER|}{SIZE_{BOX}} = \frac{|-15 - (-5)|}{10} = 1$$

$$ADAPT = UPPER - (N_{BOX} - \lfloor N_{BOX} \rfloor) \times SIZE_{BOX}$$

$$ADAPT = 5 - (1 - \lfloor 1 \rfloor) \times 10 = 5$$

Individual_{adapted} : $(5, 0.3)$

Individual : $(-3.2, 107.6)$

$$\begin{cases} OUT = 107.6 \\ LOWER = -5 \\ UPPER = 5 \\ SIZE_{BOX} = 10 \end{cases} \quad (3-5)$$

$OUT > UPPER :$

$$N_{BOX} = \frac{|OUT - UPPER|}{SIZE_{BOX}} = \frac{|107.6 - 5|}{10} = 10.26$$

$$ADAPT = LOWER + (N_{BOX} - \lfloor N_{BOX} \rfloor) \times SIZE_{BOX}$$

$$ADAPT = -5 + (10.26 - \lfloor 10.26 \rfloor) \times 10 = -2.4$$

Individual_{adapted} : $(-3.2, -2.4)$

Figure 3.3 shows an example of the normalization based on the PBC applied to a dot outside of the domain constraints of a multimodal function. Figure 3.3.a shows this dot outside the objective function domain (green dot) and in Figure 3.3.b it is put back into the domain (black dot) from the replicated images of this function.

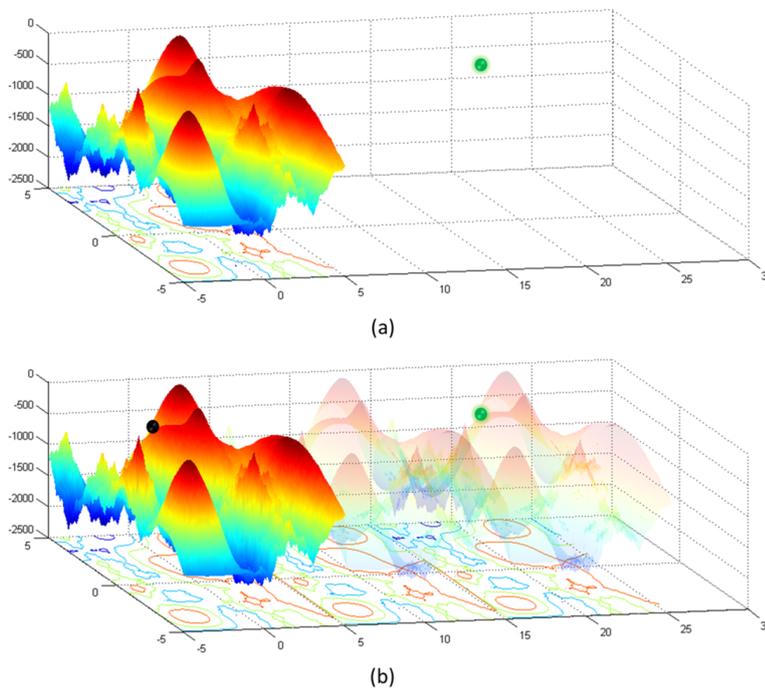


Figure 3.3: Example of the normalization based on the PBC applied to an output (green dot) of a GP tree outside of the domain constraints. The linear transformation replicates images of the box and identifies where the outside dot should be (black dot). The domain constraints determine the size of the box.

As the domain constraint with PBC is applied to the output value of the GP trees, this normalization is done directly at the input values of the objective function. Thus, the domain constraint with PBC can be used in other metaheuristic methods, whenever the problem is a constrained optimization.

4

Analyses of the Multigenic Approach

4.1

Analyses of Multigenic Individuals

Given the multi-gene nature of MG-PMA, this section describes the tests performed to analyze the effects of the number of trees k and their depths d on the optimization of different types of objective functions. The following setups were considered in order to evaluate the effect of varying the number of trees:

- $k = 1$ – the case where MG-PMA reduces to PMA;
- $k = n/2$ – an intermediate value of k , where n is the number of parameters to be optimized;
- $k = n$ – the most general form of MG-PMA, where each parameter is represented by a tree.

The maximum tree depths, d , considered were 2, 3, 5, 7 and 9, where trees with lower depth tend to embody simpler equations than trees with larger depths. Clearly, increasing d also increases the computational cost of optimization.

Two sets of benchmark functions were employed in the present work. First, the same benchmark functions and criteria from Pujol and Poli work (Pujol, 2008) were used (Table 4.1). Subsequently, the method was also tested using benchmark functions of CEC'15 competition (Liang, 2014) (Table 4.2).

Considering the benchmark functions in Table 4.1, the search for the minimum was interrupted when the fitness of the best individual was lower than the convergence criterion, which where: 10^{-6} for functions f_1 - f_3 and f_5 - f_6 ; 10^{-2} for the Schwefel's functions f_8 - f_{12} ; and $f < \bar{v}$ for f_4 , where $v \in [0, 1)$ is a noise term defined by a random variable with a uniform distribution, and expected value $\bar{v} = 0.5$. The latter is the De Jong 4 function and was evaluated on a statistical basis. Fitness was computed as $|f(\text{MG-PMA}_i(X_i)) - \bar{v}|$, and the search was interrupted when $f < \bar{v}$ (Pujol, 2008).

Considering the experiments using the CEC'2015 benchmark functions (Table 4.2), the search for the minimum was interrupted when the fitness of the best individual was lower than 10^{-8} , as established by the CEC'2015 competition (Liang, 2014).

Table 4.1: Benchmark functions from (Pujol, 2008). The table presents the name of each function, the number of parameters to be optimized and the global minimum. The Schwefel's function was performed with different number of parameters in order to analyze each method according to the dimensionality of the problem.

Abbreviation	Function Name	Number of Parameters	Global Minimum
f_1	De Jong 1	3	0
f_2	De Jong 2	2	0
f_3	De Jong 3	5	0
f_4	De Jong 4	30	$0 + \bar{v}^*$
f_5	De Jong 5	25	0.998004
f_6	Corona's parabola	4	0
f_7	Griewangk's function	10	0
f_8	Schwefel's function	10	$\approx -4,189.83$
f_9	Schwefel's function	30	$\approx -12,569.49$
f_{10}	Schwefel's function	50	$\approx -20,949.14$
f_{11}	Schwefel's function	70	$\approx -29,328.80$
f_{12}	Schwefel's function	100	$\approx -41,898.29$

* $\bar{v} \in [0, 1]$ is a noise term defined by a random variable with a $\in [0, 1]$ uniform distribution.

Table 4.2: Benchmark functions proposed for CEC'2015 competition. The table presents the name of each function, the number of the parameters to be optimized and the global minimum.

Abbreviation	Function Name	Number of Parameters	Global Minimum
c_1	Rotated High Conditioned Elliptic Function	10	100
c_2	Rotated Cigar Function	10	200
c_3	Shifted and Rotated Ackley's Function	10	300
c_4	Shifted and Rotated Rastrigin's Function	10	400
c_5	Shifted and Rotated Schwefel's Function	10	500
c_6	Hybrid Function 1 ($N = 3$)	10	600
c_7	Hybrid Function 2 ($N = 4$)	10	700
c_8	Hybrid Function 3 ($N = 5$)	10	800
c_9	Composition Function 1 ($N = 3$)	10	900
c_{10}	Composition Function 2 ($N = 3$)	10	1,000
c_{11}	Composition Function 3 ($N = 5$)	10	1,100
c_{12}	Composition Function 4 ($N = 5$)	10	1,200
c_{13}	Composition Function 5 ($N = 5$)	10	1,300
c_{14}	Composition Function 6 ($N = 7$)	10	1,400
c_{15}	Composition Function 7 ($N = 10$)	10	1,500

Based on the tests performed by Pujol and Poli (Pujol, 2008) and the CEC'2015 competition (Liang, 2014), three test sets were established. The first set consisted in the optimization of benchmark functions described in Table 4.1 and using the MG-PMA parameters described in Table 4.3 following the parameter values defined by Pujol and Poli (Pujol, 2008). The second set also used the same benchmark functions described in Table 4.1 and the parameters described

in Table 4.3, but a random shift was introduced in all input parameters for each benchmark function. Note that, in this second test, each function $f(x_1, \dots, x_i, \dots, x_n)$ was considered as $f(x_1 - r_1, \dots, x_i - r_i, \dots, x_n - r_n)$ where r is an uniformly distributed random constant in the range $(-1, +1)$, generated at the start of each run. As a result, the optimal values are not 0's and 1's, they were shifted randomly. As performed by Pujol and Poli (Pujol, 2008), it was established one hundred independent runs of the algorithms for each benchmark function and for both experimental sets, with the maximum number of evaluations before time-out equal to 2×10^6 . The third test set relied on the CEC'2015 competition benchmark functions, described in Table 4.2. For this test, the population and tournament size equal to 100 for all setups and runs (to follow the parameterization of most benchmark functions of test sets 1 and 2). As established by the CEC'2015 competition, it was performed 51 independent runs on test set 3 for each benchmark function and the maximum number of evaluations before time-out was 10^5 . For all test sets, the mathematical functions used on the MG-PMA were *plus*, *minus*, *times* and *protected division* (for $a = b/c$, if $c = 0$ then $a = 0$), and the remaining configuration of MG-PMA is described in Table 4.4.

Table 4.3: Parameters used in the test sets 1 and 2.

Benchmark Function	Population Size	Tournament Size
f_1	100	100
f_2	50	50
f_3	50	50
f_4	100	100
f_5	50	50
f_6	200	100
f_7	100	100
f_8	100	100
f_9	100	100
f_{10}	100	100
f_{11}	100	100
f_{12}	100	100

MG-PMA was implemented in the software MATLAB R2016b (MATLAB, 2016), using elements of GPTIPS 2 (Genetic Programming & Symbolic Data Mining Platform for MATLAB) library of MGGP (Searson, 2015). This library allows the use of more than one mutation operator during optimization. As described in Table 4.4, the probability of applying the mutation operator in the experiments was 30% (value defined through preliminary analyzes presenting low sensitivity). From the moment at which the mutation operator was used, it was possible to perform one of the three types of mutation (Searson, 2015) with the following probabilities: ordinary sub-tree mutation (90%), as described in

Table 4.4: General configuration of MG-PMA.

Parameters	Values
Decimal ERC* rate	90%
Integer ERC* rate	0%
ERC* range	$[-1, +1]$
Low-level crossover rate	65%
Mutation rate	30%
Direct reproduction rate	5%
High-level crossover rate	50%
Elitism rate	1%
Lexicographic pressure (Luke, 2002)	Yes

* ERC - Ephemeral Random Constant: constants generated in a specific range by the user to constitute the trees of each individual together with variables (from the initial parameter values to be mapped by MG-PMA).

Figure 2.7; switch an input terminal to another randomly selected input terminal (5%); and Gaussian perturbation of a randomly selected constant (5%, with the standard deviation of the Gaussian used equal to 10%) – the parameter values for these three types of mutation are default values of the GPTIPS 2 library (Searson, 2015). The initial population was generated using the ramped half-and-half method (Koza, 1992).

4.1.1 Results of Analyses of Multigenic Individuals

Test Set 1

In this section, the results for the optimization of functions f_1 - f_{12} (described in Table 4.1) are discussed. Results are reported numerically in the Appendix A (Tables A1-A5). For almost all cases, the number of hits on the function global minimum is equal to 100, but for the benchmark function f_2 only experiments with $k = 1$ and one experiment with $k = n$ were able to find the global minimum. Noting that the dimension of benchmark function f_2 is equal to 2, there is no experiment with $k = n/2$.

In order to provide a more clear picture of the effects of varying the number k and depth d of trees on the mean number of evaluations (MNE), graphical analysis were performed and depicted in Figures 4.1-4.3.

Figure 4.1 presents the variation of MNE as a function of d for their different values of k . Considering functions f_1 , f_2 and f_4 - f_7 , MG-PMA with $k = 1$ performs significantly better (on average) compared to configurations with $k = n/2$ and $k = n$. However, for functions f_3 and f_8 - f_{12} , the differences are more subtle, indicating that configurations with $k = n/2$ and $k = n$ become, comparatively, more effective upon increasing complexity. Overall, results improve upon increasing d . Considering configurations with d about 7-9 and functions of higher dimensionality (f_9 - f_{12}),

the setup with $k = n$ appears to be the most effective. As discussed earlier, f_2 only yielded results when $k = 1$, and one hit when $k = n$.

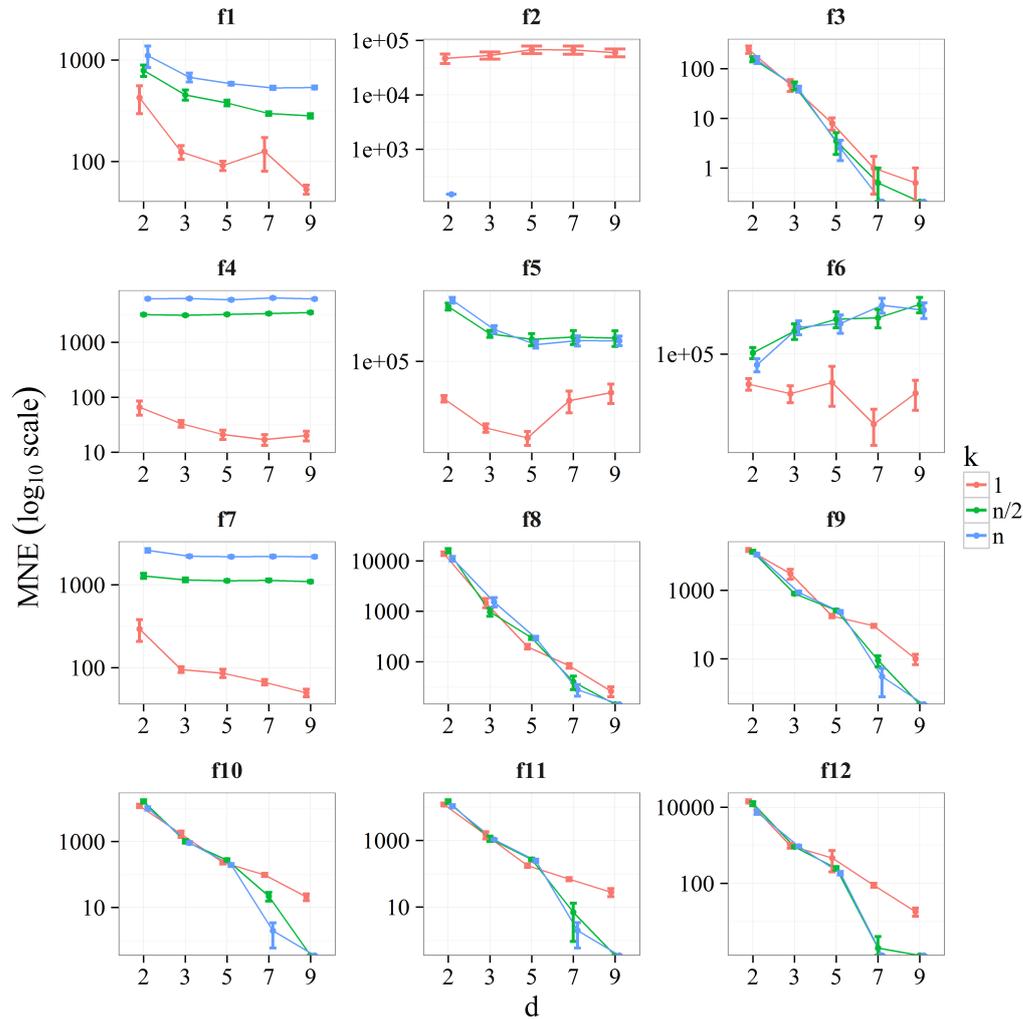


Figure 4.1: The mean number of evaluations (MNE) on logarithmic scale is plotted as a function of the tree depth d for the 12 functions of test set 1. The corresponding standard errors are represented by vertical bars. Each line color refers to a specific number of trees k (red, $k = 1$; green, $k = n/2$; blue, $k = n$).

Figure 4.2 shows a box plot graphic of MNE as a function of k and considering, for each k , an aggregation of all d values. In other words, this analysis reveals the effect of k on MNE, irrespective of d . Considering functions f_1 and f_4 - f_7 , the configuration with $k = 1$ performs better on average (black dots), and yields quantiles lower than the other configurations. Considering functions f_3 , f_8 - f_{12} , the differences are not obvious, although it seems that the configuration with $k = n$ performs better. An interesting remark on configuration with $k = n/2$ is that it, almost always, approaches the results with $k = n$.

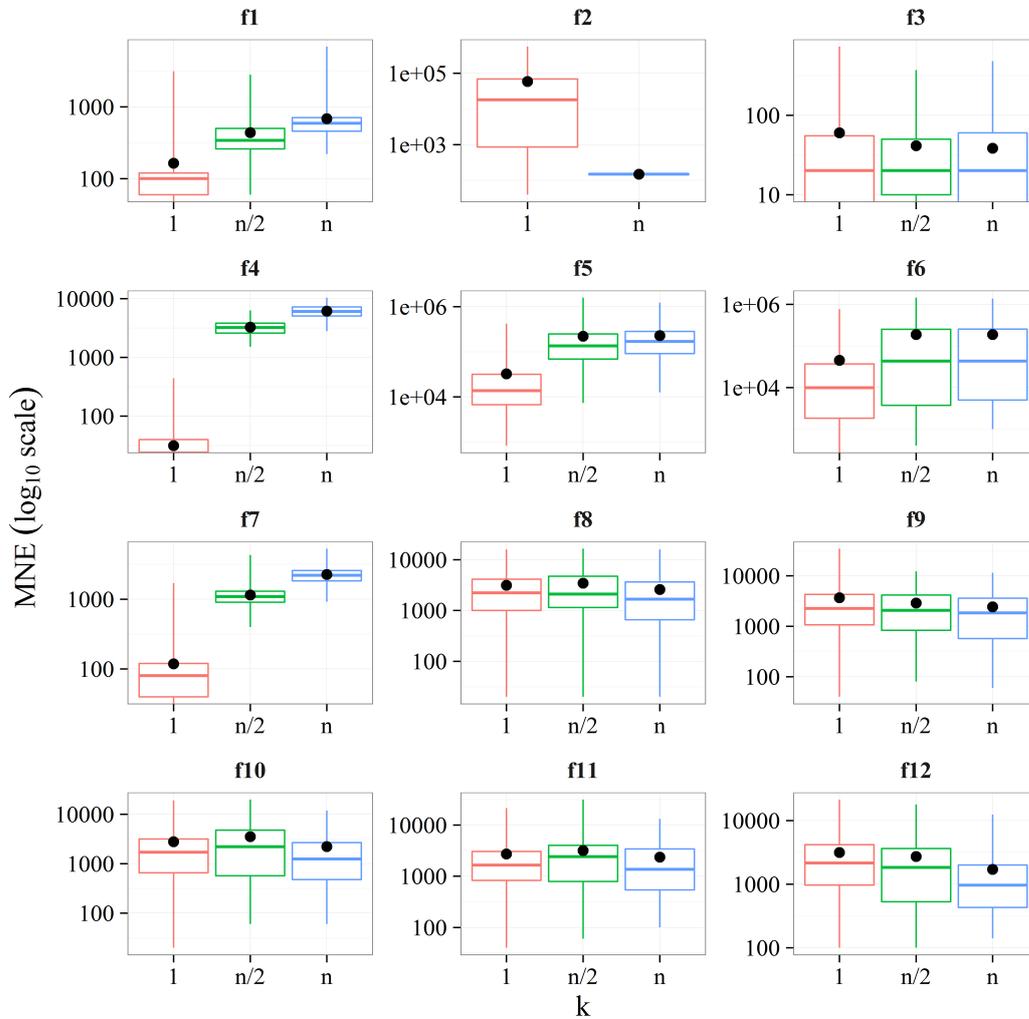


Figure 4.2: Box plot graphic showing the mean number of evaluations (MNE) for the test set 1 as a function of the number of trees k considering, for each k , an aggregation of all d values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot.

Figure 4.3 shows a box plot graphic of MNE as a function of d and considering, for each d , an aggregation of all k values. As opposed to the previous analysis, this one reveals the effect of d on MNE, irrespective of k . For most cases (f_1 , f_3 , f_5 , and f_8 - f_{12}), MNE decreases upon increasing d . Interestingly, for functions f_2 , f_4 , f_6 and f_7 , a change in d appears to have no effect on MNE.

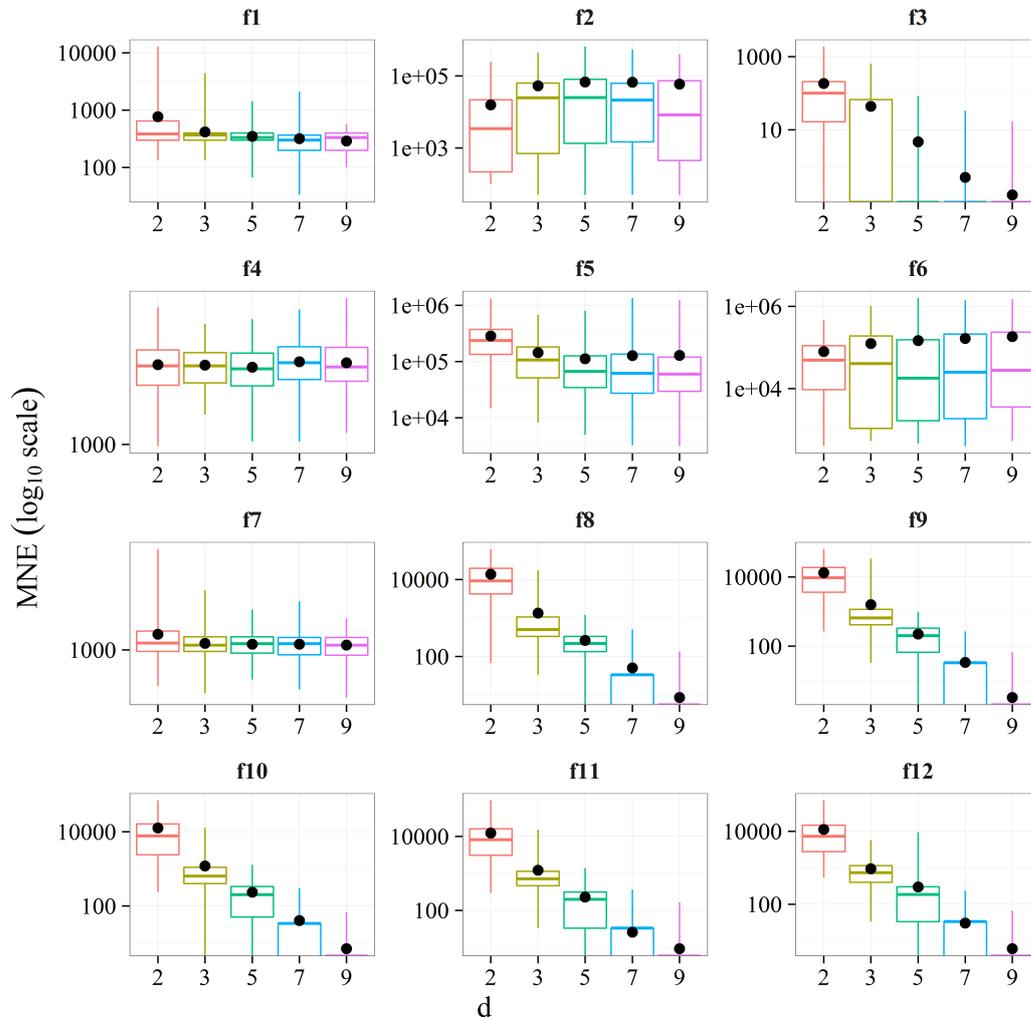


Figure 4.3: Box plot graphic showing the mean number of the evaluations (MNE) for the test set 1 as a function of the tree depth d considering, for each d , an aggregation of all k values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot.

Tables 4.5 and 4.6 show the results of Aligned Friedman's (Derrac, 2011) and Iman-Davenport's (Iman, 1980) tests, along with the Holm's (Holm, 1979) procedure based on MG-PMA performance in the test set 1 for all values of k and d considered (Tables A1-A5). These tests were carried out using the KEEL software

(Alcalá-Fdez, 2011). It can be seen that the average rank of MG-PMA with $k = 1$ and $d = 9$ was the lowest (43.318) of all configurations tested.

Table 4.5: Average rankings of Aligned Friedman's and Iman-Davenport's tests for comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 1.

i	Algorithm	Rank
14	MG-PMA with $k = n/2$ and $d = 2$	124.546
13	MG-PMA with $k = n$ and $d = 2$	124.000
12	MG-PMA with $k = n$ and $d = 3$	103.818
11	MG-PMA with $k = 1$ and $d = 2$	96.364
10	MG-PMA with $k = n/2$ and $d = 3$	95.636
9	MG-PMA with $k = n/2$ and $d = 5$	87.546
8	MG-PMA with $k = n$ and $d = 7$	85.364
7	MG-PMA with $k = n$ and $d = 9$	84.455
6	MG-PMA with $k = n/2$ and $d = 7$	81.318
5	MG-PMA with $k = n$ and $d = 5$	80.727
4	MG-PMA with $k = n/2$ and $d = 9$	80.000
3	MG-PMA with $k = 1$ and $d = 3$	64.546
2	MG-PMA with $k = 1$ and $d = 5$	48.091
1	MG-PMA with $k = 1$ and $d = 7$	45.273
0	MG-PMA with $k = 1$ and $d = 9$	43.318

Table 4.6: Holm's procedure for pairwise comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 1. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = 1$ and $d = 9$.

Test	p -value				
Aligned Friedman	0.757				
Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?	
MG-PMA with $k = n/2$ and $d = 2$	3.987	0.000	0.004	Yes	
MG-PMA with $k = n$ and $d = 2$	3.961	0.000	0.004	Yes	
MG-PMA with $k = n$ and $d = 3$	2.970	0.003	0.004	Yes	
MG-PMA with $k = 1$ and $d = 2$	2.604	0.009	0.005	No	
MG-PMA with $k = n/2$ and $d = 3$	2.568	0.010	0.005	No	
MG-PMA with $k = n/2$ and $d = 5$	2.171	0.030	0.006	No	
MG-PMA with $k = n$ and $d = 7$	2.064	0.039	0.006	No	
MG-PMA with $k = n$ and $d = 9$	2.019	0.044	0.007	No	
MG-PMA with $k = n/2$ and $d = 7$	1.865	0.062	0.008	No	
MG-PMA with $k = n$ and $d = 5$	1.836	0.066	0.010	No	
MG-PMA with $k = n/2$ and $d = 9$	1.801	0.072	0.013	No	
MG-PMA with $k = 1$ and $d = 3$	1.042	0.297	0.017	No	
MG-PMA with $k = 1$ and $d = 5$	0.234	0.815	0.025	No	
MG-PMA with $k = 1$ and $d = 7$	0.096	0.924	0.05	No	

Typically, in most statistical analyzes, the value of 0.05 for p -value is used as the threshold for significance (Derrac, 2011). If the p -value is less than 0.05,

the null hypothesis that there is no difference between the configurations tested is rejected and it is concluded that there is a significant difference between these configurations. If the p -value is greater than 0.05, it is not possible to conclude that there is a significant difference.

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value greater than 0.05 (p -value = 0.757), so it is not possible to conclude that there is a significant difference and only the rank positions can be checked. As the average rank of MG-PMA with $k = 1$ and $d = 9$ is the lowest, it was selected as the control model for the Holm's test. In Table 4.6, z represents the test statistic for Holm's test, where the ranks R_i of all models are compared to that of the control model R_0 , normalized by the standard error SE . This is used to compute the p -value, which is compared to the Holm level. If the p -value is lower than that level, the hypothesis of equality of ranks is rejected. It can be seen that MG-PMA with $k = 1$ and $d = 9$ was ranked significantly better than MG-PMA with: (i) $k = n/2$ and $d = 2$; (ii) $k = n$ and $d = 2$; and (iii) $k = n$ and $d = 3$. Nothing can be said in respect to the comparison between MG-PMA with $k = 1$ and $d = 9$ and the other algorithms statistically yielding better or worse results.

It is also important to note that, in general, the MG-PMA algorithms configured with $k = 1$ were better in the Table 4.5 ranking than the configurations with $k = n/2$ and $k = n$. This is due to the low complexity of the benchmark functions of test set 1. About the tree depths, the higher the value of d the better the position of the algorithm in the ranking.

Test Set 2

In this section, the results for the optimization of functions f_1 - f_{12} (described in Table 4.1) in which random shifts in all parameters are applied (see Section 4.1) are discussed. The analyses for the current test set are similar to those in test set 1, and are reported numerically in the Appendix A (Tables A6-A10) and are depicted in Figures 4.4-4.6. From a quick inspection of Tables A6-A10, it is possible to observe that the number of hits on the function global minimum increases upon increasing k , which suggests a different behavior if compared to test set 1. Note also that, for test set 2, it was more difficult to find the global minimum for some benchmark functions.

Figure 4.4 presents the variation of MNE as a function of d for all values of k . Considering functions f_1 , f_4 and f_7 , MG-PMA with $k = n$ performs significantly better compared to configurations with $k = 1$ and $k = n/2$. However, interestingly, for function f_2 , hits are only seen with $k = 1$. Functions f_3 and f_8 show no clear difference in terms of k , all of them showing a consistent decrease for increasing values of d . Function f_5 is best handled by a combination of $k = 1$ and $d = 2$, but

besides this point, $k = n$ seems to be the best choice for increasing values of d . Functions f_9 - f_{12} show that, for $d \leq 5$, no clear distinction between k values are observed. However, for $d > 5$, experiments with $k = n$ and $k = n/2$ significantly outperform those with $k = 1$. Note that, for function f_6 , not a single hit on the global minimum was achieved.

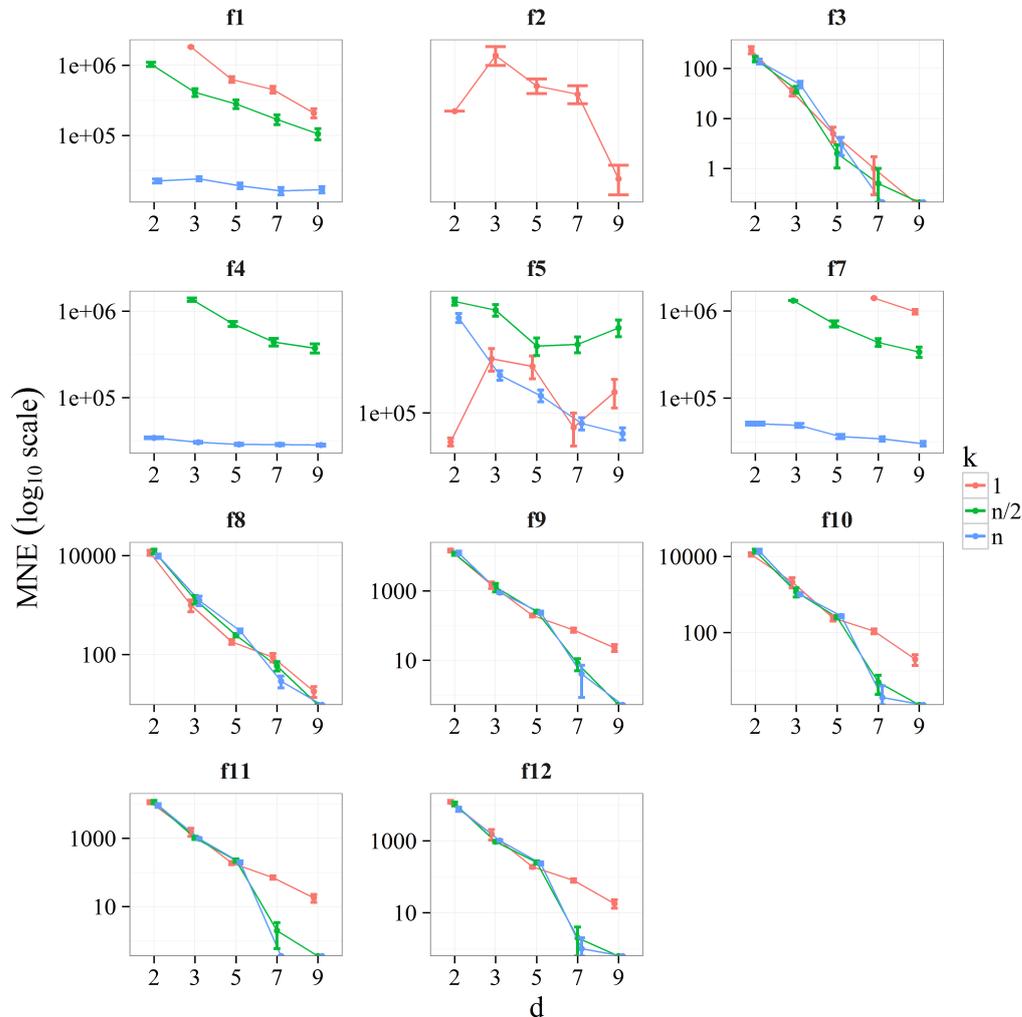


Figure 4.4: The mean number of evaluations (MNE) on logarithmic scale is plotted as a function of the tree depth d for the 12 functions of test set 2. The corresponding standard errors are represented by vertical bars. Each line color refers to a specific number of trees k (red, $k = 1$; green, $k = n/2$; blue, $k = n$).

Figure 4.5 shows a box plot graphic of MNE as a function of k considering, for each k , an aggregation of all d values. It can be seen that, regardless of functions f_1 , f_4 and f_7 , where $k = n$ leads to significantly better results compared to $k = 1$ and $k = n/2$, the aggregation in terms of d makes the differences more subtle.

Figure 4.6 presents a box plot graphic of MNE as a function of d considering,

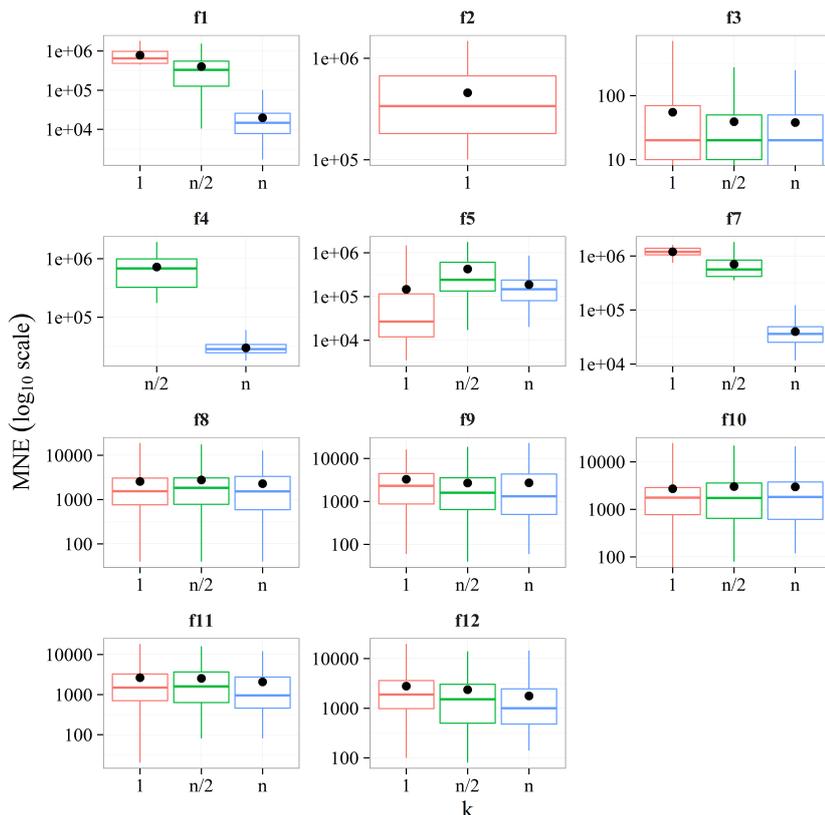


Figure 4.5: Box plot graphic showing the mean number of evaluations (MNE) for the test set 2 as a function of the number of trees k considering, for each k , an aggregation of all d values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot.

for each d , an aggregation of all k values. The pattern is similar to the corresponding figure of set 1 (Figure 4.3), indicating that MNE decreases upon increasing d , irrespective of k . This is probably due to the increase in complexity that a tree can entail upon increasing d , enabling a better coverage of a large search space.

Functions f_2 and f_7 have smaller values for $d = 2$, indicating that this would be the best configuration for these functions. However, this exception occurs for these functions because not all configurations find the global minimum, especially for $d = 2$, decreasing the value of the MNE (Tables A6-A10). As the value of d increases, the number of HITS increases, improving the graph data of these two functions and confirming the effect observed in the other functions (MNE decreases upon increasing d , irrespective of k).

Tables A11 and A12 show the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on MG-PMA performance in the test set 2 for all values of k and d considered (Tables A6-A10). It can be seen that the average rank of MG-PMA with $k = n$ and $d = 9$ was the lowest

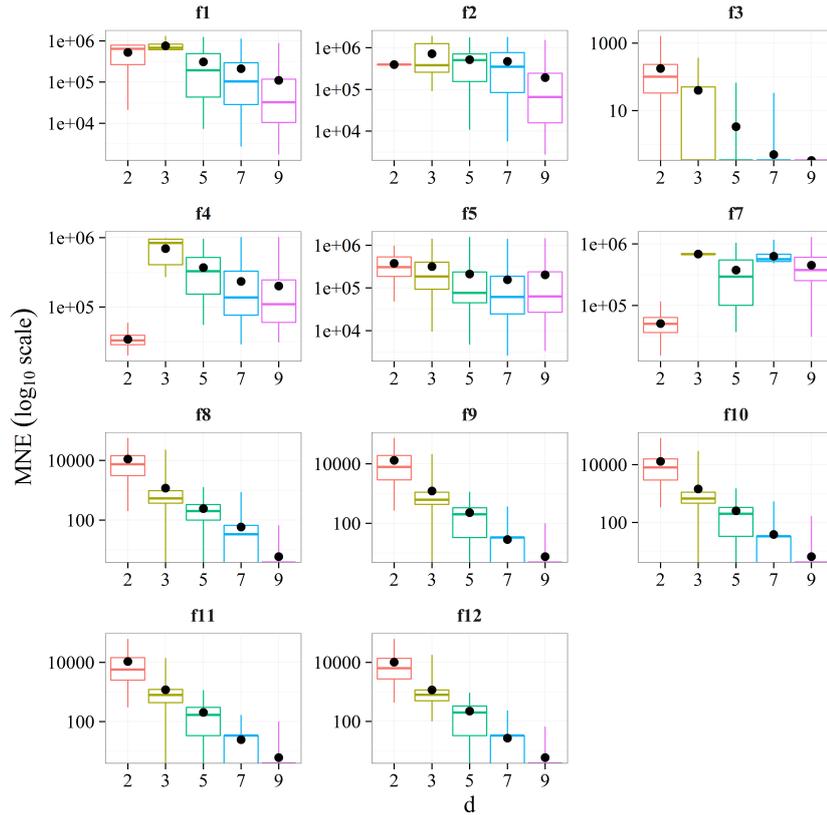


Figure 4.6: Box plot graphic showing the mean number of the evaluations (MNE) for the test set 2 as a function of the tree depth d considering, for each d , an aggregation of all k values. This graphic displays the distribution of the MNE based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot.

(45.950) of all configurations tested.

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value greater than 0.05 (p -value = 0.839), so it is not possible to conclude that there is a significant difference and only the rank positions can be checked. As the average rank of MG-PMA with $k = n$ and $d = 9$ is the lowest, it was selected as the control model for the Holm's test and it performed substantially better than MG-PMA with: (i) $k = n/2$ and $d = 2$; (ii) $k = 1$ and $d = 2$; (iii) $k = 1$ and $d = 3$; and (iv) $k = n/2$ and $d = 3$. Nothing can be said in respect to the comparison between MG-PMA with $k = n$ and $d = 9$ and the other algorithms statistically yielding better or worse results.

It is also important to note that, for the benchmark functions of the test set 2 the higher the values of k and d the better the position of the algorithm in the ranking. Noting that for a test set more complex than test set 1, an approach with MG-PMA using more trees ($k > 1$) results in better solutions.

Test Set 3

Test set 3 considers the CEC'2015 benchmark functions (Liang, 2014) (Table 4.2) as described in Section 4.1. The results regarding this set are shown in the Appendix A (Tables A13-A17). Appendix B presents the properties of the CEC'2015 benchmark functions and their 3-D maps for the 2-D functions available in the problem definition for the CEC'2015 competition (Liang, 2014).

Figure 4.7 presents the variation of the mean of the best fitness values (MBFV) as a function of d for all values of k . In general, MG-PMA with $k = n$ have lower average MBFV compared to other configurations. However, this difference cannot be considered as statistically significant (note the overlap of the error bars). The most pronounced differences occur, to a large extent, for functions c_2 , c_{10} and, to a minor extent, for c_1 , c_6 , and c_8 . One can still notice the dependence upon d (results usually improve upon increasing d), but, for certain functions (c_3 , c_4 , c_5 , c_9 , c_{12} , and c_{13}), this is not observed.

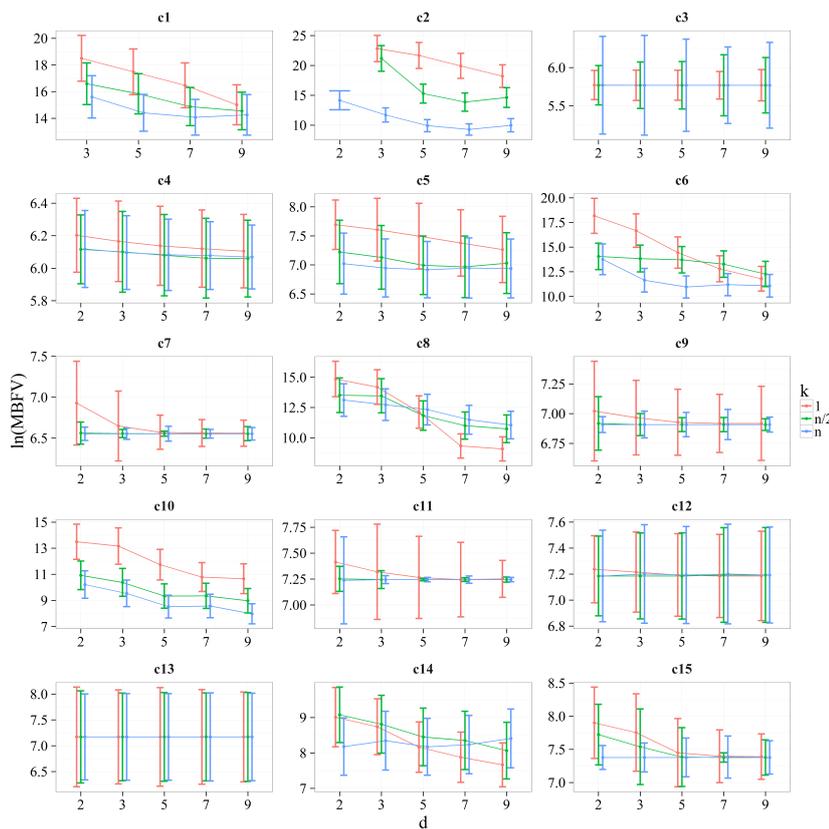


Figure 4.7: The mean of the best fitness values (MBFV) on logarithmic scale is plotted as a function of the tree depth d for the 15 functions of test set 3. The corresponding standard errors are represented by vertical bars. Each line color refers to a specific number of trees k (red, $k = 1$; green, $k = n/2$; blue, $k = n$).

Figure 4.8 shows a box plot graphic of MBFV as a function of k and

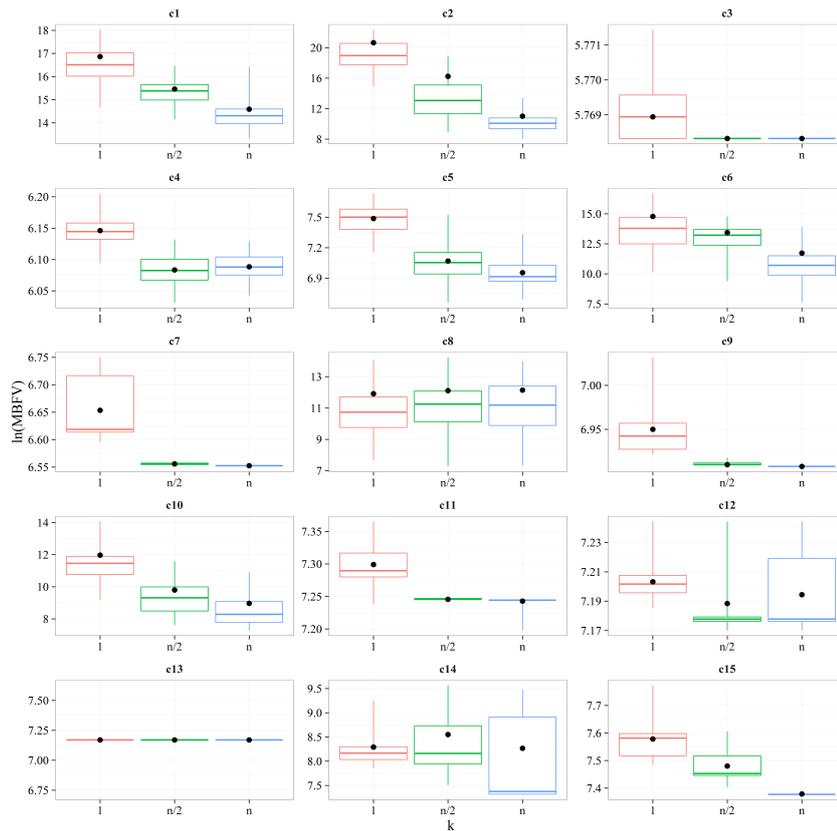


Figure 4.8: Box plot graphic showing the mean of the best fitness values (MBFV) for the test set 3 as a function of the number of trees k considering, for each k , an aggregation of all d values. This graphic displays the distribution of the MBFV based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot.

considering, for each k , an aggregation of all d values. The results show that the configuration with $k = n$ performs, on average, better compared to $k = 1$. This is most evident for functions c_1 - c_6 , c_7 , c_9 - c_{11} , c_{13} , and c_{15} . In most cases, the quantiles of $k = n$ are lower compared to $k = 1$. For some functions (c_3 - c_5 , c_7 , c_9 , c_{11} , and c_{12}), $k = n/2$ was almost as good as (or even better than) $k = n$. These results indicate that finding an optimal k might yield better results for certain optimization problems.

Figure 4.9 shows a box plot graphic of MBFV as a function of d considering, for each d , an aggregation of all k values. Similarly to the other test sets, there is a clear trend indicating that MBFV decreases upon increasing d . This is more evident for certain functions (c_1 , c_3 , c_4 , c_7 , c_9 , c_{10} , and c_{15}), and less evident for others (c_2 , c_5 , c_{12} , and c_{13}).

Results were also aggregated according to the type (unimodal, multimodal and hybrid; see Table 4.2) of the 15 functions considered. Overall, Figure 4.10 shows that for unimodal and multimodal functions the best configuration is $k = n$

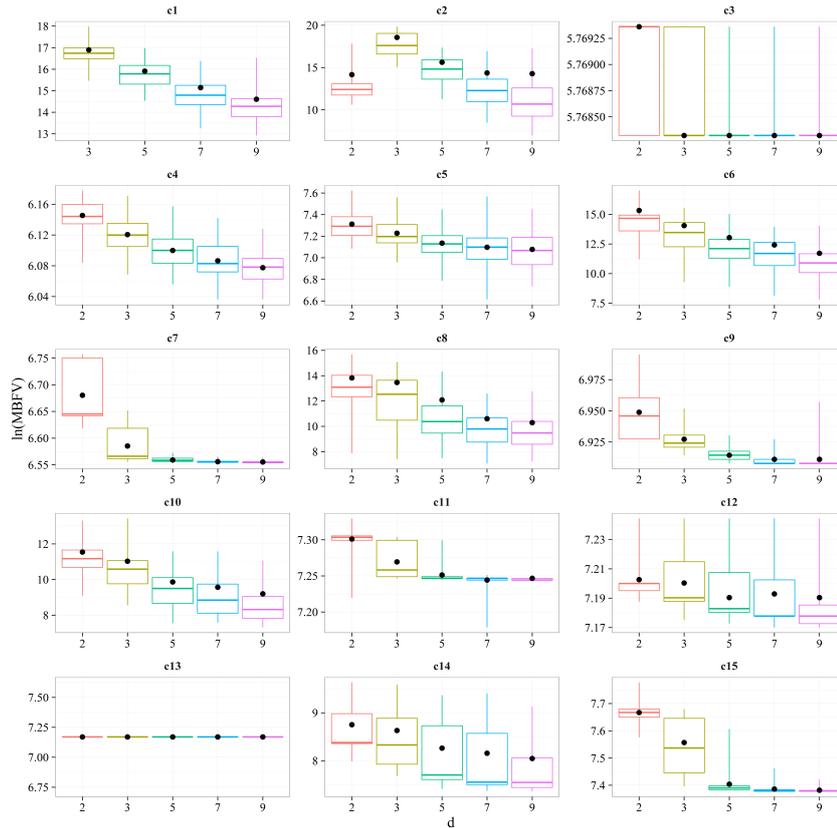


Figure 4.9: Box plot graphic showing the mean of the best fitness values (MBFV) for the test set 3 as a function of the tree depth d considering, for each d , an aggregation of all k values. This graphic displays the distribution of the MBFV based on the minimum, first quartile, median, third quartile and maximum values. The average value is also represented by a black dot.

for all tree depths considered. In the case of hybrid functions, $k = n$ provides better results for lower values of d . However, as d increases, the differences between configurations in terms of k becomes subtle.

Tables A18 and A19 show the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on MG-PMA performance in the test set 3 for all values of k and d considered (Tables A13-A17). It can be seen that the average rank of MG-PMA with $k = n/2$ and $d = 9$ was the lowest (82.667) of all configurations tested.

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value greater than 0.05 (p -value = 0.535), so it is not possible to conclude that there is a significant difference and only the rank positions can be checked. As the average rank of MG-PMA with $k = n/2$ and $d = 9$ is the lowest, it was selected as the control model for the Holm's test and it performed substantially better than MG-PMA with: (i) $k = 1$ and $d = 2$; and (ii) $k = 1$ and $d = 3$. Nothing can be said in respect to the comparison between MG-PMA with $k = n/2$ and $d = 9$ and the other algorithms

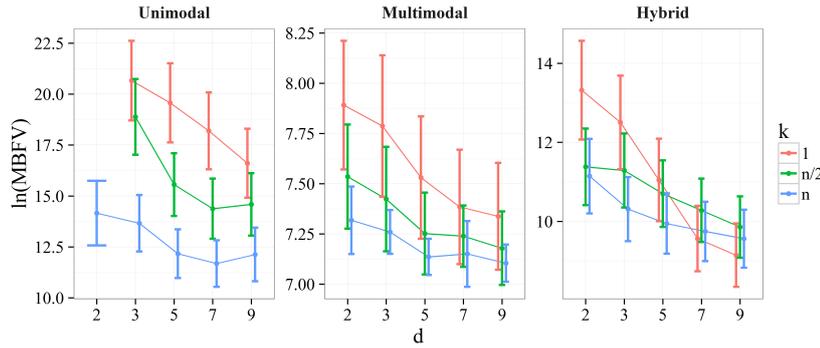


Figure 4.10: Variation of the mean of the best fitness values (MBFV) as a function of the tree depth d . Values were aggregated according to the nature (unimodal, multimodal or hybrid) of each function (c_1 - c_{15}) of the CEC benchmark (test set 3).

statistically yielding better or worse results.

It is also important to note that, for the benchmark functions of the test set 3 the higher the values of k and d the better the position of the algorithm in the ranking. As in the analysis of the results of test set 2, the approaches with MG-PMA using more trees ($k > 1$) results in better solutions.

4.2 MG-PMA with Feedback

Given the dynamic character of MG-PMA with feedback, this section describes the analysis on adapting the initial values to be mapped. This new variation was compared to MG-PMA using the same number of trees (k) shown in the previous optimizations. These values continued to be analyzed, since for problems of different complexities, different values of k may be more appropriate. The tree maximum depth d considered was 7, since there is no significant difference with $d = 9$ (as can be seen in Table A19), but the computational time for the former one was significantly lower.

The tests with MG-PMA with feedback algorithm were performed using the CEC'2015 competition benchmark functions (Table 4.2). The same parameters and configurations (Table 4.4, maximum number of evaluations before time-out equal to 10^5 and mathematical functions – *plus*, *minus*, *times* and *protected division*) of the test set 3 were used.

4.2.1 Results of MG-PMA with Feedback

Table A22 shows the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the performance of MG-PMA and

MG-PMA with feedback methods (Tables A20 and A21). It can be seen that the average rank of MG-PMA with $k = n$ was the lowest (30.167) of all algorithms tested and MG-PMA with $k = n/2$ was second (30.633).

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value less than 0.05 (p -value = 0.025), the hypothesis that there is no difference between the configurations tested is rejected. As the average rank of MG-PMA with $k = n$ is the lowest, it was selected as the control model for the Holm's test and it was ranked significantly better than MG-PMA with $k = n/2$ with feedback, and $k = 1$ with feedback. Nothing can be said in respect to the comparison between MG-PMA with $k = n$ and the other algorithms statistically yielding better or worse results.

The MG-PMA approach with feedback could be useful to avoid the evolution process to get stuck in local minimum (or maximum) due to the generation of a new mapping scheme. However, the comparison between this approach with $k = n$ and MG-PMA (without feedback) for the CEC'2015 benchmark functions didn't find significantly difference. Therefore, future tests should be performed to validate the potential of this approach.

4.3 Analysis of Tournament Size and Crossover

As an attempt to further improve the settings involved in MG-PMA, this section aims at analyzing different tournament sizes and other high-level crossover operators. In Section 4.1, the size of the tournament used for test set 3 was equal to the population size (100), since Pujol and Poli have used this approach in almost all the benchmark functions tested by them (Pujol, 2008)¹. As a way to increase the probabilistic effect of the selection process, the tournament size has been changed to 50, 25 and 2.

The high-level crossover used in Section 4.1 was provided by MGGP GPTIPS 2 library (Searson, 2015). This section compares this operator with the high-level two-point crossover from the first version of MGGP GPTIPS library (Searson, 2010) and three other multigenic crossover operators: (i) restricted high-level crossover; (ii) high-level single-point crossover; and (iii) high-level uniform crossover.

The high-level operator available in the GPTIPS 2 library allows the crossover between an individual and itself resulting in low genetic diversity throughout evolution. In this way the restricted high-level crossover operator is equal to

¹The original PMA uses the non-destructive headless chicken crossover (Angeline, 1997). This operator performs the standard crossover between an individual selected from the population and a randomly created individual. The resulting offspring is inserted in the population if it is superior to the parent, otherwise it is discarded. This procedure is repeated a number of times equal to the size of the population (Pujol, 2008)

the high-level crossover operator, but does not allow the crossover between an individual and itself. The other crossover operators were presented in Chapter 2.

The results were compared to MG-PMA using the same number of trees (k) used in the previous simulations and the maximum tree depth d considered was 7. The change of the tournament size and the exchange of the crossover operator were performed not considering all configurations due to the large number of possibilities. Thus, when the tournament size was changed, the crossover operator used was the same as that of the Section 4.1 (high-level crossover operator) and when the crossover operator was changed, the size of the tournament used was equal to 100.

The simulations of this section were performed using the CEC'2015 competition benchmark functions (Table 4.2). The same parameters and configurations (Table 4.4, maximum number of evaluations before time-out equal to 10^5 and mathematical functions – *plus*, *minus*, *times* and *protected division*) of the test set 3 were used.

4.3.1

Results of Analysis of Tournament Size and Crossover

Tables A30 and A31 show the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the accuracy of MG-PMA with variations of k (Table A20), tournament size (Tables A23-A25) and high-level crossover operators (Tables A26-A29). It can be seen that the average rank of MG-PMA with $k = n$, tournament size equal to 2 and high-level crossover operator was the lowest (100.433) of all algorithms tested and MG-PMA with $k = n/2$, tournament size equal to 2 and high-level crossover operator in second place (117.067). The tournament size equal to 2 is the best configuration since it allows greater genetic diversity throughout the evolution.

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value greater than 0.05 (p -value = 0.925), so it is not possible to conclude that there is a significant difference and only the rank positions can be checked. As the average rank of MG-PMA with $k = n$, tournament size equal to 2 and high-level crossover is the lowest, it was selected as the control model for the Holm's test and it was ranked significantly better compared to algorithms with $i \geq 17$ in Table A30.

4.4

Comparison with Known Methods

As observed in Section 4.1, MG-PMA performed well for the benchmark functions of test sets 1 and 2. However on more complex functions, such as those in test set 3, MG-PMA faced difficulty in finding the global minimum. The

generalization settings of the PMA (MG-PMA with $k = n/2$ and $k = n$) performed better than the PMA (MG-PMA with $k = 1$), a method already compared to GA, PSO and DE (Pujol, 2008). Therefore, this section is devoted to go a step forward and present a comparison of MG-PMA with state-of-the-art metaheuristics for continuous optimization problems.

In order to solve the problems proposed by CEC'2015, within the rules proposed by the competition, it was chosen to use a hybrid strategy incorporating local minimization into global minimization. In this way, it was determined that 4-5%² of the maximum number of evaluations used were destined to local minimization. Therefore the global optimization algorithm made the greatest effort of the optimization and allowed the local minimizer to arrive as close as possible to the global minimum. The local optimization algorithm used was quasi-Newton, available in MATLAB through the *fminunc* function (MATLAB, 2016), and it was used on all individuals of the last generation, starting with the individual of better fitness to the one of worst fitness.

This strategy was used by MG-PMA and PSO, so that the comparison between the methods was as fair as possible. For PSO, the maximum number of local optimization used was 5% of the maximum number of evaluations, the population size was equal to 100 and the number of iterations was 95, the learning factors (weight for social and cognitive learning (Shi, 1998)) were equal to 2 and the inertia weight was equal to 0.9. These two methods were compared with the winners of the CEC'2015: SPS-L-SHADE-EIG (Guo, 2015), MVMO-SH (Awad, 2015), LSHADE-ND (Sallam, 2015), DEsPA (Rueda, 2015).

The best MG-PMA configuration found was used, with $k = n$ and tournament size equal to 2. The maximum tree depth d considered was 7. The crossover operator used was the same as that of the Section 4.1 (high-level crossover operator). All 15 functions of the CEC'2015 benchmark were considered (Table 4.2). The same parameters and configurations (Table 4.4, maximum number of evaluations before time-out equal to 10^5 and mathematical functions – *plus*, *minus*, *times* and *protected division*) of the test set 3 were used.

4.4.1 Results of Comparison with Known Methods

Table 4.7 presents the MBFV of each algorithm for all benchmark functions. The best results, among those found by the algorithms, are in bold, and it is possible to notice that SPS-L-SHADE-EIG (the winner of the competition of CEC'2015

²The maximum number of local evaluations varies according to the number of global evaluations performed. Whenever the direct reproduction operator is used and elitism is applied, individuals who are passed to the following generation are not reevaluated, making the number of global evaluations vary for each simulation.

(Guo, 2015)) was able to find the best result in most functions. MG-PMA and PSO obtained a lower performance than the other algorithms, but for some functions (MG-PMA for c_1 and c_{13} ; PSO for c_3 , c_9 , c_{13} and c_{15}), were able to find the best results. All numerical results of MG-PMA and PSO can be found in the Appendix A (Tables A32 and A32).

Table 4.7: Mean of the best fitness values (MBFV) for all functions of the CEC'2015 benchmark found by SPS-L-SHADE-EIG, DEsPA, LSHADE-ND, MVMO-SH, PSO and MG-PMA algorithms. The best results among those found by the algorithms are in bold.

Benchmark Function	SPS-L-SHADE-EIG	DEsPA	LSHADE-ND	MVMO-SH	PSO	MG-PMA
c_1	1.000e+02	1.000e+02	1.000e+02	1.000e+02	1.328e+02	1.000e+02
c_2	2.000e+02	2.000e+02	2.000e+02	2.000e+02	2.017e+02	2.004e+02
c_3	3.200e+02	1.970e+03	3.200e+02	3.200e+02	3.196e+02	3.200e+02
c_4	4.010e+02	4.036e+02	4.030e+02	4.020e+02	4.050e+02	4.303e+02
c_5	5.152e+02	5.519e+02	5.070e+02	5.118e+02	7.640e+02	9.606e+02
c_6	6.000e+02	6.016e+02	6.004e+02	6.014e+02	8.103e+02	8.731e+02
c_7	7.000e+02	7.003e+02	7.000e+02	7.000e+02	7.010e+02	7.015e+02
c_8	8.000e+02	8.002e+02	8.003e+02	8.003e+02	9.130e+02	9.007e+02
c_9	1.000e+03	1.006e+03	1.000e+03	1.000e+03	1.000e+03	1.001e+03
c_{10}	1.217e+03	1.008e+03	1.217e+03	1.217e+03	1.444e+03	1.446e+03
c_{11}	1.100e+03	1.195e+03	1.102e+03	1.104e+03	1.379e+03	1.369e+03
c_{12}	1.300e+03	1.301e+03	1.301e+03	1.301e+03	1.301e+03	1.303e+03
c_{13}	1.300e+03	1.318e+03	1.300e+03	1.300e+03	1.300e+03	1.300e+03
c_{14}	1.500e+03	1.709e+03	4.335e+03	1.500e+03	2.793e+03	2.400e+03
c_{15}	1.600e+03	1.705e+03	1.600e+03	1.600e+03	1.600e+03	1.602e+03

Table A34 shows the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the accuracy PSO, SPS-L-SHADE-EIG, DEsPA, LSHADE-ND, MVMO-SH and MG-PMA (Table 4.7). It can be seen that the average rank of SPS-L-SHADE-EIG was the lowest (30.700) of all algorithms tested.

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value less than 0.05 (p -value = 0.032), the hypothesis that there is no difference between the algorithms compared is rejected. As the average rank of SPS-L-SHADE-EIG is the lowest, it was selected as the control model for the Holm's test and it performed substantially better than MG-PMA and PSO. Note also that both PSO and MG-PMA performed with very similar ranks.

Table A34 shows that the SPS-L-SHADE-EIG algorithm resulted in the best performance for the benchmark functions of the CEC'2015. It was expected that the winning algorithms of the competition would be better than PSO and MG-PMA, since they were configured specifically to win the CEC'2015 (Guo, 2015, Awad, 2015, Sallam, 2015, Rueda, 2015), in which the participants were allowed to optimize the parameters of their proposed optimization algorithm for each problem (Liang, 2014), while the PSO and MG-PMA are optimizers configured to solve

general problems. In fact, as recently shown (Molina, 2017), algorithms optimized to win a specific edition of a CEC competition usually lead to poor results in other editions. It is interesting to note that MG-PMA performed similarly well compared to PSO, which suggests that MG-PMA is a promising approach for highly complex optimization problems. An important advantage of MG-PMA versus PSO is the fact that there is no need to establish the domains of the input parameters of the objective function for MG-PMA, while PSO is similar to a GA at the initialization of the population of random solutions (Eberhart, 1995).

5 Analyses of Niching MG-PMA

This chapter describes the analysis of multimodal optimization through Niching MG-PMA. Configurations were simulated sequentially to know the influence of the following parameters and operators: (i) high-level crossover operator; (ii) local optimization frequency; and (iii) feedback frequency. Finally, the best configuration was compared with two well-established algorithms.

The performance evolution of the algorithms was based on 20 benchmark multimodal functions with different characteristics and levels of difficulty from the 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization (Li, 2013). Appendix D presents the properties of these benchmark multimodal functions and their 3-D maps for the 2-D functions. Table 5.1 presents these benchmark set.

Table 5.1: Benchmark functions proposed for 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization. The table presents the abbreviation code, the name of each function, the variable ranges, the number of the parameters to be optimized and the number of global optima.

Abbreviation	Function Name	Variable Ranges	Number of Parameters	Number of Global Optima
p_1	Five-Uneven-Peak Trap	$[0, 30]$	1	2
p_2	Equal Maxima	$[0, 1]$	1	5
p_3	Uneven Decreasing Maxima	$[0, 1]$	1	1
p_4	Himmelblau	$[-6, 6]^2$	2	4
p_5	Six-Hump Camel Back	$[-1.9, 1.9]$ $[-1.1, 1.1]$	2	2
p_6	Shubert	$[-10, 10]^2$	2	18
p_7	Vincent	$[0.259, 10]^2$	2	36
p_8	Shubert	$[-10, 10]^3$	3	81
p_9	Vincent	$[0.259, 10]^3$	3	216
p_{10}	Modified Rastrigin - All Global Optima	$[0, 1]^2$	2	12
p_{11}	Composition Function 1	$[-5, 5]^2$	2	6
p_{12}	Composition Function 2	$[-5, 5]^2$	2	8
p_{13}	Composition Function 3	$[-5, 5]^2$	2	6
p_{14}	Composition Function 3	$[-5, 5]^3$	3	6
p_{15}	Composition Function 4	$[-5, 5]^3$	3	8
p_{16}	Composition Function 3	$[-5, 5]^5$	5	6
p_{17}	Composition Function 4	$[-5, 5]^5$	5	8
p_{18}	Composition Function 3	$[-5, 5]^{10}$	10	6
p_{19}	Composition Function 4	$[-5, 5]^{10}$	10	8
p_{20}	Composition Function 4	$[-5, 5]^{10}$	20	8

All benchmark functions are formulated as maximization problems and the domain constraints of all benchmark functions was followed using the PBC presented in Section 3.3. Five levels of accuracy were used as threshold values to consider if a global optimum was found: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. The number of independent runs was equal to 10 for each test and benchmark function performed, except for the comparison with the well-established algorithms, where the number was equal to 50 (Section 5.4). Table 5.2 presents the maximum number of evaluations before time-out for each benchmark function, as defined in the 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization (Li, 2013). In addition to the internal local optimization of the winners used in Niching MG-PMA, the same hybrid strategy of the last chapter was done, incorporating local optimization after Niching MG-PMA optimization. In this way, it was determined that 5% of the maximum number of evaluations used were destined to this local optimization. The local optimization algorithm used both for internal local optimization of Niching MG-PMA and for local optimization after Niching MG-PMA was Interior-Point, available in MATLAB through the *fmincon* function (MATLAB, 2016).

The change of the local optimization algorithm, relative to the previous chapter, was due to the fact that this algorithm handles large, sparse problems, as well as small dense problems and satisfies the bounds at all iterations (constraint needed to optimize the benchmark multimodal functions 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization (Li, 2013)). The number of iterations for each local optimization has been set for a maximum of 100 evaluations.

Table 5.2: Maximum number of function evaluations (*MaxFEs*) for each benchmark multimodal function.

Range of functions	MaxFEs
p_1-p_5	5.0e+04
$p_6,p_7,p_{10}-p_{13}$	2.0e+05
$p_8,p_9,p_{14}-p_{20}$	4.0e+05

The mathematical functions used in the Niching MG-PMA were *plus*, *minus*, *times* and *protected division*. The remaining configuration parameters of Niching MG-PMA are listed in Table 5.3.

As listed in Table 5.3, the probability of applying the mutation operator in the experiments was 30% (value defined through preliminary analyzes presenting low sensitivity). Three types of mutation (Searson, 2015) were employed with the following probabilities: ordinary sub-tree mutation (5%); switch an input terminal to another randomly selected input terminal (47.5%); and Gaussian perturbation of a randomly selected constant (47.5%, with the standard deviation of the Gaussian

Table 5.3: General configuration of Niching MG-PMA.

Parameters	Values
maximum tree depth d	7
Decimal ERC* rate	90%
Integer ERC* rate	0%
ERC* range	$[-1, +1]$
Low-level crossover rate	65%
Mutation rate	30%
Direct reproduction rate	5%
High-level crossover rate	50%
Elitism rate	1%
Lexicographic pressure (Luke, 2002)	Yes

* ERC - Ephemeral Random Constant: constants generated in a specific range by the user to constitute the trees of each individual together with variables (from the initial parameter values to be mapped by MG-PMA).

used equal to 10%) – the parameter values for these three mutation types were found in preliminary tests with the same benchmark multimodal functions. The initial population was generated using the ramped half-and-half method (Koza, 1992).

Equation 5-1 presents the calculation of the niche radius R_g , which is used in the variable niche radius approach created in this thesis for Niching MG-PMA. The value of R_g is changed according to population, avoiding very large or small sizes. Different approaches can be found in the literature (Li, 2016): (i) use a single uniform niche radius (Jelasity, 1998, Dick, 2010); (ii) use a variable niche radius (Ursem, 1999, Bird, 2006, Gan, 2001); and (iii) avoid specifying the niche radius (Ursem, 1999, Mahfoud, 1992).

$$R_g = \frac{\text{median}(\text{AllDistances}_g)}{10} \quad (5-1)$$

where R_g is the niche radius of the generation g and AllDistances_g are all distances between the individuals of the population in the generation g .

As a performance measure, this chapter uses peak ratio (PR) (Li, 2013) to evaluate the performance of the different cases studied. PR measures the average percentage of all known global optima found over multiple runs (Equation 5-2).

$$PR = \frac{\sum_{run=1}^{NR} NPF_i}{NKP \times NR} \quad (5-2)$$

where NPF_i is the number of global optima found in the end of the i -th run, NKP is the number of known global optima and NR is the number of runs.

5.1 High-Level Crossover Operator

This section presents an analysis of the high-level crossover operators for the benchmark multimodal functions. The analysis covered the comparison of the following operators: (i) high-level crossover; (ii) high-level single-point crossover; (iii) high-level two-point crossover; and (iv) high-level uniform crossover.

For this first analysis, the local optimization and the updating of the initial parameters by the feedback were applied to each generation (local optimization frequency and feedback frequency parameters equal to 1).

5.1.1 Results of High-Level Crossover Operator

Figure 5.1 (Figures C1-C4 in Appendix C) depicts the results for the different high-level crossover operators and show the *PR* for each benchmark function with five level of accuracy.

Figure 5.1.a (Figure C1 in Appendix C) shows that the high-level crossover operator was able to find all known global optima in all runs for p_1 - p_6 and p_{10} for the five level of accuracy (except in p_6 with accuracy equal to $1e-05$, where no global optima was found) and almost all known global optima in all runs for p_7 . Niching MG-PMA with high-level crossover was not able to find any global optima for p_{16} - p_{20} . For functions p_8 , p_9 and p_{11} - p_{15} , this multi-gene genetic operator was able to find *PR* values between 0.1 and 0.9.

Figure 5.1.b (Figure C2 in Appendix C) shows that the high-level single-point crossover operator was able to find all known global optima in all runs for p_1 - p_6 and p_{10} for the five level of accuracy (except in p_6 with accuracy equal to $1e-05$, where no global optima was found). Niching MG-PMA with high-level single-point crossover was able to find all known global optima in almost all runs in p_7 ($PR = 0.994$ for $1e-01$ accuracy and $PR = 0.992$ for the other accuracies). Better than high-level crossover, high-level single-point crossover finds *PR* values greater than zero in p_{16} and p_{17} for all accuracies, but for p_{18} - p_{20} was also not able to find any global optima too. For functions p_8 , p_9 and p_{11} - p_{15} , this multi-gene genetic operator was able to find *PR* values between 0.1 and 1.

Figure 5.1.c (Figure C3 in Appendix C) shows that the high-level two-point crossover operator was able to find all known global optima in all runs for p_1 - p_6 and p_{10} for all accuracies levels (except in p_6 with accuracy equal to $1e-05$, where no global optima was found) and almost all known global optima in all runs in p_7 . High-level two-point crossover finds *PR* values greater than zero for function p_{16} with the five level of accuracy, but for p_{17} - p_{20} was not able to find any global optima

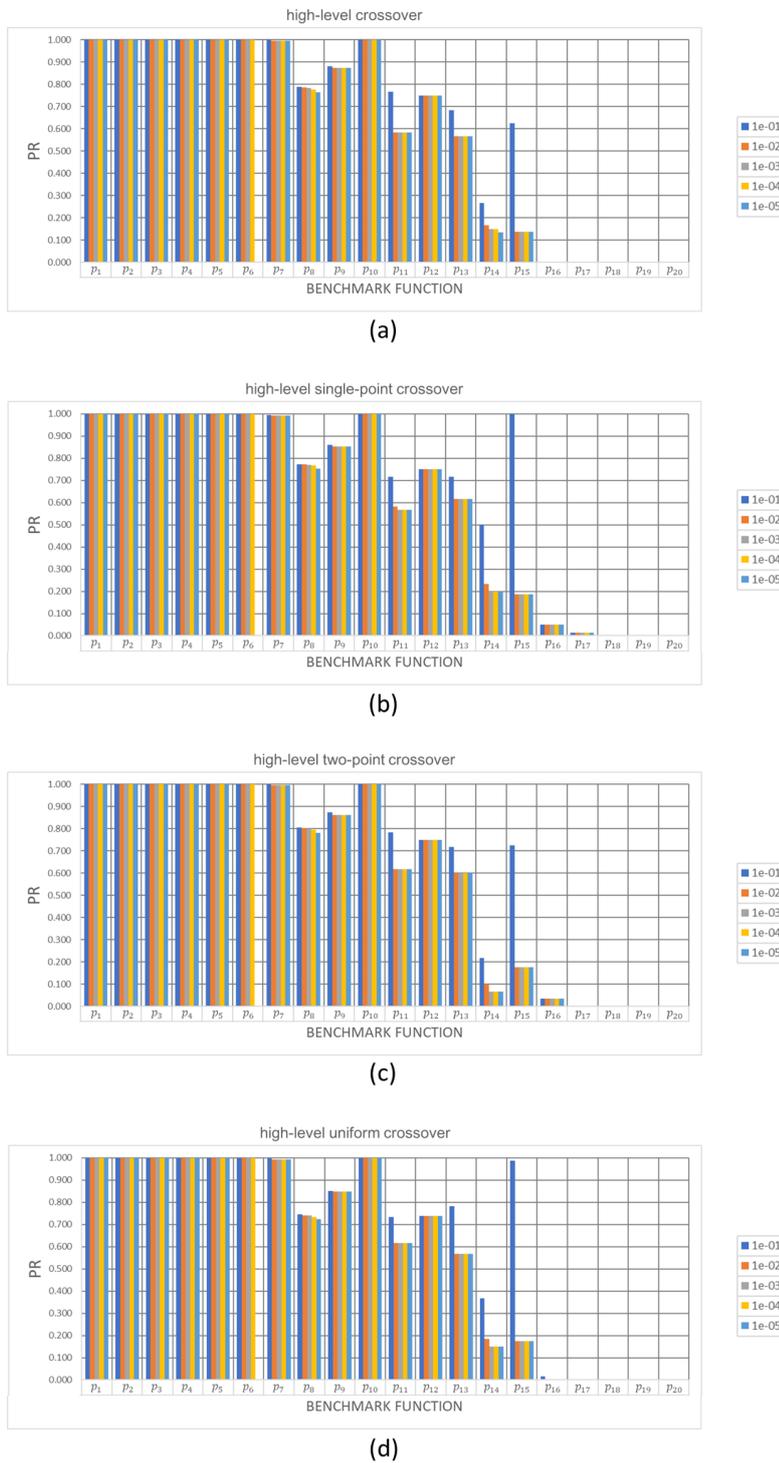


Figure 5.1: Bar chart with peak ratio (PR) results from Niching MG-PMA using four types of crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. (a) high-level crossover; (b) high-level single-point crossover; (c) high-level two-point crossover; and (d) high-level uniform crossover.

too. For functions p_8 , p_9 and p_{11} - p_{15} , this multi-gene genetic operator was able to find PR values between 0 and 1.

Figure 5.1.d (Figure C4 in Appendix C) shows that the high-level uniform crossover operator was able to find all known global optima in all runs for p_1 - p_6 and p_{10} for all accuracies (except in p_6 with accuracy equal to $1e-05$, where no global optima was found). Niching MG-PMA with high-level uniform crossover was able to find all known global optima in almost all runs for function p_7 within the five level of accuracy ($PR = 1.000$ for $1e-01$ accuracy and $PR = 0.992$ for the others). Niching MG-PMA with high-level uniform crossover was not able to find any global optima for p_{16} - p_{20} (except for p_{16} for $1e-01$ accuracy, where $PR = 0.017$). For functions p_8 , p_9 and p_{11} - p_{15} , this multi-gene genetic operator was able to find PR values between 0.1 and 1.

Two facts observations can be made regardless of the high-level crossover operator used by Niching MG-PMA for the five level of accuracy. The method: (i) found all known global optima in all runs for functions p_1 - p_6 and p_{10} (except in p_6 with accuracy equal to $1e-05$ where no global optima was found); and (ii) was not able to find any global optima for functions p_{18} - p_{20} . It is important to note that for $1e-1$ the high-level single-point crossover presented better results for benchmark functions with higher dimensionalities (p_{15} and p_{17}) that are closer to real-world problems.

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value less than 0.05 (p -value = 0.017), the hypothesis that there is no difference between all high-level crossover operators tested is rejected (Tables C1-C5). As the average rank of Niching MG-PMA with high-level two-point crossover is the lowest, it was selected as the control model for the Holm's tests and the best operator among the tested operators. Tables C1-C5 present the remaining results of the statistical procedure (z -scores, p -values and Holm's correction for the significance level).

5.2

Local Optimization Frequency

This section presents an analysis of the local optimization frequency for Niching MG-PMA, that is the local optimization is used every i generations, with the values considered for i equal to $\{1, 10, 50, 100, 150, 200\}$. These configurations are identified in this section as Li (L1, L10, L50, L100, L150 and L200). The high-level two-point and single-point crossover were used in these tests, since the two-point was the best configuration of the statistical analysis of the previous section and single-point obtained superior qualitative results (it was better for functions with greater dimensionality).

For this analysis, the updating of the initial parameters by the feedback was

applied to each generation (feedback frequency parameters equal to 1).

5.2.1 Results of Local Optimization Frequency

Figures 5.2 (L150 with high-level single-point crossover operator) and the Appendix C (Figures C2 – L1 with high-level single-point crossover operator; C3 – L1 with high-level two-point crossover operator; and C5-C13) present the numerical results for local optimization frequency analysis. The different configurations of Niching MG-PMA found all known global optima in all runs for p_1 - p_5 and p_{10} . For accuracy equal to $1e-05$, the method was not able to find any global optima in p_6 .

The Aligned Friedman’s and Iman-Davenport’s tests, along with the Holm’s procedure based on the different values of local optimization frequency tested for Niching MG-PMA, were performed for all five level of accuracy. The average rank of Niching MG-PMA with high-level single-point crossover was the lowest of all configurations tested, being that for $1e-01$ accuracy it was the method using local optimization frequency equal to 150 (Figure 5.2) and for the other accuracies the lowest of all configurations was the method using local optimization frequency equal to 1 (Figure C2).

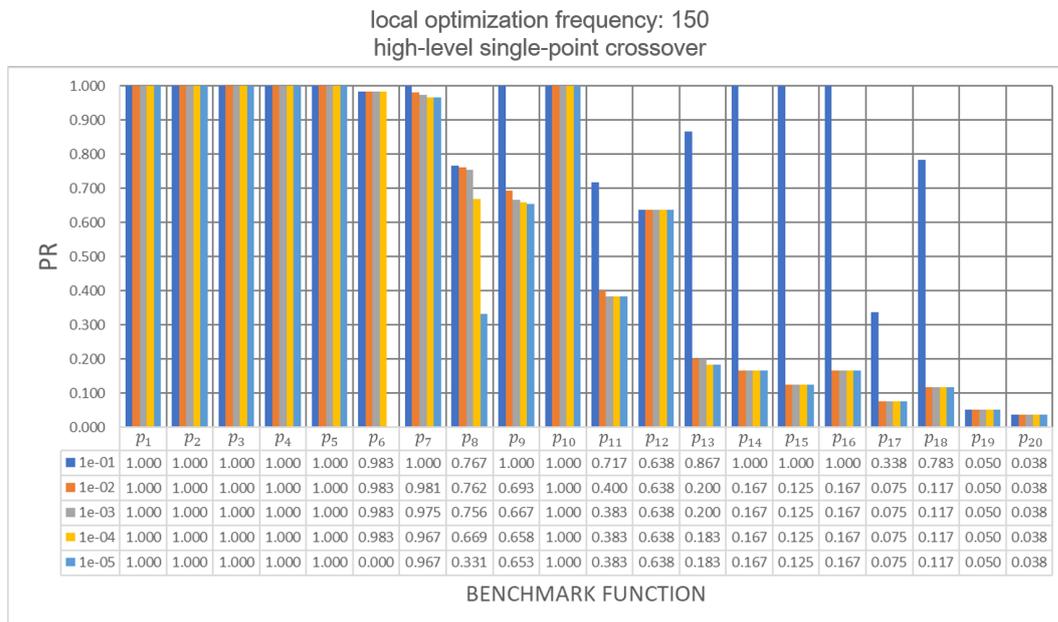


Figure 5.2: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 150, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

Both tests (Aligned Friedman’s and Iman-Davenport’s) found a p -value less

than 0.05 (p -value = 0.005), the hypothesis that there is no difference between the configurations tested is rejected. The lowest average rank for each accuracy was considered as control models for the Holm's tests.

Although the configuration with local optimization frequency equals to 1 was better classified in the rank for the $1e-02$, $1e-03$, $1e-04$ and $1e-05$ accuracies, the configuration with local optimization frequency equals to 150 (the best ranking for $1e-01$ accuracy) obtained good results for almost all functions. In addition, this configuration was able to find some global optima for the functions with greater dimensionality (p_{16} - p_{20}), showing its potential for application in real-world problems.

5.3

Feedback Frequency

From the Niching MG-PMA with local optimization frequency equals to 150, and using high-level single-point crossover, this section analyzes the effect of the feedback frequency parameter. Feedback is applied every j generations, with the values considered for j equal to $\{0, 1, 10, 50, 100, 150, 200\}$. These configurations are identified in this section as F_j (F_0 – without the feedback approach, F_1 , F_{10} , F_{50} , F_{100} , F_{150} and F_{200}).

5.3.1

Results of Feedback Frequency

Figure 5.3 and Appendix C (Figures C14-C18) present the numerical results for feedback frequency analysis. The different configurations of Niching MG-PMA found all known global optima in all runs for p_1 - p_5 and p_{10} . For accuracy equal to $1e-05$, the method was not able to find any global optima for p_6 .

The Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the different values of feedback frequency tested for Niching MG-PMA, were performed for all five level of accuracy. The average rank of Niching MG-PMA with feedback frequency of 1 was the lowest of all configurations tested for $1e-01$ - $1e-04$ accuracies (Figure 5.2) and for $1e-05$ accuracy the lowest of all configurations was the method using feedback frequency of 10 (Figure 5.3).

Both tests (Aligned Friedman's and Iman-Davenport's) rejected the hypothesis that there is no difference between the configurations tested is rejected (p -value ≤ 0.05). The lowest average rank for each accuracy was considering as control models for the Holm's tests. Tables C11-C15 present the remaining results of the statistical procedure (z -scores, p -values and Holm's correction for the significance level). In the first two accuracies, the use of feedback frequency

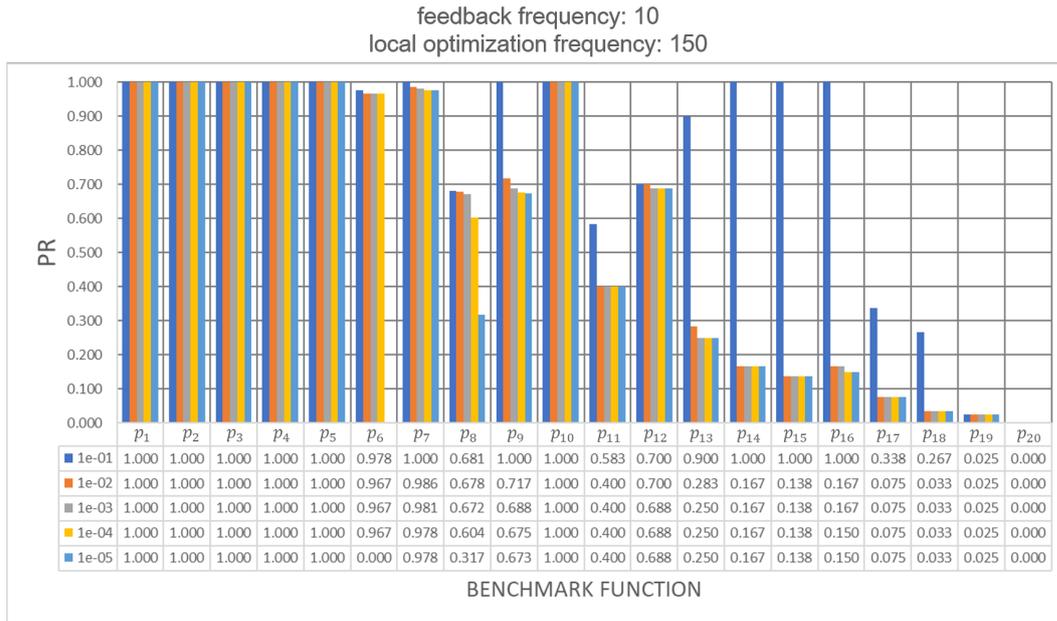


Figure 5.3: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 10, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

of 1 was ranked better than the other configurations, but nothing can say about the significant difference. For $1.0e-03$ accuracy, the use of feedback frequency of 1 was better than 50. For $1.0e-04$ accuracy, the feedback parameter of 1 was better than values of 50 and 150. Finally, for the fifth accuracy the feedback parameter of 10 was better than values equal to 50 and 150.

In addition to the feedback parameter of 1 being better ranked than 10 in the first four accuracies, it is possible to note that the feedback parameter equal to 1 has better results for higher dimensionality functions. Therefore, the feedback frequency parameter of 1 was considered the best value for Niching MG-PMA in the benchmark tested.

5.4 Comparison with Well-Established Algorithms

This section presents a comparison between Niching MG-PMA and two algorithms recommended by 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization (Li, 2013) as baseline: DE/nrand/1/bin (Epitropakis, 2011) and Crowding DE/rand/1/bin (Thomsen, 2004). This baseline algorithms are DE niching variants that use: (i) spatial information about the neighborhood of each potential solution to produce a niching formation –

DE/nrand/1/bin; and (ii) crowding technique to maintain a better population diversity and therefore to prevent premature convergence to an optimum – Crowding DE/rand/1/bin.

The best configurations found for Niching MG-PMA in this thesis were used in this comparison: (i) local optimization and feedback frequency parameters equal to 1 (Niching MG-PMA L1); and (ii) local optimization frequency parameter equal to 150 and feedback frequency parameter equal to 1 (Niching MG-PMA L150).

5.4.1

Results of Comparison with Well-Established Algorithms

Figures 5.4-5.8 depict the PR of the Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy.

Results for $1e-01$ Accuracy

Figure 5.4 shows the results for the accuracy of $1e-01$. The algorithms were able to find all known global optima in all runs for functions p_1 - p_5 and p_{10} . DE/nrand/1/bin was the best algorithm to optimize p_{12} , but was the worst algorithm to deal with p_6 - p_9 , p_{13} and p_{15} . Crowding DE/rand/1/bin was the best algorithm in p_6 (with the same PR value of Niching MG-PMA L1), p_8 , p_{11} , p_{17} and p_{20} and it was the worst one to handle p_{12} and p_{19} (with the same PR value of Niching MG-PMA L1). Niching MG-PMA L1 was the best algorithm in p_6 (with the same PR value of Crowding DE/rand/1/bin), p_7 and p_{15} (with the same PR values of Niching MG-PMA L150), but was the worst one in p_{14} and p_{16} - p_{20} (with the same PR value of Crowding DE/rand/1/bin in p_{19}). Niching MG-PMA L150 was the best algorithm in p_7 , p_9 , p_{13} - p_{16} and p_{18} (with the same PR values of Niching MG-PMA L1 in p_7 and p_{15}).

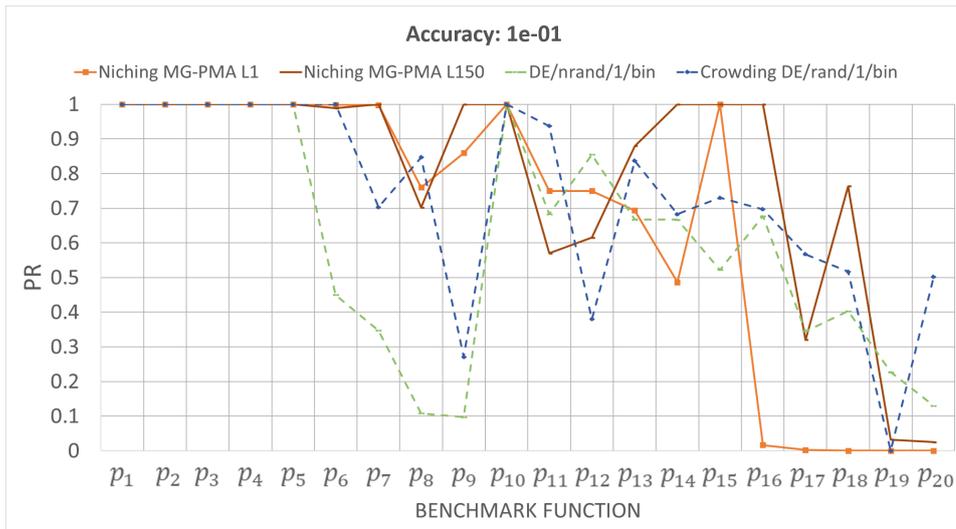


Figure 5.4: Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1 - p_{20}) with accuracy equal to $1e-01$.

Overall, the Niching MG-PMA L150 was the best algorithm for accuracy equal to $1e-01$. In order to evaluate if this advantage was not a mere coincidence, but a statistically significant outperformance, Table C16 shows the results of Aligned Friedman's (Derrac, 2011) and Iman-Davenport's (Iman, 1980) tests, along with the Holm's (Holm, 1979) procedure based on the algorithms performance in the benchmark multimodal functions. These tests were carried out using the KEEL software (Alcalá-Fdez, 2011). It can be seen that the average rank of Niching MG-PMA L150 was the lowest (31.225) of all algorithms tested.

Both tests (Aligned Friedman's and Iman-Davenport's) found a p -value less than 0.05 (p -value = 0.025), so there is a significant difference between these algorithms. As the average rank of Niching MG-PMA L150 is the lowest, it was selected as the control model for the Holm's test. Table C16 presents the remaining results of the statistical procedure (z -scores, p -values and Holm's correction for the significance level). It can be seen that Niching MG-PMA L150 was ranked significantly better than DE/nrand/1/bin. Nothing can be said in respect to the comparison between Niching MG-PMA L150 and the other algorithms statistically yielding better or worse results.

Results for $1e-02$ Accuracy

Figure 5.5 shows the results for the accuracy equal to $1e-02$. Almost all algorithms were able to find all known global optima for functions p_1 - p_5 and p_{10} , with the exception of Crowding DE/rand/1/bin for function p_1 where PR value was between 0.7 and 0.8. DE/nrand/1/bin was the best algorithm to optimize p_{12} , p_{14}

and p_{16} (with the same PR values of Crowding DE/rand/1/bin), and p_{18} - p_{20} , but was the worst algorithm to deal with p_6 - p_9 . Crowding DE/rand/1/bin was the best algorithm for p_6 (with the same PR value of Niching MG-PMA L1), p_8 , p_{11} and p_{13} - p_{17} (with the same PR values of DE/nrand/1/bin in p_{14} and p_{16}) and it was the worst one to handle p_{12} and p_{19} (with the same PR value of Niching MG-PMA L1). Niching MG-PMA L1 was the best algorithm for p_6 (with the same PR value of Crowding DE/rand/1/bin) and p_7 , but was the worst one in p_{16} - p_{20} (with the same PR value of Crowding DE/rand/1/bin in p_{19}). Niching MG-PMA L150 was the worst one to handle for p_{11} and p_{13} - p_{15} .

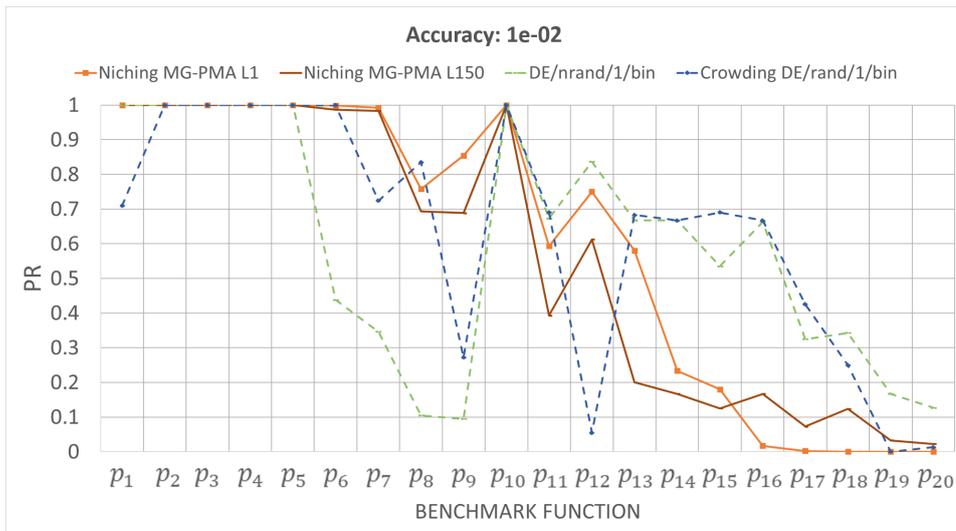


Figure 5.5: Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1 - p_{20}) with accuracy equal to $1e-02$.

Table C17 shows the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the algorithms performance in the benchmark multimodal functions. It can be seen that the average rank of DE/nrand/1/bin was the lowest (36.300) of all algorithms tested.

Results for $1e-03$ Accuracy

Figure 5.6 shows the results for the accuracy equal to $1e-03$. As well as $1e-02$ accuracy, almost all algorithms were able to find all known global optima for functions p_1 - p_5 and p_{10} , with the exception of Crowding DE/rand/1/bin for function p_1 where PR value was between 0 and 0.1. DE/nrand/1/bin was the best algorithm to optimize p_{11} - p_{14} and p_{16} - p_{20} (with the same PR values of Crowding DE/rand/1/bin in p_{13} , p_{14} and p_{16}), but was the worst algorithm to deal with p_6 - p_9 . Crowding DE/rand/1/bin was the best algorithm for p_{13} - p_{16} (with the same PR value of DE/nrand/1/bin in p_{13} , p_{14} and p_{16}) and it was the worst one to handle p_{12} ,

p_{19} and p_{20} (with the same PR value of Niching MG-PMA L1). Niching MG-PMA L1 was the best algorithm for p_6 - p_9 , but was the worst one for p_{16} - p_{20} (with the same PR values of Crowding DE/rand/1/bin in p_{19} and p_{20}). Niching MG-PMA L150 was the worst one to handle in p_8 , p_{11} and p_{13} - p_{15} .

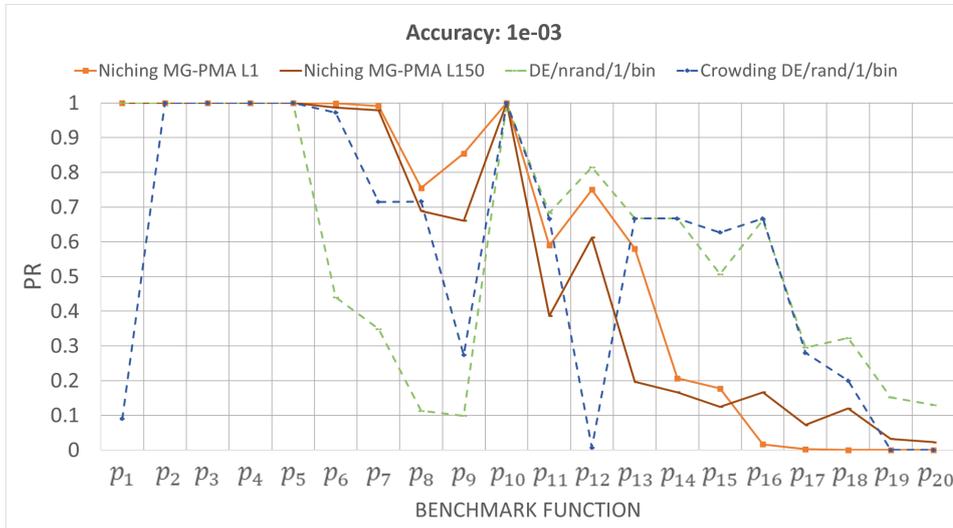


Figure 5.6: Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1 - p_{20}) with accuracy equal to $1e-03$.

Table C18 shows the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the algorithms performance in the benchmark multimodal functions. It can be seen that the average rank of DE/nrand/1/bin was the lowest (36.350) of all algorithms tested.

Results for $1e-04$ Accuracy

Figure 5.7 shows the results for the accuracy equal to $1e-04$. Once again, almost all algorithms were able to find all known global optima for functions p_1 - p_5 and p_{10} , with the exception of Crowding DE/rand/1/bin for function p_1 where PR value was between 0 and 0.1. DE/nrand/1/bin was the best algorithm to optimize p_{11} - p_{20} (with the same PR values of Crowding DE/rand/1/bin for p_{13} , p_{14} and p_{16}), but was the worst one for p_7 - p_9 . Crowding DE/rand/1/bin was the best algorithm for p_{13} , p_{14} and p_{16} (with the same PR values of DE/nrand/1/bin in this functions) and it was the worst one to handle p_6 , p_{12} , p_{19} and p_{20} (with the same PR value of Niching MG-PMA L1). Niching MG-PMA L1 was the best algorithm for p_6 - p_9 , but was the worst one for p_{16} - p_{20} (with the same PR values of Crowding DE/rand/1/bin for p_{19} and p_{20}). Niching MG-PMA L150 was the worst one to handle in p_8 , p_{11} and p_{13} - p_{15} .

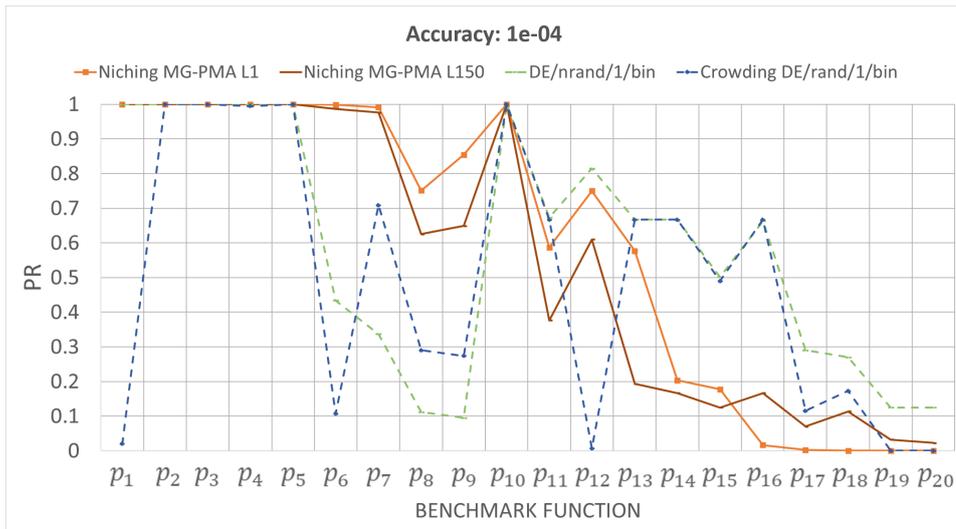


Figure 5.7: Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1 - p_{20}) with accuracy equal to $1e-04$.

Table C19 shows the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the algorithms performance in the benchmark multimodal functions. It can be seen that the average rank of DE/nrand/1/bin was the lowest (34.400) of all algorithms tested.

Results for $1e-05$ Accuracy

Figure 5.8 shows the results for the accuracy equal to $1e-05$. Again, almost all algorithms were able to find all known global optima for functions p_1 - p_5 and p_{10} , with the exception of Crowding DE/rand/1/bin for functions p_1 and p_4 where PR values were equal to 0.020 and 0.995, respectively. DE/nrand/1/bin was the best algorithm to optimize p_{11} - p_{15} and p_{17} - p_{20} (with the same PR values of Crowding DE/rand/1/bin in p_{13} and p_{14}), but was the worst one for p_7 - p_9 . Crowding DE/rand/1/bin was the best algorithm for p_{13} , p_{14} and p_{16} (with the same PR values of DE/nrand/1/bin in p_{13} and p_{14}) and it was the worst one to handle p_6 , p_{12} , p_{19} and p_{20} (with the same PR values of Niching MG-PMA L1 in p_{19} and p_{20}). Niching MG-PMA L1 was the best algorithm for p_6 - p_9 , but was the worst one for p_{16} - p_{20} (with the same PR values of Crowding DE/rand/1/bin in p_{19} and p_{20}). Niching MG-PMA L150 was the worst one to handle for p_8 , p_{11} and p_{13} - p_{15} .

Table C20 shows the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the algorithms performance in the benchmark multimodal functions. It can be seen that the average rank of DE/nrand/1/bin was the lowest (31.225) of all algorithms tested.

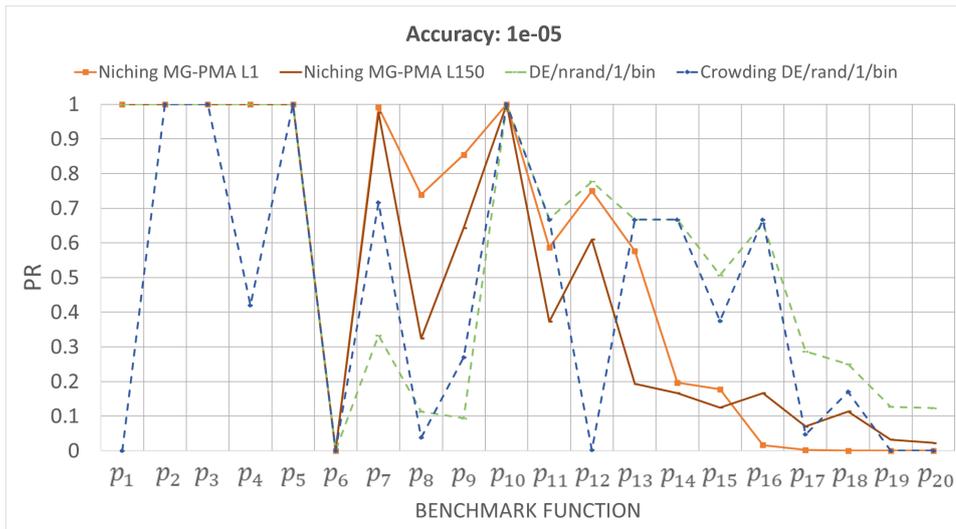


Figure 5.8: Peak ratio (PR) of the MG-PMA L1, MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin algorithms for 20 benchmark multimodal functions (p_1 - p_{20}) with accuracy equal to $1e-05$.

Statistical Analysis for $1e-02$ - $1e-05$ Accuracies

For the accuracies $1e-02$ - $1e-05$, both tests (Aligned Friedman's and Iman-Davenport's) found a p -value less than 0.05 (p -value = 0.017), so there is a significant difference between the algorithms. As the average rank of DE/nrand/1/bin is the lowest, it was selected as the control model for the Holm's test.

Conclusions of Comparison with Well-Established Algorithms

There is a significant difference between Niching MG-PMA L150 and DE/nrand/1/bin for accuracy $1e-01$. It shows the potential of using Niching MG-PMA for real-world problems that demand a multitude of solutions. For the other four accuracies, DE/nrand/1/bin was significantly better than Niching MG-PMA. However, the positive result found in the first accuracy ($1e-01$) also demonstrates that Niching MG-PMA can be optimized to be competitive in the other accuracies ($1e-02$ - $1e-05$).

6 Optimization of Quantum Well Infrared Photodetectors

Quantum well infrared photodetectors (QWIPs) are photon detectors based on nanostructured materials known as quantum wells (QWs) (Levine, 1993). QWs are able to selectively detect radiation in the mid-infrared region ($3\text{-}6\mu\text{m}$ or $207\text{-}413\text{ meV}$) using optical transitions within the conduction band (intraband absorption – see Figure 6.1) (Penello, 2013). Electrons that are at a fundamental level of a QW are photoexcited to a higher level in energy and then collected, thus generating a photocurrent that depends fundamentally on the energy levels of the nanostructure.

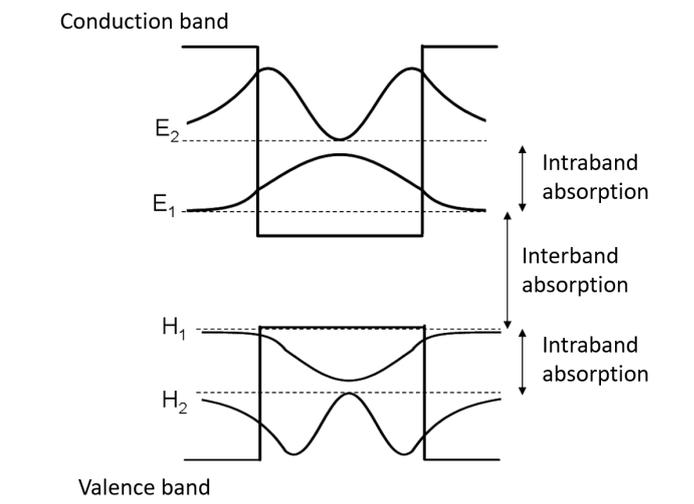


Figure 6.1: Interband and intraband absorption in a quantum well structure (Penello, 2013).

It can be noticed that the structure of the sample is a key point in the development of photodetectors, since, by controlling the structure, it is possible to control the energy difference between the levels of the different states. By experimenting with different thicknesses for the QWs and coupling wells to form superlattices, it is possible to control the energy levels below the potential barrier (levels inside the well) as well as the energy levels above this barrier.

Two advantages can be obtained by controlling the levels (Penello, 2013): (i) the transitions are no longer limited by the height of the potential barrier (bandoffset) determined by the material of the wells and barriers; and (ii) control of energy levels above the barrier can be used to minimize the dark current

of a detector. The important point of the second advantage is the possibility of photodetectors operating at temperatures close to room temperature. The operation of QWIPs is currently limited to cryogenic temperatures, the search for operation at higher temperatures is critical in reducing the complexity of using the QWIPs in portable and easy-to-implement devices.

In 2018, Penello (Penello, 2018) presented a photodetector based on an asymmetric InGaAs/InAlAs superlattice with a structural defect. The structural defect creates localized states in the continuum. Due to the asymmetry of the heterostructure, the electron in the localized state in the continuum has a preferable direction to flow. That is, the photodetector based on asymmetric superlattice can be operated at photovoltaic mode, without an applied bias voltage. The photocurrent spectrum presented a good signal to noise ratio up to room temperature.

Figure 6.2 presents the conduction band of this asymmetric superlattice that has five QWs with a thickness of 2.0 nm, the sixth one has a thickness of 2.5 nm and the seventh QW is as thick as the first five ones. The sixth QW can be interpreted as a defect in the superlattice and creates localized states within and outside the miniband (allowed regions for the electronic state in the continuum). The QWs are separated by 7.0 nm thick barriers. The asymmetric superlattice was repeated 20 times and each superlattice was separated from the following one by a thick quantum barrier of 30 nm to avoid the coupling of neighboring superlattices.

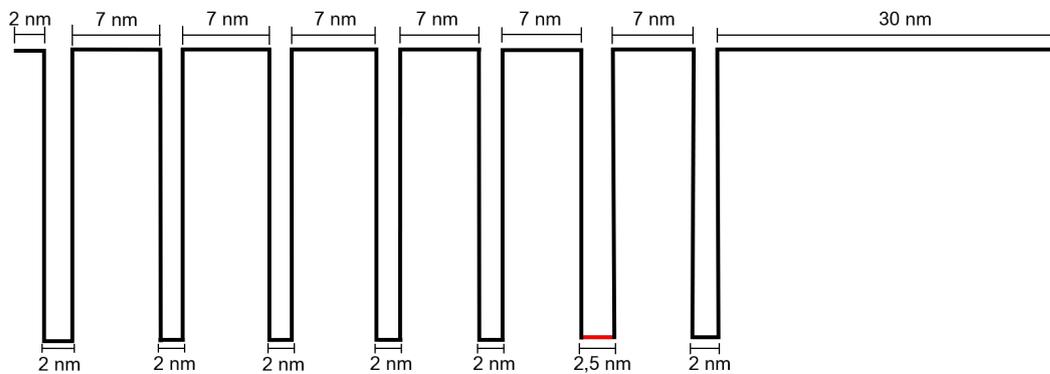


Figure 6.2: Asymmetric superlattice with the thickness of the quantum wells and barriers proposed by Penello (Penello, 2018). The red line represents the defective quantum well.

In 2013, Penello (Penello, 2013) developed an easy-to-use computational program that calculates the states of the heterostructure. With this program, it is possible to calculate the energy levels of the structures, the wave functions, the oscillator strengths between the levels and the absorption spectrum. Figure 6.3 shows the potential profile for the conduction band superimposed with the squared

modulus of the wave function and oscillator strength versus energy for the photodetector presented in 2018 (Penello, 2018). The ground state and the state located in the continuum for the energy transition of 300 meV is highlighted in Figure 6.3 (in blue). It is possible to note that the wave function of the state located in the continuum is confined in the defect QW.

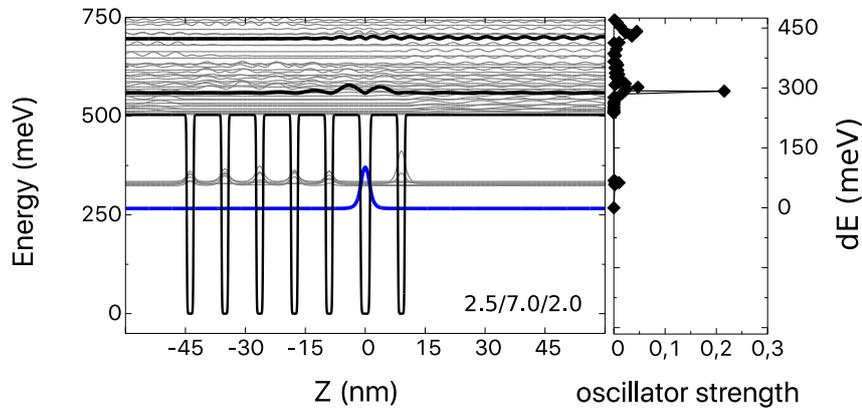


Figure 6.3: Potential profile of the structure presented by Penello in 2018 (Penello, 2018), showing the modulus squared wave functions and oscillator strength *versus* energy.

By altering the thicknesses of the wells and barriers it is possible to tailor the energy and the oscillator strength of the optical transitions between the ground state and the localized states in the continuum. Until the work presented in this chapter, the program that calculates the states of the heterostructure had to be used by trial and error to try to reproduce experimental results or design new structures with certain transition energies, which ended up being monotonous and time-consuming.

By applying MG-PMA to control and optimize the layers' thicknesses of photodetectors, it is possible to find desired structures and test new ideas in the area. This chapter presents an optimized structure of the asymmetric superlattice proposed by Penello (Penello, 2018) using the program that calculates the states of the heterostructure. For comparison purposes, the same photodetector was optimized using GA, PSO and covariance matrix adaptation evolution strategy (CMA-ES) (Hansen, 2001, Hansen, 2006).

6.1 Representation

The representation of individuals is given by three real variables: (i) a – the thickness of the first five and the seventh QWs; (ii) b – the thickness of the barriers separating the QWs; and (iii) c – the thickness of the sixth QW that represents the defect in the superlattice. The initial barrier (B_{ini}) has a fixed thickness equal to 2 nm and the last barrier (λ) varies according to the thickness of the variable wells

and barriers (values of a , b and c). Equation 6-1 presents the thickness calculation for the last barrier.

$$\begin{aligned} center &= B_{ini} + 5a + 5b + \frac{c}{2} \\ \lambda &= center - \frac{c}{2} - a - b \end{aligned} \quad (6-1)$$

This variation of the thickness of the last barrier is due to the fact that the heterostructure is centralized in the middle of the defective well by the software that calculates the energy levels of the structure. Figure 6.4 presents the representation of the asymmetric superlattice for the optimization.

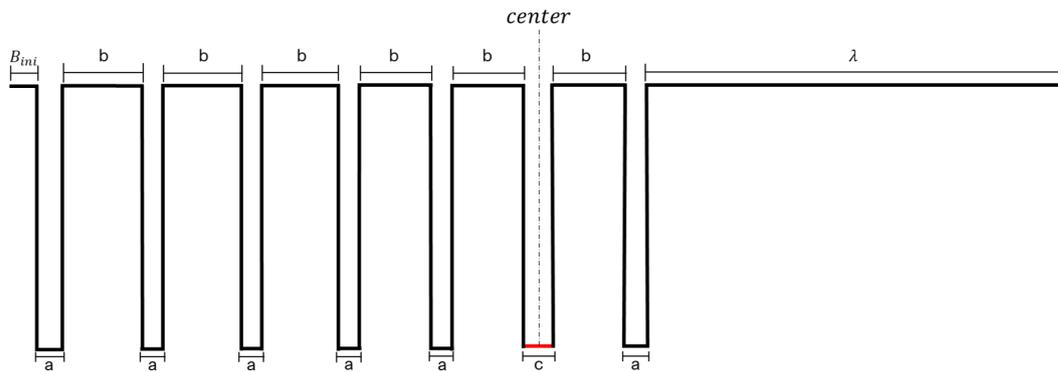


Figure 6.4: Asymmetric superlattice with the thickness of the quantum wells and barriers for the structural optimization. The red line represents the defective quantum well.

The optimization was performed starting from the heterostructure proposed by Penello (Figure 6.2). The values of a , b and c were allowed to vary by $\pm 50\%$ with respect to the initial values.

6.2 Objective Function

This structural optimization is a maximization problem that aims to find the heterostructure with the greater oscillator strength (f_{0d}) of the optical transition between the ground state and the desired energy within a given energy range. The oscillator strength is a magnitude that determines how strong the two-state transition is by means of an oscillating field (i.e. infrared radiation). The greater the oscillator strength, the stronger this transition will be, giving indications that an intense photocurrent peak will occur. Equation 6-2 presents the calculation of the oscillator strength.

$$f_{0d} = \frac{2m^*}{\hbar} (E_d - E_0) |M_{0d}|^2$$

$$|M_{0d}|^2 = \left| \int_{-\infty}^{+\infty} \psi_d^*(z) (-ez) \psi_0(z) dz \right|^2 \quad (6-2)$$

where m^* is the effective mass of the electron, \hbar is the Plank constant divided by 2π , E_d is the desired energy and E_0 is the ground state energy, M_{0d} is the dipole moment between the ground state and the desired energy, ψ_0 is the wave function of the ground state, ψ_d is the wave function of the desired state, z is the position, e is the electronic charge. More details on these calculations can be found in Penello's Ph.D. thesis (Penello, 2013).

The energy range was restricted to values between 295 meV and 305 meV. An application of a photodetector operating withing that energy range is in the environmental area with the production of carbonic gas detectors, since this gas absorbs at 300 meV.

6.3

Configuration and Parameterization of the Optimization Algorithms

For the four algorithms, the population size was equal to 50 and the number of generations equal to 100, totaling the maximum of 5,000 evaluations per run. As a measure of performance, the mean, minimum and maximum values of the oscillator strength of the best structures found in 10 runs were calculated. Tables 6.1-6.3 presents the parameters used for GA, PSO and CMA-ES (default parameters, from Yarpiz library (Heris, 2015) for MATLAB).

Table 6.1: Configuration of GA.

Parameters	Values
Selection	Stochastic uniform
Elitism count	1
Crossover rate	80%
Mutation rate	20%
Mutation	Adaptative feaseble*

* the default mutation function in MATLAB (MATLAB, 2016) when there are constraints, randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. The mutation chooses a direction and step length that satisfies bounds and linear constraints.

Table 6.2: Configuration of PSO.

Parameters	Values
Learning factors	2
Inertia weight	0.9

Table 6.3: Configuration of CMA-ES.

Parameters	Values
λ – Number of children to produce at each generation	50
Number of parents to keep from the λ children	25
Cumulation constant for step-size (cs)	0.7267

MG-PMA used the high-level crossover operator available from GPTIPS 2 (Searson, 2015). Table 6.4 presents the remaining parameters and configurations used by MG-PMA.

Table 6.4: Configuration of MG-PMA.

Parameters	Values
maximum tree depth d	7
Decimal ERC* rate	90%
Integer ERC* rate	0%
ERC* range	$[-1, +1]$
Low-level crossover rate	65%
Mutation rate	30%
Direct reproduction rate	5%
High-level crossover rate	50%
Elitism count	1
Lexicographic pressure (Luke, 2002)	Yes

* ERC - Ephemeral Random Constant: constants generated in a specific range by the user to constitute the trees of each individual together with variables (from the initial parameter values to be mapped by MG-PMA).

As described in Table 6.4, the probability of applying the mutation operator in the experiments was 30%. The three types of mutation (Searson, 2015) were employed with the following probabilities: ordinary sub-tree mutation (5%); switch an input terminal to another randomly selected input terminal (47.5%); and Gaussian perturbation of a randomly selected constant (47.5%, with the standard deviation of the Gaussian used equal to 10%). The initial population was generated using the ramped half-and-half method (Kora, 2017).

Domain constraint with PBC (presented in Section 3.3) was used in CMA-ES and MG-PMA.

6.4 Results

Table 6.5 presents the GA, PSO, CMA-ES and MG-PMA results for the structural optimization of the asymmetric superlattice. Analyzing the performance characterization, the four algorithms performed similarly:

- **maximum** – all tested algorithms found the best individual with same oscillator strength value;

- **mean** – on average, PSO, CMA-ES and MG-PMA were able to find the greater value of the oscillator strength;
- **minimum** – the worst structures found by PSO and CMA-ES had the same oscillator strength values of their best solutions;
- **standard deviation** – this characterization of performance shows that CMA-ES is the most stable among the algorithms tested, presenting a lower standard deviation. The second most stable algorithm is PSO, followed by MG-PMA and finally GA.

Table 6.5: Results of GA, PSO, CMA-ES and MG-PMA for the structural optimization of the QWIP. The table shows the maximum, mean, minimum, and standard deviation values of the oscillator strength of the best individuals in each of the 10 runs for both algorithms. Being the maximum value referring to the best individual and the minimum to the worst individual among the solutions presented in these 10 runs.

Method	Maximum (best individual)	Mean	Minimum (worst individual)	Standard Deviation
GA	0.35	0.34	0.34	$3.42e-3$
PSO	0.35	0.35	0.35	$7.07e-5$
CMA-ES	0.35	0.35	0.35	$1.01e-6$
MG-PMA	0.35	0.35	0.34	$9.99e-4$

Figure 6.5 shows a mean of the evolution of CMA-ES and MG-PMA to the best individuals over the generations in 10 runs and the oscillator strength value from the structure proposed by Penello (Penello, 2018). It is possible to note that these two algorithms were able to improve the value of the oscillator strength of the initial structure, from 0.22 to 0.35 and CMA-ES converges, on average, faster than the MG-PMA. GA and PSO did not save the evolution history of the solutions, but during the optimization, the convergence of the algorithms was obtained between the generations 20th and 40th.

Figure 6.6 shows the potential profile for the conduction band superimposed with the squared modulus of the wave function and oscillator strength *versus* energy for the best structures found among the solutions obtained by GA, PSO, CMA-ES and MG-PMA in 10 runs. The ground state and the state located in the continuum for the energy transition between 295 meV and 305 meV is highlighted in Figure 6.6 (in blue).

The algorithms found the following values of a , b and c :

- GA: $a = 1.45$ nm, $b = 7.92$ nm and $c = 2.66$ nm;
- PSO: $a = 1.45$ nm, $b = 7.99$ nm and $c = 2.67$ nm;
- CMA-ES: $a = 1.45$ nm, $b = 7.99$ nm and $c = 2.67$ nm;

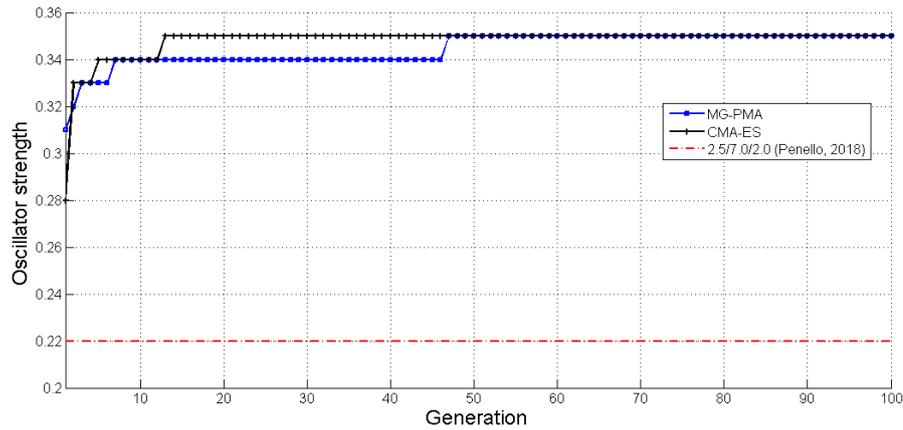


Figure 6.5: Evolution curve of the mean best values of the oscillator strength in 10 runs performed by MG-PMA (in blue) and CMA-ES (in black). The red line is the oscillator strength value of the structure proposed by Penello (Penello, 2018).

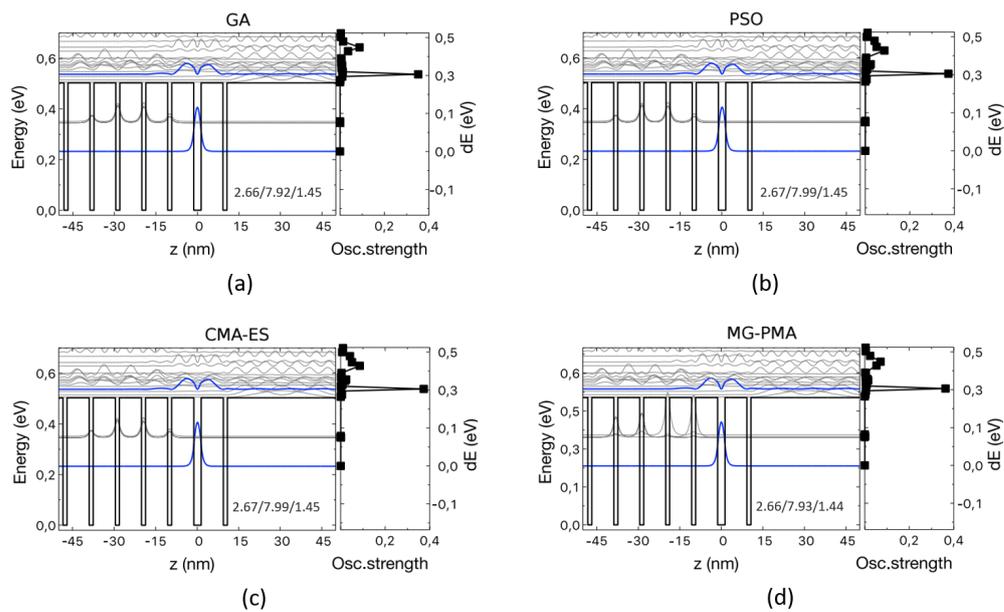


Figure 6.6: Potential profile of the best structures found among the solutions obtained, in 10 runs, by: (a) GA; (b) PSO; (c) CMA-ES; and (d) MG-PMA. The potential profile shows the modulus squared wave functions and oscillator strength *versus* energy.

– MG-PMA: $a = 1.44$ nm, $b = 7.93$ nm and $c = 2.66$ nm.

Comparing the four structures found by the algorithms, it is possible to note that the structures are practically the same, with differences in the second

decimal place (precision not reached during the production of these structures). These structures present the wave function of the state located in the continuum more confined in the QW defect when compared to the wave function of the initial structure (Figure 6.3). It is also possible to observe that the optimized structures present greater oscillator strength values which shows the efficiency of the algorithms in finding satisfactory solutions for this optimization problem.

All optimization algorithms were able to optimize this initial structure leading to new structures with greater oscillator strength values. These theoretically predicted structures can be synthesized and tested in order to validate this computational procedure. Assuming that the theoretical framework implemented in the simulator is correct, this study shows the importance of optimization algorithms in the design of new structures, replacing trial and error by intelligent design. Moreover, it is important to stress that only small variations were allowed with respect to the initial conditions. By assuming more flexibility, one could achieve structures with even better properties.

7

Conclusions and Future Work

7.1

Conclusions

In the year 2007, the development of an approach of the CGP to solve real-valued optimization problems, by Walker and Miller, has widened the road of optimization techniques by introducing a new category of methods, termed in this thesis as: optimization by genetic programming (OGP). In 2008, Pujol and Poly developed PMA that can be considered the first way of using the canonical GP in OGP. Koshiyama extends PMA to a more general form, through MGPP, that is able to optimize certain benchmark functions without the need of specify the domain (Koshiyama, 2013). In this form, an arbitrary number k of equations (instead of only one, in PMA) is used to map the optimizable parameters during the search for optimal values. In this thesis, Koshiyama's method has been thoroughly studied, its code has been revised and updated and it was renamed to multi-gene parameter mapping approach (MG-PMA).

In summary, the first results of this thesis show that MG-PMA is a flexible and robust optimization method that can be applied to functions of varying dimensionality and complexity. The performance of the method depends on the number k and depth d of trees (the representation of GP programs in this method). With $k = n$, which is the maximum number of trees (n corresponding to the number of parameters to be optimized), MG-PMA tends to perform better for more complex functions. A future work of automatic configuration of k must be done, since the choice of the value of this parameter is an NP-hard problem and a sub-optimal solution can further improve MG-PMA performance. Performance also strongly depends upon the maximum allowed tree depth d , and usually improves upon increasing d (note that the use of deeper trees also implies higher computational costs).

This work also presented a normalization for domain constraint through the periodic boundary conditions (PBC) used by molecular modeling. PBC can be used for the output values of a GP tree and for other constrained optimization algorithms.

MG-PMA with feedback was presented to avoid premature convergence to local optima due to the generation of a new mapping scheme. This approach was

compared with the MG-PMA by varying the value of k . MG-PMA without the feedback was the best algorithm for the CEC'2015 benchmark functions. The updating of the initial parameters to be mapped has a destructive effect on the evolution of GP programs. This is because the output of the trees of the best program of a given generation replaces a random column of the initial parameters. In this way, programs that used this column that was updated end up being changed. Future work should be considered in order to study the destructive effect of the feedback approach.

The change in tournament size was also evaluated and revealed an improvement factor for MG-PMA, increasing the stochastic character of the algorithm. Other high-level crossover operators were also used, being two of them proposed in this thesis (high-level single-point and high-level uniform), but there was no improvement for the benchmark tested. Finally, MG-PMA was compared with PSO and the winning methods of CEC'2015. Because its parameters were not tuned using optimization, but rather through a systematic analysis to understand the effect of each parameter, for the CEC'2015, MG-PMA was not able to outperform the algorithms that won the competition. However, the algorithm performed similarly to PSO, which indicates that MG-PMA is a promising strategy for complex optimization problems.

The analysis of the MG-PMA and MG-PMA with feedback was limited to exploring several ways of representing the individuals, the variation of the tournament size and the use of other high-level crossover operators. Better results might be reached by varying other parameters, such as the mathematical operators, the configuration of genetic operators (e.g. elitism, mutation probabilities).

This thesis also presented a new method using MGGP to multimodal numerical optimization problem, called Niching MG-PMA. The new algorithm used niching techniques based on the clearing procedure to maintain the population diversity, in order to perform a multivariate mapping among initial guesses to optimal parameters for a system. A set with 20 benchmark multimodal functions with different characteristics and difficulty levels was considered.

From statistical and qualitative analysis of the high-level crossover operator, the local optimization frequency and the feedback frequency it was possible to find two configurations for Niching MG-PMA: (i) Niching MG-PMA L1 - Niching MG-PMA using the high-level single-point crossover operator, local optimization frequency of 1 and feedback frequency of 1; and (ii) Niching MG-PMA L150 - Niching MG-PMA using the high-level single-point crossover operator, local optimization frequency of 150 and feedback frequency of 1.

These two configurations of Niching MG-PMA were compared to two well-established DE niching algorithms (DE/nrand/1/bin and Crowding DE/rand/1/bin)

in five levels of accuracy. Statistical analysis showed that for accuracy of $1e-01$, Niching MG-PMA L150 has significantly better than DE/rand/1/bin, showing potential usage for real-world problems that demand a multitude of solutions.

Finally, an MG-PMA application was presented for a real-world problem in the field of semiconductor nanodevices. The algorithm proposed in this thesis was able to perform the structural optimization of a QWIP, finding structures with oscillator strength superior to the developed structures in the area. In the comparison with other algorithms (GA, PSO and CMA-ES), all algorithms had similar performances finding the same solution. Comparing MG-PMA with GA, both algorithms had similar performances with a small difference in the minimum and standard deviation values of the oscillator strength of the best structures found. In these cases, MG-PMA is slightly better than GA. However, PSO and CMA-ES were even more stable, finding structures with equal oscillator strength in all simulations (equal values for maximum, minimum and mean).

The use of the evolutionary computation (EC) algorithms to determine a heterostructure with a desired electronic energy configuration opens a new path for the specialists in the field of the semiconductor nanodevices without the need to waste time on trial and error methods to find a desired structure. From the results obtained in this thesis, the use of these algorithms is validated not only to optimize known structures but also to design new structures with different numbers of wells, barriers and defects, starting from random dimensions. This allows the use of computational intelligence to design new structures that can be studied and fabricated by specialists with lower cost and greater efficiency.

7.2

Future Work

From the results obtained in this thesis, new paths for the OGP area are open, a topic rarely explored in EC, with projects of improvement of the algorithms and their applications:

- Application of optimization methods to tune MG-PMA, MG-PMA with feedback and Niching MG-PMA parameters for the purpose of making them competitive for IEEE CEC competitions.
- Development of an automatic selection of k values during the evolution process in order to solve the problem of arbitrarily choosing k thus increasing the flexibility of the methods (NP-hard problem).
- Study of the destructive effect of the MG-PMA with feedback.
- Application of other niching techniques (such as fitness sharing (Goldberg, 1987), crowding methods (De Jong, 1975, Mahfoud, 1992)

- and restricted tournament selection (Harik, 1995) with MG-PMA, increasing and enriching the area of multimodal optimization by genetic programming.
- Growth of the optimized semiconductor heterostructure found in this thesis in the laboratory.
 - Design, optimize and grow new QWIPs, varying the wells and barriers in size, position and quantity.
 - Optimization of photodetector materials to increase the accuracy of electronic state control and decrease the cost of the production process.

Bibliography

- [Aaronson, 2005] AARONSON, S.. **NP-complete problems and physical reality**. Submitted to: Sigact News, 2005.
- [Alcalá-Fdez, 2011] ALCALÁ-FDEZ, J.; FERNÁNDEZ, A.; LUENGO, J.; DERRAC, J.; GARCÍA, S.. **Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework**. Multiple-Valued Logic and Soft Computing, 17(2-3):255–287, 2011.
- [Ali, 2014] ALI, M. Z.; AWAD, N. H.. **A novel class of niche hybrid cultural algorithms for continuous engineering optimization**. Information Sciences, 267:158–190, 2014.
- [Angeline, 1997] ANGELINE, P. J.. **Subtree crossover: Building block engine or macromutation?** In: Koza, J. R.; Deb, K.; Dorigo, M.; Fogel, D. B.; Garzon, M.; Iba, H. ; Riolo, R. L., editors, GENETIC PROGRAMMING 1997: PROCEEDINGS OF THE SECOND ANNUAL CONFERENCE, p. 9–17, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.
- [Angus, 2009] ANGUS, D.. **Niching for Ant Colony Optimisation**, p. 165–188. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [Awad, 2015] AWAD, N.; ALI, M. Z.; REYNOLDS, R. G.. **A differential evolution algorithm with success-based parameter adaptation for cec2015 learning-based optimization**. In: 2015 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 1098–1105, May 2015.
- [Back, 1997] BACK, T.; HAMMEL, U.; SCHWEFEL, H. -P.. **Evolutionary computation: Comments on the history and current state**. Trans. Evol. Comp, 1(1):3–17, apr 1997.
- [Banzhaf, 1993] BANZHAF, W.. **Genetic programming for pedestrians**. In: PROCEEDINGS OF THE 5TH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, p. 628–, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

- [Banzhaf, 1998] BANZHAF, W.; FRANCONI, F. D.; KELLER, R. E. AND NORDIN, P.. **Genetic Programming: An Introduction: on the Automatic Evolution of Computer Programs and Its Applications**. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [Beasley, 1993] BEASLEY, D.; BULL, D. R.; MARTIN, R. R.. **A sequential niche technique for multimodal function optimization**. *Evolutionary Computation*, 1(2):101–125, June 1993.
- [Ben-Israel, 1966] BEN-ISRAEL, A.. **A newton-raphson method for the solution of systems of equations**. *Journal of Mathematical Analysis and Applications*, 15(2):243–252, 1966.
- [Bessaou, 2000] BESSAOU, M.; PÉTROWSKI, A.; SIARRY, P.. **Island Model Cooperating with Speciation for Multimodal Optimization**, p. 437–446. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.
- [Bird, 2006] BIRD, S.; LI, X.. **Adaptively choosing niching parameters in a pso**. In: PROCEEDINGS OF THE 8TH ANNUAL CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTATION, GECCO '06, p. 3–10, New York, NY, USA, 2006. ACM.
- [Bird, 2010] BIRD, S.; LI, X.. **Improving Local Convergence in Particle Swarms by Fitness Approximation Using Regression**, p. 265–293. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [Changra, 2009] CHANDRA, S.; MEHRA, A.. **Numerical Optimization with Applications**. Alpha Science International, 2009.
- [Chelouah, 2003] CHELOUAH, R.; SIARRY, P.. **Genetic and nelder–mead algorithms hybridized for a more accurate global optimization of continuous multimodal functions**. *European Journal of Operational Research*, 148(2):335–348, 2003. Sport and Computers.
- [Chen, 2011] CHEN, X.; ONG, Y. S.; LIM, M. H.; TAN, K. C.. **A multi-facet survey on memetic computation**. *IEEE Transactions on Evolutionary Computation*, 15(5):591–607, Oct 2011.
- [Clauser, 1974] CLAUSER, J. F.. **Experimental distinction between the quantum and classical field-theoretic predictions for the photoelectric effect**. *Phys. Rev. D*, 9:853–860, Feb 1974.
- [Cortes, 2004] CORTES, O.; SAAVEDRA, O.. **Estratégias evolutivas paralelas em otimização multimodal**. *INFOCOMP Journal of Computer Science*, 2(1):63–68, 2004.

- [Cramer, 1985] CRAMER, N. L.. **A representation for the adaptive generation of simple sequential programs.** In: PROCEEDINGS OF THE 1ST INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, p. 183–187, Hillsdale, NJ, USA, 1985. L. Erlbaum Associates Inc.
- [De Jong, 1975] DE JONG, KENNETH ALAN. **An Analysis of the Behavior of a Class of Genetic Adaptive Systems.** PhD thesis, Ann Arbor, MI, USA, 1975. AAI7609381.
- [Deb, 2012] DEB, K.; SAHA, A.. **Multimodal optimization using a bi-objective evolutionary algorithm.** Evolutionary Computation, 20(1):27–62, March 2012.
- [Delibasis, 2010] DELIBASIS, K.; ASVESTAS, P. A.; MATSOPOULOS, G. K.. **Multimodal genetic algorithms-based algorithm for automatic point correspondence.** Pattern Recognition, 43(12):4011–4027, 2010.
- [Derrac, 2011] DERRAC, J.; GARCÍA, S.; MOLINA, D.; HERRERA, F.. **A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms.** Swarm and Evolutionary Computation, 1(1):3 – 18, 2011.
- [Dick, 2010] DICK, G.. **Automatic identification of the niche radius using spatially-structured clearing methods.** In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, p. 1–8, July 2010.
- [Eberhart, 1995] EBERHART, R.; KENNEDY, J.. **A new optimizer using particle swarm theory.** In: MICRO MACHINE AND HUMAN SCIENCE, 1995. MHS '95., PROCEEDINGS OF THE SIXTH INTERNATIONAL SYMPOSIUM ON, p. 39–43, Oct 1995.
- [Eiben, 2002] EIBEN, A. E.; SCHOENAUER, M.. **Evolutionary computing.** Information Processing Letters, 82(1):1–6, 2002.
- [Einstein, 1905] EINSTEIN, A.. **Über einen die erzeugung und verwandlung des liches betreffenden heuristischen gesichtspunkt.** Annalen der Physik, 322(6):132–148, 1905.
- [Ellefsen, 2017] ELLEFSEN, K. O.; LEPIKSON, H. A.; ALBIEZ, J. C.. **Multiobjective coverage path planning: Enabling automated inspection of complex, real-world structures.** Applied Soft Computing, 61:264–282, 2017.

- [Engelbrecht, 2002] ENGELBRECHT, A.. **Computational Intelligence: An Introduction**. Halsted Press, New York, NY, USA, 2002.
- [Epitropakis, 2011] EPITROPAKIS, M. G.; PLAGIANAKOS, V. P.; VRAHATIS, M. N.. **Finding multiple global optima exploiting differential evolution's niching capability**. In: 2011 IEEE SYMPOSIUM ON DIFFERENTIAL EVOLUTION (SDE), p. 1–8, April 2011.
- [Espejo, 2010] ESPEJO, P. G.; VENTURA, S.; HERRERA, F.. **A survey on the application of genetic programming to classification**. IEEE Transactions on Systems, Man, and Cybernetics-part C: applications and Reviews, 4(2):121–144, 3 2010.
- [Fieldsend, 2014] FIELDSSEND, J. E.. **Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser**. In: 2014 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 2593–2600, July 2014.
- [Fletcher, 1964] FLETCHER, R.; REEVES, C. M.. **Function minimization by conjugate gradient**. The Computer Journal, 7(2):149–154, 1964.
- [Fogel, 1966] FOGEL, L. J.; OWENS, A. J.; WALSH, M. J.. **Artificial Intelligence Through Simulated Evolution**. John Wiley & Sons, 1966.
- [Fogel, 1998] FOGEL, D. B.. **Evolutionary Computation: The Fossil Record**. Wiley-IEEE Press, 1st edition, 1998.
- [Forrest, 1993] FORREST, S.; JAVORNIK, B.; SMITH, R. E.; PERELSON, A. S.. **Using genetic algorithms to explore pattern recognition in the immune system**. Evolutionary Computation, 1(3):191–211, 1993.
- [Forsyth, 1981] FORSYTH, R.. **Beagle a darwinian approach to pattern recognition**. Kybernetes, 10(3):159–166, 1981.
- [Friedberg, 1958] FRIEDBERG, R. M.. **A learning machine: Part i**. IBM J. Res. Dev., 2(1):2–13, jan 1958.
- [Friedberg, 1959] FRIEDBERG, R. M.; DUNHAM, B.; NORTH, J. H.. **A learning machine: Part ii**. IBM J. Res. Dev., 3(3):282–287, jul 1959.
- [Gan, 2001] GAN, J.; WARWICK, K.. **Dynamic niche clustering: a fuzzy variable radius niching technique for multimodal optimisation in gas**. In: PROCEEDINGS OF THE 2001 CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE CAT. NO.01TH8546), volumen 1, p. 215–222 vol. 1, 2001.

- [Garey, 1979] GAREY, M. R.; JOHNSON, D. S.. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. W. H. Freeman & Co., San Francisco, 1979.
- [Glibovets, 2013] GLIBOVETS, N. N.; GULAYEVA, N. M.. **A review of niching genetic algorithms for multimodal function optimization**. *Cybern Syst Anal*, 49:815–820, November 2013.
- [Goldberg, 1987] GOLDBERG, D. E.; RICHARDSON, J.. **Genetic algorithms with sharing for multimodal function optimization**. In: PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS ON GENETIC ALGORITHMS AND THEIR APPLICATION, p. 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.
- [Guntsch, 2002] GUNTSCH, M.; MIDDENDORF, M.. **Applying Population Based ACO to Dynamic Optimization Problems**, p. 111–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
- [Guo, 2015] GUO, S. M.; TSAI, J. S. H.; YANG, C. C.; HSU, P. H.. **A self-optimization approach for l-shade incorporated with eigenvector-based crossover and successful-parent-selecting framework on cec 2015 benchmark set**. In: 2015 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 1003–1010, May 2015.
- [Hannah, 2014] HANNAH, L. A.. **Stochastic optimization**, 2014.
- [Hansen, 2001] HANSEN, N.; OSTERMEIER, A.. **Completely derandomized self-adaptation in evolution strategies**. *Evolutionary Computation*, 9(2):159–195, 2001.
- [Hansen, 2006] HANSEN, N.. **The CMA Evolution Strategy: A Comparing Review**, p. 75–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [Harik, 1995] HARIK, G. R.. **Finding multimodal solutions using restricted tournament selection**. In: PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, p. 24–31, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [Harman, 2013] HARMAN, M.; LANGDON, W. B.; WEIMER, W.. **Genetic programming for reverse engineering**. In: 2013 20TH WORKING CONFERENCE ON REVERSE ENGINEERING (WCRE), p. 1–10, Oct 2013.
- [Harman, 2014] HARMAN, M.; JIA, Y.; LANGDON, W. B.; PETKE, J.; MOGHADAM, I. H.; YOO, S.; WU, F.. **Genetic improvement for**

- adaptive software engineering (keynote)**. In: PROCEEDINGS OF THE 9TH INTERNATIONAL SYMPOSIUM ON SOFTWARE ENGINEERING FOR ADAPTIVE AND SELF-MANAGING SYSTEMS, SEAMS 2014, p. 1–4, New York, NY, USA, 2014. ACM.
- [Heris, 2015] HERIS, S. M. K.. **Ypea108: Covariance matrix adaptation evolution strategy (cma-es)**. <http://www.yarpiz.com>, 2015.
- [Hinchliffe, 1996] HINCHLIFFE, M. P.; WILLIS, M. J.; HIDDEN, H.; THAM, M. T.; MCKAY, B.; BARTON, G. W.. **Modelling chemical process systems using a multi-gene genetic programming algorithm**. In: GENETIC PROGRAMMING: PROCEEDINGS OF THE FIRST ANNUAL CONFERENCE, p. 56–65. MIT Press, 1996.
- [Holland, 1975] HOLLAND, J.. **Adaptation in Natural and Artificial Systems**. MIT Press, Cambridge, 1975.
- [Holm, 1979] HOLM, S.. **A simple sequentially rejective multiple test procedure**. Scandinavian Journal of Statistics, 6:65–70, 1979.
- [Iman, 1980] IMAN, R. L.; DAVENPORT, J. M.. **Approximations of the critical region of the Friedman statistic**. Communications in Statistics - Theory and Methods, 9(6):571–595, Jan. 1980.
- [Islam, 2017] ISLAM, MD. J.; LI, X.; DEB, K.. **Multimodal truss structure design using bilevel and niching based evolutionary algorithms**. In: PROCEEDINGS OF THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, GECCO '17, p. 274–281, New York, NY, USA, 2017. ACM.
- [Jain, 2018] JAIN, I.; JAIN, V. K.; JAIN, R.. **Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification**. Applied Soft Computing, 62:203–215, 2018.
- [Jelasity, 1998] JELASITY, M.; DOMBI, J.. **Gas, a concept on modeling species in genetic algorithms**. Artificial Intelligence, 99(1):1 – 19, 1998.
- [Kora, 2017] KORA, P.; YADLAPALLI, P.. **Crossover operators in genetic algorithms: A review**. International Journal of Computer Applications, 162(10):34–36, Mar 2017.
- [Koshiyama, 2013] KOSHIYAMA, A. S.; DIAS, D. M.; ABS DA CRUZ, A. V.; PACHECO, M. A. C.. **Numerical optimization by multi-gene genetic programming**. In: PROCEEDINGS OF THE 15TH ANNUAL

CONFERENCE COMPANION ON GENETIC AND EVOLUTIONARY COMPUTATION, GECCO '13 Companion, p. 145–146, New York, NY, USA, 2013. ACM.

[Koza, 1992] KOZA, J. R.. **Genetic Programming: On the Programming of Computers by Means of Natural Selection**. MIT Press, Cambridge, MA, USA, 1992.

[Kruisselbrink, 2009] KRUISELBRINK, J. W.; ALEMAN, A.; EMMERICH, M. T. M.; IJZERMAN, AD P.; BENDER, A.; BAECK, T.; VAN DER HORST, E.. **Enhancing search space diversity in multi-objective evolutionary drug molecule design using niching**. In: PROCEEDINGS OF THE 11TH ANNUAL CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTATION, GECCO '09, p. 217–224, New York, NY, USA, 2009. ACM.

[Langdon, 2002] LANGDON, W. B.; POLI, R.. **Foundations of Genetic Programming**. Springer-Verlag, Berlin, 2002.

[Langdon, 2015] LANGDON, W. B.. **Genetic Improvement of Software for Multiple Objectives**, p. 12–28. Springer International Publishing, Cham, 2015.

[Langdon, 2017] LANGDON, W. B.; LAM, B. Y. H.; MODAT, M.; PETKE, J.; HARMAN, M.. **Genetic improvement of gpu software**. Genetic Programming and Evolvable Machines, 18(1):5–44, Mar 2017.

[Leach, 2001] LEACH, A. R.. **Molecular Modelling: Principles and Applications**. Pearson Education. Prentice Hall, 2001.

[Levine, 1993] LEVINE, B. F.. **Quantum-well infrared photodetectors**. Journal of Applied Physics, 74(8):R1–R81, 1993.

[Leyton-Brown, 2014] LEYTON-BROWN, K.; HOOS, H. H.; HUTTER, F.; XU, L.. **Understanding the empirical hardness of np-complete problems**. Commun. ACM, 57(5):98–107, May 2014.

[Li, 2002] LI, J-P.; BALAZS, M. E.; PARKS, G. T.; CLARKSON, P. J.. **A species conserving genetic algorithm for multimodal function optimization**. Evolutionary Computation, 10(3):207–234, 2002.

[Li, 2013] LI, X.; ENGELBRECHT, A.; EPITROPAKIS, M. G.. **Benchmark functions for cec'2013 special session and competition on niching methods for multimodal function optimization**. 2013.

- [Li, 2016] LI, X.; EPITROPAKIS, M. G.; DEB, K.; ENGELBRECHT, A.. **Seeking multiple solutions: An updated survey on niching methods and their applications**. IEEE Transactions on Evolutionary Computation, 21(4):518–538, Aug 2017.
- [Liang, 2014] LIANG, J. J.; QU, B. Y.; SUGANTHAN, P. N.; Q. CHEN. **Problem definitions and evaluation criteria for the cec2015 competition on learning-based real parameter single objective optimization**. Technical Report 201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, November 2014.
- [Liberti, 2005] LIBERTI, L.; KUCHERENKO, S.. **Comparison of deterministic and stochastic approaches to global optimization**. International Transactions in Operational Research, 12(3):263–285, 2005.
- [Liu, 2016] LIU, Y.; CHENG, Z.; XU, J.; YANG, J.; WANG, Q.. **Improvement and validation of genetic programming symbolic regression technique of silva and applications in deriving heat transfer correlations**. Heat Transfer Engineering, 37(10):862–874, 5 2016.
- [Luh, 2011] LUH, G.-C.; LIN, C.-Y.. **Optimal design of truss-structures using particle swarm optimization**. Computers & Structures, 89(23):2221–2232, 2011.
- [Luke, 2002] LUKE, S.; PANAIT, L.. **Lexicographic parsimony pressure**. In: PROCEEDINGS OF THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE (GECCO 2002), p. 829–836. Morgan Kaufmann Publishers, 2002.
- [MATLAB, 2016] MATLAB. **R2016a**. The MathWorks Inc., Natick, Massachusetts, 2016.
- [Mahfoud, 1992] MAHFOUD, S. W.. **Crowding and preselection revisited**. Parallel Problem Solving from Nature, 2, p. 27–36, 1992.
- [McCarthy, 1960] MCCARTHY, J.. **Recursive functions of symbolic expressions and their computation by machine, part i**. Commun. ACM, 3(4):184–195, apr 1960.
- [Michalewicz, 1996] MICHALEWICZ, Z.. **Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.)**. Springer-Verlag, London, UK, UK, 1996.
- [Miller, 2000] MILLER, J. F.; THOMSON, P.. **Cartesian Genetic Programming**, p. 121–132. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

- [Miller, 2011] MILLER, J.. **Cartesian Genetic Programming**. Natural Computing Series. Springer Berlin Heidelberg, 2011.
- [Miller, 2013] MILLER, J. F.; MOHID, M.. **Function optimization using cartesian genetic programming**. In: PROCEEDINGS OF THE 15TH ANNUAL CONFERENCE COMPANION ON GENETIC AND EVOLUTIONARY COMPUTATION, GECCO '13 Companion, p. 147–148, New York, NY, USA, 2013. ACM.
- [Molina, 2017] MOLINA, D.; MORENO-GARCÍA, F.; HERRERA, F.. **Analysis among winners of different ieeec competitions on real-parameters optimization: Is there always improvement?** In: 2017 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 805–812, June 2017.
- [Myszkowski, 2018] MYSZKOWSKI, P. B.; OLECH, Ł. P.; LASZCZYK, M.; SKOWROŃSKI, MAREK E.. **Hybrid differential evolution and greedy algorithm (degr) for solving multi-skill resource-constrained project scheduling problem**. Applied Soft Computing, 62:1–14, 2018.
- [O'Neill, 2003] O'NEILL, M.; RYAN, C.. **Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language**. Kluwer Academic Publishers, Norwell, MA, USA, 2003.
- [Ono, 2007] ONO, S.; HIROTANI, Y.; NAKAYAMA, S.. **Multiple solution search based on hybridization of real-coded evolutionary algorithm and quasi-newton method**. In: 2007 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, p. 1133–1140, Sept 2007.
- [Parsopoulos, 2001] PARSOPOULOS, K. E.; PLAGIANAKOS, V. P.; MAGOULAS, G. D.; VRAHATIS, M. N.. **Objective function “stretching” to alleviate convergence to local minima**. Nonlinear Analysis: Theory, Methods & Applications, 47(5):3419–3424, 2001. Proceedings of the Third World Congress of Nonlinear Analysts.
- [Parsopoulos, 2004] PARSOPOULOS, K. E.; VRAHATIS, M. N.. **On the computation of all global minimizers through particle swarm optimization**. IEEE Transactions on Evolutionary Computation, 8(3):211–224, June 2004.
- [Penello, 2013] PENELLO, G. M.. **Investigação teórica e experimental do papel de super-redes em fotodetectores de infravermelho médio baseado em poços quânticos**. PhD thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, August 2013.

- [Penello, 2018] PENELLO, G. M.; PEREIRA, P. H.; PIRES, M. P.; SIVCO, D.; GMACHL, C.; SOUZA, P. L.. **Leaky electronic states for photovoltaic photodetectors based on asymmetric superlattices**. Applied Physics Letters, 112(3):033503, 2018.
- [Peng, 2002] PENG, J. X.; THOMPSON, S.; LI, K.. **A gradient-guided niching method in genetic algorithm for solving continuous optimisation problems**. In: PROCEEDINGS OF THE 4TH WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION (CAT. NO.02EX527), volumen 4, p. 3333–3338 vol.4, 2002.
- [Pérez, 2012] PÉREZ, E.; POSADA, M.; HERRERA, F.. **Analysis of new niching genetic algorithms for finding multiple solutions in the job shop scheduling**. Journal of Intelligent Manufacturing, 23:341–356, 2012.
- [Pérez, 2016] PÉREZ, E.; POSADA, M.; LORENZANA, A.. **Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms**. Soft Computing, 20:1879–1896, 2016.
- [Petrowski, 1996] PETROWSKI, A.. **A clearing procedure as a niching method for genetic algorithms**. In: PROCEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION, p. 798–803, May 1996.
- [Poli, 1996] POLI, R.. **Parallel distributed genetic programming**. Technical report, School of Computer Science, University of Birmingham, 1996.
- [Poli, 2008] POLI, R.; LANGDON, W. B.; MCPHEE, N. F.. **A Field Guide to Genetic Programming**. Lulu.com, Raleigh, 2008.
- [Preuss, 2012] PREUSS, M.; BURELLI, P.; YANNAKAKIS, G. N.. **Diversified virtual camera composition**. In: APPLICATIONS OF EVOLUTIONARY COMPUTATION - EVOAPPLICATIONS 2012: EVOCOMNET, EVOCOMPLEX, EVOFIN, EVOGAMES, EVOHOT, EVOIASP, EVONUM, EVOPAR, EVORISK, EVOSTIM, AND EVOSTOC, MÁLAGA, SPAIN, APRIL 11-13, 2012, PROCEEDINGS, p. 265–274, 2012.
- [Preuss, 2015] PREUSS, M.. **Multimodal Optimization by Means of Evolutionary Algorithms**. Springer Publishing Company, Incorporated, 1st edition, 2015.

- [Pujol, 2008] PUJOL, J. C. F.; POLI R.. **Parameter mapping: A genetic programming approach to function optimization**. International Journal of Knowledge-Based and Intelligent Engineering Systems, 12(1):29–45, 2008.
- [Rechenberg, 1973] RECHENBERG, I.. **Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution**. Número 15 em Problemata. Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1973.
- [Rueda, 2015] RUEDA, J. L.; ERLICH, I.. **Testing mvmo on learning-based real-parameter single objective benchmark optimization problems**. In: 2015 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 1025–1032, May 2015.
- [Sallam, 2015] SALLAM, K. M.; SARKER, R. A.; ESSAM, D. L.; ELSAYED, S. M.. **Neurodynamic differential evolution algorithm and solving cec2015 competition problems**. In: 2015 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 1033–1040, May 2015.
- [Sareni, 1998] SARENI, B.; KRAHENBUHL, L.; NICOLAS, A.. **Niching genetic algorithms for optimization in electromagnetics. i. fundamentals**. IEEE Transactions on Magnetics, 34(5):2984–2987, Sep 1998.
- [Schmidhuber, 1987] SCHMIDHUBER, J.. **Evolutionary principles in self-referential learning**. Diploma thesis, Technische Universität München, Germany, may 1987.
- [Schwefel, 1995] SCHWEFEL, H. P.. **Evolution and Optimum Seeking**. Sixth Generation Computer Technologies. Wiley, 1995.
- [Searson, 2010] SEARSON, D. P.; LEAHY, D.E.; WILLIS, M. J.. **Gptips: an open source genetic programming toolbox for multigene symbolic regression**. In: PROCEEDINGS OF THE INTERNATIONAL MULTICONFERENCE OF ENGINEERS AND COMPUTER SCIENTISTS 2010 (IMECS 2010), Hong Kong, 2010.
- [Searson, 2015] SEARSON, D. P.. **Gptips 2: an open-source software platform for symbolic data mining**. 2015.
- [Shi, 1998] SHI, Y.; EBERHART, R.. **A modified particle swarm optimizer**. In: 1998 IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION PROCEEDINGS. IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE (CAT. NO.98TH8360), p. 69–73, May 1998.

- [Shir, 2008] SHIR, O. M.. **Niching in derandomized evolution strategies and its applications in quantum control**. PhD thesis, Natural Computing Group, LIACS, Faculty of Science, Leiden University, 6 2008.
- [Silva, 2014] SILVA, M.; KOSHIYAMA, A.; VELLASCO, M.; CATALDO, E.. **Evolutionary features and parameter optimization of spiking neural networks for unsupervised learning**. p. 2391–2398, Beijing, 7 2014. 2014 International Joint Conference on Neural Networks (IJCNN), IEEE.
- [Smith, 1980] SMITH, S. F.. **A Learning System Based on Genetic Adaptive Algorithms**. PhD thesis, Pittsburgh, PA, USA, 1980. AAI8112638.
- [Spears, 1993] SPEARS, W. M.; DE JONG, K. A.; BÄCK, T.; FOGEL, D. B., DE GARIS, H.. **An Overview of Evolutionary Computation**. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.
- [Spector, 2002] SPECTOR, LEE ;ROBINSON, A.. **Genetic programming and autoconstructive evolution with the push programming language**. Genetic Programming and Evolvable Machines, 3(1):7–40, 2002.
- [Storn, 1995] STORN, R.; PRICE, K.. **Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces**. Technical report, International Computer Science Institute, Berkeley, 1995.
- [Tan, 2017] TAN, C.; CHANG, S.; LIU, L.. **Hierarchical genetic-particle swarm optimization for bistable permanent magnet actuators**. Applied Soft Computing, 61:1–7, 2017.
- [Thomsen, 2004] THOMSEN, R.. **Multimodal optimization using crowding-based differential evolution**. In: PROCEEDINGS OF THE 2004 CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE CAT. NO.04TH8753), volumen 2, p. 1382–1389 Vol.2, June 2004.
- [Ursem, 1999] URSEM, R. K.. **Multinational evolutionary algorithms**. In: PROCEEDINGS OF THE 1999 CONGRESS ON EVOLUTIONARY COMPUTATION-CEC99 (CAT. NO. 99TH8406), volumen 3, p. 1640 Vol. 3, 1999.
- [Vilela Neto, 2012] VILELA NETO, O. P. V.; PACHECO, M. A. C.. **Nanotecnologia computacional inteligente: concebendo a engenharia em nanotecnologia**. Interciência: PUC-Rio, Rio de Janeiro, 2012.
- [Walker, 2007] WALKER, J. A.; MILLER, J. F.. **Solving real-valued optimisation problems using cartesian genetic programming**. In: PROCEEDINGS OF

THE 9TH ANNUAL CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTATION, GECCO '07, p. 1724–1730, New York, NY, USA, 2007. ACM.

- [Wei, 2005] WEI, L.; ZHAO, M.. **A niche hybrid genetic algorithm for global optimization of continuous multimodal functions.** Applied Mathematics and Computation, 160(3):649–661, 2005.
- [Yang, 2005] YANG, R.; KALNIS, P.; TUNG, A. K. H.. **Similarity evaluation on tree-structured data.** In: PROCEEDINGS OF THE 2005 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, SIGMOD '05, p. 754–765, New York, NY, USA, 2005. ACM.
- [Yang, 2017] YANG, Q.; CHEN, W. N.; YU, Z.; GU, T.; LI, Y.; ZHANG, H.; ZHANG, J.. **Adaptive multimodal continuous ant colony optimization.** IEEE Transactions on Evolutionary Computation, 21(2):191–205, April 2017.
- [Yeboah-Antwi, 2017] YEBOAH-ANTWI, K.; BAUDRY, B.. **Online genetic improvement on the java virtual machine with ecslr.** Genetic Programming and Evolvable Machines, 18(1):83–109, Mar 2017.
- [Yin, 1993] YIN, X.; GERMAI, N.. **A Fast Genetic Algorithm with Sharing Scheme Using Cluster Analysis Methods in Multimodal Function Optimization,** p. 450–457. Springer Vienna, Vienna, 1993.
- [Yoshida, 2017] YOSHIDA, S.; HARADA, T.; THAWONMAS, R.. **Multimodal genetic programming by using tree structure similarity clustering.** In: 2017 IEEE 10TH INTERNATIONAL WORKSHOP ON COMPUTATIONAL INTELLIGENCE AND APPLICATIONS (IWCIA), p. 85–90, Nov 2017.
- [de Castro, 2002] DE CASTRO, L. N.; TIMMIS, J.. **An artificial immune network for multimodal function optimization.** In: EVOLUTIONARY COMPUTATION, 2002. CEC '02. PROCEEDINGS OF THE 2002 CONGRESS ON, volumen 1, p. 699–704, May 2002.

A

Analyses of the Multigenic Approach – Numerical Results

A1

Introduction

This appendix presents all the numerical results obtained in the experiments of the Chapter 4 of this thesis.

A2

Numerical Results of the Analysis of Multigenic Individuals

This section presents the numerical results for benchmark-based sensitivity analysis of the number of trees (k) as well as variations of the tree depth (d). Tables corresponding to test set 1 (Tables A1- A5) and test set 2 (Tables A6- A10) show the number of hits on the function global minimum (HITS), the mean number of evaluations (MNE), the median number of evaluations (Median), the minimum number of evaluations (Min), the maximum number of evaluations (Max), the standard deviation of evaluations (Std), the first quartile of evaluations (1st Quartile) and the third quartile of evaluations (3rd Quartile). Tables of test set 3 (Tables A13- A17) show the mean of the best fitness values (MBFV), the median of the best fitness values (Median), the minimum of the best fitness values (Min), the maximum of the best fitness values (Max), the standard deviation of the best fitness values (Std), the first quartile of the best fitness values (1st Quartile) and the third quartile of the best fitness values (3rd Quartile).

A2.1

Set 1

Table A1: Numerical results for the test set 1 with the tree depth (d) equal to 2.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	2	100	426.000	100.000	0.000	8200.000	1302.648	100.000	200.000
	$n/2$	2	100	788.000	400.000	100.000	6100.000	1014.369	300.000	900.000
	n	2	100	1103.000	650.000	300.000	24200.000	2654.763	500.000	850.000
f_2	1	2	100	46776.500	10000.000	0.000	739700.000	91480.119	350.000	64775.000
	n	2	1	150.000	150.000	150.000	150.000	0.000	150.000	150.000
f_3	1	2	100	244.000	100.000	0.000	2300.000	417.561	0.000	225.000
	$n/2$	2	100	156.500	100.000	0.000	1150.000	188.247	50.000	200.000
	n	2	100	151.000	100.000	0.000	2150.000	241.939	0.000	200.000
f_4	1	2	100	66.000	0.000	0.000	1800.000	190.279	0.000	100.000
	$n/2$	2	100	3193.000	3100.000	1600.000	5300.000	870.998	2500.000	3850.000
	n	2	100	6180.000	6200.000	2200.000	10800.000	1650.467	5000.000	7200.000
f_5	1	2	100	40363.000	31750.000	3100.000	166050.000	31408.007	17425.000	56150.000
	$n/2$	2	100	378184.000	313600.000	26000.000	1995250.000	317216.314	168800.000	488600.000
	n	2	100	440125.000	360725.000	15100.000	1771050.000	314084.844	217600.000	582150.000
f_6	1	2	100	54648.000	32400.000	0.000	282000.000	63476.999	7200.000	81000.000
	$n/2$	2	97	102496.907	65200.000	400.000	531200.000	114975.269	9500.000	151600.000
	n	2	99	80446.465	48800.000	800.000	560600.000	103709.283	11750.000	101500.000
f_7	1	2	100	293.000	100.000	0.000	6600.000	856.025	100.000	200.000
	$n/2$	2	100	1279.000	1050.000	400.000	8300.000	868.098	900.000	1350.000
	n	2	100	2590.000	2300.000	1000.000	10100.000	1276.398	1900.000	2900.000
f_8	1	2	100	13849.000	10550.000	100.000	55900.000	11856.580	4700.000	19150.000
	$n/2$	2	100	15916.000	9650.000	100.000	68600.000	15532.363	5200.000	22500.000
	n	2	100	11045.000	7350.000	0.000	58100.000	11179.214	2700.000	16650.000
f_9	1	2	100	15001.000	10600.000	200.000	80600.000	14480.370	5050.000	19500.000
	$n/2$	2	100	13379.000	9350.000	400.000	57400.000	13207.431	3550.000	19450.000
	n	2	100	11019.000	8200.000	200.000	49800.000	10692.192	2300.000	16800.000
f_{10}	1	2	100	11924.000	7800.000	100.000	70100.000	13397.679	3000.000	14350.000
	$n/2$	2	100	16266.000	10150.000	300.000	87000.000	18085.528	2300.000	22400.000
	n	2	100	10061.000	5200.000	300.000	52100.000	11680.424	1900.000	11950.000
f_{11}	1	2	100	11916.000	7650.000	100.000	79800.000	12574.680	3850.000	13950.000
	$n/2$	2	100	14495.000	10900.000	300.000	136100.000	17913.498	3450.000	18650.000
	n	2	100	10539.000	5600.000	500.000	61800.000	11850.993	2000.000	15400.000
f_{12}	1	2	100	14173.000	9950.000	500.000	69500.000	13527.352	4550.000	19500.000
	$n/2$	2	100	12412.000	8000.000	500.000	85300.000	14917.499	2150.000	16550.000
	n	2	100	7421.000	3800.000	600.000	55900.000	9341.547	1650.000	8450.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A2: Numerical results for the test set 1 with the tree depth (d) equal to 3.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	3	100	124.000	100.000	0.000	1700.000	190.226	100.000	100.000
	$n/2$	3	100	453.000	400.000	100.000	4600.000	510.784	300.000	400.000
	n	3	100	673.000	600.000	300.000	6900.000	681.035	500.000	700.000
f_2	1	3	100	53307.000	24725.000	50.000	437400.000	82367.857	700.000	63575.000
	$n/2$	3	0	–	–	–	–	–	–	–
	n	3	100	–	–	–	–	–	–	–
f_3	1	3	100	47.500	0.000	0.000	1150.000	127.004	0.000	50.000
	$n/2$	3	100	45.500	0.000	0.000	550.000	83.815	0.000	50.000
	n	3	100	39.000	0.000	0.000	200.000	53.927	0.000	100.000
f_4	1	3	100	33.000	0.000	0.000	100.000	47.258	0.000	100.000
	$n/2$	3	100	3100.000	3100.000	1600.000	5100.000	875.595	2400.000	3750.000
	n	3	100	6264.000	6200.000	3800.000	9700.000	1265.751	5300.000	6950.000
f_5	1	3	100	19835.000	15425.000	600.000	164400.000	20265.572	7350.000	26000.000
	$n/2$	3	98	194594.388	145750.000	8200.000	862000.000	162636.010	74050.000	254850.000
	n	3	100	216574.000	157100.000	15900.000	997200.000	202886.565	73425.000	265375.000
f_6	1	3	100	45290.000	10500.000	0.000	460800.000	76564.551	400.000	63300.000
	$n/2$	3	99	158579.798	64400.000	400.000	1563000.000	246482.867	800.000	217550.000
	n	3	99	170652.525	46800.000	1200.000	1031200.000	238011.414	2050.000	292650.000
f_7	1	3	100	95.000	100.000	0.000	600.000	75.712	100.000	100.000
	$n/2$	3	100	1144.000	1100.000	300.000	6200.000	597.185	900.000	1300.000
	n	3	100	2202.000	2100.000	900.000	3700.000	522.036	1900.000	2550.000
f_8	1	3	100	1479.000	300.000	0.000	20900.000	3070.824	200.000	1200.000
	$n/2$	3	100	975.000	500.000	0.000	11600.000	1770.429	300.000	800.000
	n	3	100	1534.000	700.000	100.000	19800.000	3183.741	500.000	1200.000
f_9	1	3	100	3086.000	500.000	0.000	90100.000	9852.221	200.000	1600.000
	$n/2$	3	100	790.000	700.000	0.000	2900.000	466.125	500.000	1000.000
	n	3	100	865.000	750.000	100.000	6400.000	735.997	550.000	900.000
f_{10}	1	3	100	1682.000	400.000	0.000	22400.000	3692.235	200.000	1100.000
	$n/2$	3	100	1015.000	700.000	0.000	9900.000	1305.418	500.000	1100.000
	n	3	100	905.000	800.000	0.000	6200.000	772.164	500.000	1100.000
f_{11}	1	3	100	1462.000	350.000	100.000	24900.000	3527.719	200.000	1050.000
	$n/2$	3	100	1126.000	800.000	0.000	17900.000	1997.150	500.000	1100.000
	n	3	100	1053.000	1000.000	0.000	2400.000	487.304	700.000	1300.000
f_{12}	1	3	100	979.000	500.000	0.000	9400.000	1450.691	200.000	1000.000
	$n/2$	3	100	908.000	900.000	0.000	2500.000	491.890	500.000	1200.000
	n	3	100	940.000	800.000	100.000	5200.000	640.549	500.000	1250.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A3: Numerical results for the test set 1 with the tree depth (d) equal to 5.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	5	100	91.000	100.000	0.000	900.000	95.447	100.000	100.000
	$n/2$	5	99	376.768	300.000	0.000	2100.000	242.785	300.000	400.000
	n	5	100	582.000	600.000	200.000	1300.000	189.832	500.000	700.000
f_2	1	5	100	67777.500	25025.000	50.000	648500.000	106830.875	1350.000	81375.000
	$n/2$	5	0	–	–	–	–	–	–	–
	n	5	100	–	–	–	–	–	–	–
f_3	1	5	100	8.000	0.000	0.000	100.000	22.156	0.000	0.000
	$n/2$	5	100	3.500	0.000	0.000	100.000	16.291	0.000	0.000
	n	5	100	2.500	0.000	0.000	50.000	10.952	0.000	0.000
f_4	1	5	100	21.000	0.000	0.000	100.000	40.936	0.000	0.000
	$n/2$	5	100	3232.000	3250.000	1500.000	5800.000	949.416	2500.000	3800.000
	n	5	100	5923.000	5750.000	2500.000	9800.000	1434.556	4950.000	6950.000
f_5	1	5	100	15590.000	8275.000	100.000	147950.000	26096.659	4600.000	15175.000
	$n/2$	5	98	170390.816	80175.000	400.000	1560250.000	250189.823	37800.000	178500.000
	n	5	100	149828.000	110900.000	14400.000	700000.000	127382.178	60925.000	184800.000
f_6	1	5	100	56622.000	1400.000	0.000	1773200.000	216023.201	400.000	11900.000
	$n/2$	5	97	200843.299	21400.000	400.000	1363400.000	320776.187	1600.000	261600.000
	n	5	97	184793.814	31000.000	1000.000	1660200.000	338268.067	3000.000	188950.000
f_7	1	5	100	86.000	100.000	0.000	900.000	98.494	0.000	100.000
	$n/2$	5	100	1116.000	1100.000	500.000	2600.000	287.701	950.000	1300.000
	n	5	100	2171.000	2200.000	1100.000	3500.000	516.651	1850.000	2550.000
f_8	1	5	100	199.000	100.000	0.000	1500.000	204.245	100.000	200.000
	$n/2$	5	100	296.000	300.000	0.000	1100.000	205.441	200.000	400.000
	n	5	100	294.000	250.000	0.000	1000.000	229.985	100.000	400.000
f_9	1	5	100	177.000	100.000	0.000	1400.000	189.019	100.000	200.000
	$n/2$	5	100	258.000	300.000	0.000	700.000	197.039	100.000	400.000
	n	5	100	233.000	200.000	0.000	800.000	224.308	0.000	400.000
f_{10}	1	5	100	232.000	200.000	0.000	2000.000	271.873	100.000	250.000
	$n/2$	5	100	273.000	200.000	0.000	1100.000	246.534	50.000	450.000
	n	5	100	193.000	200.000	0.000	700.000	188.698	0.000	300.000
f_{11}	1	5	100	176.000	100.000	0.000	1500.000	223.435	100.000	200.000
	$n/2$	5	100	268.000	300.000	0.000	1200.000	240.740	0.000	400.000
	n	5	100	246.000	200.000	0.000	1400.000	254.820	0.000	350.000
f_{12}	1	5	100	462.000	150.000	0.000	26300.000	2623.622	100.000	200.000
	$n/2$	5	100	248.000	200.000	0.000	1200.000	256.030	0.000	400.000
	n	5	100	185.000	200.000	0.000	800.000	194.040	0.000	300.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A4: Numerical results for the test set 1 with the tree depth (d) equal to 7.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	7	100	126.000	100.000	0.000	4600.000	458.064	0.000	100.000
	$n/2$	7	100	296.000	300.000	0.000	600.000	122.202	200.000	400.000
	n	7	100	531.000	500.000	100.000	1100.000	187.861	400.000	600.000
f_2	1	7	100	67115.000	21175.000	50.000	542100.000	115401.338	1475.000	62400.000
	$n/2$	7	0	—	—	—	—	—	—	—
	n	7	0	—	—	—	—	—	—	—
f_3	1	7	100	1.000	0.000	0.000	50.000	7.035	0.000	0.000
	$n/2$	7	100	0.500	0.000	0.000	50.000	5.000	0.000	0.000
	n	7	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_4	1	7	100	17.000	0.000	0.000	100.000	37.753	0.000	0.000
	$n/2$	7	100	3324.000	3300.000	1500.000	6400.000	980.427	2600.000	3900.000
	n	7	100	6411.000	6350.000	2500.000	11000.000	1564.240	5400.000	7650.000
f_5	1	7	100	38571.500	6400.000	100.000	829100.000	99642.140	1650.000	29225.000
	$n/2$	7	95	180195.263	69700.000	1100.000	1737300.000	304490.680	30337.500	190862.500
	n	7	100	164993.500	108700.000	8550.000	1493700.000	204234.686	49750.000	187050.000
f_6	1	7	99	24567.677	1600.000	0.000	529600.000	85436.979	600.000	4800.000
	$n/2$	7	98	206573.469	10000.000	400.000	1853600.000	376493.359	1600.000	262000.000
	n	7	98	265324.490	63100.000	800.000	1825400.000	395656.570	3400.000	367400.000
f_7	1	7	100	67.000	100.000	0.000	200.000	51.355	0.000	100.000
	$n/2$	7	100	1131.000	1100.000	400.000	2600.000	341.623	900.000	1300.000
	n	7	99	2185.859	2200.000	900.000	5500.000	674.311	1800.000	2500.000
f_8	1	7	100	83.000	100.000	0.000	400.000	86.521	0.000	100.000
	$n/2$	7	100	40.000	0.000	0.000	800.000	119.764	0.000	0.000
	n	7	100	28.000	0.000	0.000	300.000	72.586	0.000	0.000
f_9	1	7	100	91.000	100.000	0.000	400.000	93.306	0.000	100.000
	$n/2$	7	100	9.000	0.000	0.000	200.000	32.083	0.000	0.000
	n	7	100	3.000	0.000	0.000	200.000	22.270	0.000	0.000
f_{10}	1	7	100	97.000	100.000	0.000	500.000	104.886	0.000	100.000
	$n/2$	7	100	22.000	0.000	0.000	300.000	64.479	0.000	0.000
	n	7	100	2.000	0.000	0.000	100.000	14.071	0.000	0.000
f_{11}	1	7	100	69.000	100.000	0.000	400.000	78.746	0.000	100.000
	$n/2$	7	100	7.000	0.000	0.000	600.000	60.728	0.000	0.000
	n	7	100	2.000	0.000	0.000	100.000	14.071	0.000	0.000
f_{12}	1	7	100	89.000	100.000	0.000	500.000	104.345	0.000	100.000
	$n/2$	7	100	2.000	0.000	0.000	200.000	20.000	0.000	0.000
	n	7	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A5: Numerical results for the test set 1 with the tree depth (d) equal to 9.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	9	100	53.000	100.000	0.000	200.000	54.039	0.000	100.000
	$n/2$	9	100	281.000	300.000	100.000	600.000	135.360	200.000	400.000
	n	9	100	536.000	600.000	200.000	900.000	177.252	400.000	700.000
f_2	1	9	99	59526.263	8250.000	50.000	400600.000	96834.093	450.000	73237.500
	$n/2$	9	0	—	—	—	—	—	—	—
	n	9	0	—	—	—	—	—	—	—
f_3	1	9	100	0.500	0.000	0.000	50.000	5.000	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_4	1	9	100	20.000	0.000	0.000	100.000	40.202	0.000	0.000
	$n/2$	9	100	3497.000	3400.000	1400.000	9000.000	1156.750	3000.000	3950.000
	n	9	100	6123.000	5800.000	3000.000	10800.000	1601.190	4850.000	7500.000
f_5	1	9	100	46753.000	6475.000	200.000	778950.000	108406.904	2300.000	30950.000
	$n/2$	9	91	175979.121	64800.000	650.000	1770000.000	327628.160	32900.000	132525.000
	n	9	100	164769.500	107075.000	8550.000	1186600.000	189325.178	53275.000	198050.000
f_6	1	9	96	45658.333	3200.000	0.000	782400.000	132793.043	600.000	24100.000
	$n/2$	9	93	269539.785	53600.000	400.000	1931800.000	415599.070	5300.000	364050.000
	n	9	98	241077.551	26400.000	1200.000	1802600.000	375844.584	4800.000	319800.000
f_7	1	9	100	50.000	0.000	0.000	200.000	52.223	0.000	100.000
	$n/2$	9	100	1092.000	1100.000	400.000	1800.000	310.971	900.000	1300.000
	n	9	97	2169.072	2200.000	700.000	3800.000	622.221	1775.000	2500.000
f_8	1	9	100	26.000	0.000	0.000	400.000	57.945	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_9	1	9	100	10.000	0.000	0.000	200.000	33.333	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{10}	1	9	100	21.000	0.000	0.000	200.000	49.838	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{11}	1	9	100	28.000	0.000	0.000	500.000	75.318	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{12}	1	9	100	18.000	0.000	0.000	200.000	43.531	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

A2.2

Set 2

Table A6: Numerical results for the test set 2 with the tree depth (d) equal to 2.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	2	0	–	–	–	–	–	–	–
	$n/2$	2	4	1034875.000	1268050.000	39200.000	1564200.000	712590.071	519900.000	1549850.000
	n	2	100	22414.000	20600.000	2900.000	93600.000	14610.174	12950.000	28050.000
f_2	1	2	1	393850.000	393850.000	393850.000	393850.000	0.000	393850.000	393850.000
	$n/2$	2	0	–	–	–	–	–	–	–
	n	2	100	–	–	–	–	–	–	–
f_3	1	2	100	235.000	100.000	0.000	3100.000	388.698	50.000	300.000
	$n/2$	2	100	155.500	100.000	0.000	1000.000	200.617	50.000	200.000
	n	2	100	138.500	100.000	0.000	750.000	153.882	0.000	200.000
f_4	1	2	0	–	–	–	–	–	–	–
	$n/2$	2	0	–	–	–	–	–	–	–
	n	2	100	34230.000	33000.000	20000.000	59000.000	8001.395	28600.000	39350.000
f_5	1	2	13	62600.000	57550.000	14550.000	132600.000	39909.998	28775.000	90562.500
	$n/2$	2	14	603325.000	476625.000	55300.000	1294500.000	357100.665	340450.000	917100.000
	n	2	99	463122.727	381950.000	73500.000	1486600.000	341744.782	191812.500	593530.000
f_6	1	2	0	–	–	–	–	–	–	–
	$n/2$	2	0	–	–	–	–	–	–	–
	n	2	0	–	–	–	–	–	–	–
f_7	1	2	0	–	–	–	–	–	–	–
	$n/2$	2	0	–	–	–	–	–	–	–
	n	2	100	50800.000	50100.000	15400.000	115800.000	20438.130	36300.000	63950.000
f_8	1	2	100	11515.000	7200.000	200.000	64800.000	13328.252	3500.000	13900.000
	$n/2$	2	100	12178.000	8400.000	200.000	67500.000	13254.361	3400.000	14000.000
	n	2	100	9857.000	6600.000	200.000	36300.000	9251.287	2350.000	15400.000
f_9	1	2	100	14579.000	10750.000	300.000	53500.000	13767.634	4050.000	20700.000
	$n/2$	2	100	11853.000	7050.000	200.000	57000.000	12697.049	2750.000	16300.000
	n	2	100	12436.000	5650.000	300.000	108100.000	15392.347	1900.000	20200.000
f_{10}	1	2	100	11188.000	8200.000	0.000	67800.000	11555.184	3600.000	13100.000
	$n/2$	2	100	13706.000	7700.000	400.000	78300.000	16614.894	2650.000	16600.000
	n	2	100	13628.000	8000.000	600.000	100800.000	16676.537	2500.000	17300.000
f_{11}	1	2	100	11408.000	6650.000	100.000	60100.000	12206.346	3200.000	14950.000
	$n/2$	2	100	11394.000	6950.000	400.000	67900.000	12729.746	2650.000	16600.000
	n	2	100	9127.000	3550.000	400.000	56300.000	11787.134	1700.000	12100.000
f_{12}	1	2	100	12143.000	8600.000	400.000	51400.000	10530.123	4600.000	16750.000
	$n/2$	2	100	10650.000	6450.000	400.000	65200.000	12735.955	1900.000	13600.000
	n	2	100	7598.000	3800.000	500.000	67100.000	10190.270	1700.000	10550.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A7: Numerical results for the test set 2 with the tree depth (d) equal to 3.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	3	2	1837850.000	1837850.000	1774900.000	1900800.000	89024.744	1774900.000	1900800.000
	$n/2$	3	46	411421.739	181800.000	1500.000	1963700.000	531838.156	48900.000	580400.000
	n	3	100	24076.000	20400.000	1700.000	86900.000	17037.209	10450.000	32350.000
f_2	1	3	9	711283.333	378350.000	90550.000	1901800.000	710437.407	258575.000	1245887.500
	$n/2$	3	0	—	—	—	—	—	—	—
	n	3	0	—	—	—	—	—	—	—
f_3	1	3	100	34.000	0.000	0.000	350.000	58.552	0.000	50.000
	$n/2$	3	100	38.000	0.000	0.000	300.000	59.933	0.000	50.000
	n	3	100	48.000	0.000	0.000	450.000	72.097	0.000	50.000
f_4	1	3	0	—	—	—	—	—	—	—
	$n/2$	3	5	1360160.000	1642400.000	522600.000	1901300.000	623573.935	788175.000	1868675.000
	n	3	100	30514.000	29600.000	18400.000	50600.000	6646.804	25100.000	35000.000
f_5	1	3	44	239328.409	33125.000	1950.000	1614950.000	430928.923	14900.000	144175.000
	$n/2$	3	34	524875.000	361075.000	16150.000	1757950.000	483876.068	176700.000	834950.000
	n	3	100	183382.000	158850.000	10250.000	870850.000	141891.204	88900.000	228575.000
f_6	1	3	0	—	—	—	—	—	—	—
	$n/2$	3	0	—	—	—	—	—	—	—
	n	3	0	—	—	—	—	—	—	—
f_7	1	3	0	—	—	—	—	—	—	—
	$n/2$	3	1	1314600.000	1314600.000	1314600.000	1314600.000	0.000	1314600.000	1314600.000
	n	3	100	48509.000	42450.000	12300.000	165000.000	25539.896	32750.000	58800.000
f_8	1	3	100	1005.000	300.000	0.000	26000.000	2706.086	200.000	1100.000
	$n/2$	3	100	1309.000	600.000	0.000	17700.000	2548.642	400.000	900.000
	n	3	100	1242.000	700.000	0.000	24700.000	2700.175	500.000	900.000
f_9	1	3	100	1485.000	450.000	0.000	24900.000	3174.150	200.000	1200.000
	$n/2$	3	100	1286.000	700.000	0.000	33900.000	3531.475	500.000	1000.000
	n	3	100	889.000	700.000	0.000	4400.000	588.423	600.000	1150.000
f_{10}	1	3	100	2109.000	400.000	0.000	52400.000	6269.537	200.000	1100.000
	$n/2$	3	100	1169.000	800.000	0.000	31400.000	3104.988	600.000	1000.000
	n	3	100	1006.000	800.000	0.000	4300.000	639.258	600.000	1250.000
f_{11}	1	3	100	1539.000	550.000	0.000	28800.000	4101.167	200.000	1100.000
	$n/2$	3	100	1021.000	800.000	0.000	10100.000	1093.442	500.000	1300.000
	n	3	100	976.000	1000.000	0.000	3000.000	518.763	600.000	1250.000
f_{12}	1	3	100	1548.000	500.000	100.000	45600.000	5004.815	200.000	1000.000
	$n/2$	3	100	943.000	900.000	0.000	3600.000	544.440	600.000	1200.000
	n	3	100	1017.000	1000.000	200.000	4100.000	544.774	700.000	1300.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A8: Numerical results for the test set 2 with the tree depth (d) equal to 5.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	5	32	630781.250	434500.000	13200.000	1770700.000	581802.917	78900.000	1111750.000
	$n/2$	5	75	281428.000	132900.000	7100.000	1854900.000	396877.793	43500.000	341150.000
	n	5	100	19104.000	13850.000	1500.000	97100.000	17758.031	6950.000	24500.000
f_2	1	5	28	514430.357	502700.000	10700.000	1756150.000	402057.528	154300.000	704925.000
	$n/2$	5	0	—	—	—	—	—	—	—
	n	5	0	—	—	—	—	—	—	—
f_3	1	5	100	5.000	0.000	0.000	100.000	16.667	0.000	0.000
	$n/2$	5	100	2.000	0.000	0.000	50.000	9.847	0.000	0.000
	n	5	100	3.000	0.000	0.000	50.000	11.934	0.000	0.000
f_4	1	5	0	—	—	—	—	—	—	—
	$n/2$	5	70	712271.429	628800.000	93700.000	1844600.000	472993.180	284300.000	1005800.000
	n	5	100	28782.000	26450.000	17000.000	69400.000	8518.612	23200.000	33250.000
f_5	1	5	96	211866.667	21850.000	200.000	1924050.000	387141.201	7175.000	233000.000
	$n/2$	5	73	293853.425	116400.000	7750.000	1771500.000	413556.237	68850.000	301287.500
	n	5	100	132101.000	89575.000	6100.000	971100.000	126032.633	58100.000	173675.000
f_6	1	5	0	—	—	—	—	—	—	—
	$n/2$	5	0	—	—	—	—	—	—	—
	n	5	0	—	—	—	—	—	—	—
f_7	1	5	0	—	—	—	—	—	—	—
	$n/2$	5	50	713536.000	553350.000	61600.000	1958000.000	579299.195	180500.000	1053100.000
	n	5	99	36152.525	31900.000	12700.000	109900.000	19750.037	21450.000	44550.000
f_8	1	5	100	181.000	100.000	0.000	1500.000	207.289	100.000	200.000
	$n/2$	5	100	246.000	200.000	0.000	700.000	185.003	100.000	400.000
	n	5	100	302.000	300.000	0.000	1600.000	280.685	100.000	400.000
f_9	1	5	100	196.000	200.000	0.000	1300.000	183.633	100.000	200.000
	$n/2$	5	100	251.000	200.000	0.000	1300.000	232.898	0.000	400.000
	n	5	100	237.000	200.000	0.000	800.000	222.772	0.000	400.000
f_{10}	1	5	100	242.000	100.000	0.000	2800.000	414.675	100.000	200.000
	$n/2$	5	100	251.000	200.000	0.000	800.000	232.029	0.000	400.000
	n	5	100	270.000	300.000	0.000	900.000	227.192	0.000	400.000
f_{11}	1	5	100	189.000	100.000	0.000	1400.000	200.955	100.000	200.000
	$n/2$	5	100	223.000	200.000	0.000	1300.000	251.000	0.000	400.000
	n	5	100	198.000	200.000	0.000	700.000	188.015	0.000	300.000
f_{12}	1	5	100	189.000	200.000	0.000	700.000	150.350	100.000	200.000
	$n/2$	5	100	250.000	200.000	0.000	1000.000	252.862	0.000	400.000
	n	5	100	236.000	200.000	0.000	1100.000	231.604	0.000	400.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A9: Numerical results for the test set 2 with the tree depth (d) equal to 7.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	7	76	450513.158	255050.000	4500.000	1966400.000	499497.757	67350.000	655400.000
	$n/2$	7	90	169258.889	47550.000	2700.000	1329200.000	268662.462	14900.000	211600.000
	n	7	100	16176.000	7500.000	900.000	108900.000	19624.763	3450.000	21850.000
f_2	1	7	80	470735.000	348950.000	5650.000	1806250.000	447305.257	84375.000	763300.000
	$n/2$	7	0	—	—	—	—	—	—	—
	n	7	0	—	—	—	—	—	—	—
f_3	1	7	100	1.000	0.000	0.000	50.000	7.035	0.000	0.000
	$n/2$	7	100	0.500	0.000	0.000	50.000	5.000	0.000	0.000
	n	7	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_4	1	7	0	—	—	—	—	—	—	—
	$n/2$	7	91	438524.176	247100.000	39800.000	1959900.000	446027.468	129200.000	628375.000
	n	7	100	28681.000	27050.000	17800.000	72400.000	8260.991	23500.000	31850.000
f_5	1	7	99	79162.121	10800.000	350.000	1748000.000	206296.440	4112.500	48575.000
	$n/2$	7	80	302208.125	120425.000	1600.000	1982400.000	384506.111	34525.000	415825.000
	n	7	100	84274.500	52025.000	5850.000	472250.000	82952.680	35300.000	100425.000
f_6	1	7	0	—	—	—	—	—	—	—
	$n/2$	7	0	—	—	—	—	—	—	—
	n	7	0	—	—	—	—	—	—	—
f_7	1	7	1	1412800.000	1412800.000	1412800.000	1412800.000	0.000	1412800.000	1412800.000
	$n/2$	7	78	435564.103	246250.000	25200.000	1952000.000	446131.809	115900.000	581900.000
	n	7	95	33951.579	30200.000	9000.000	118900.000	19968.086	19175.000	43075.000
f_8	1	7	100	88.000	100.000	0.000	1400.000	170.726	0.000	100.000
	$n/2$	7	100	59.000	0.000	0.000	800.000	123.987	0.000	100.000
	n	7	100	29.000	0.000	0.000	400.000	78.232	0.000	0.000
f_9	1	7	100	74.000	100.000	0.000	600.000	99.107	0.000	100.000
	$n/2$	7	100	8.000	0.000	0.000	200.000	30.748	0.000	0.000
	n	7	100	4.000	0.000	0.000	300.000	31.527	0.000	0.000
f_{10}	1	7	100	109.000	100.000	0.000	1200.000	156.409	0.000	100.000
	$n/2$	7	100	5.000	0.000	0.000	200.000	26.112	0.000	0.000
	n	7	100	2.000	0.000	0.000	200.000	20.000	0.000	0.000
f_{11}	1	7	100	71.000	100.000	0.000	400.000	80.773	0.000	100.000
	$n/2$	7	100	2.000	0.000	0.000	100.000	14.071	0.000	0.000
	n	7	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{12}	1	7	100	79.000	100.000	0.000	400.000	83.236	0.000	100.000
	$n/2$	7	100	2.000	0.000	0.000	200.000	20.000	0.000	0.000
	n	7	100	1.000	0.000	0.000	100.000	10.000	0.000	0.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Table A10: Numerical results for the test set 2 with the tree depth (d) equal to 9.

Benchmark Function	k	d	HITS	MNE	Median	Min	Max	Std	1st Quartile	3rd Quartile
f_1	1	9	94	209245.745	61850.000	1800.000	1567700.000	311962.734	17100.000	255300.000
	$n/2$	9	98	105754.082	24150.000	1900.000	961600.000	194180.495	9100.000	77600.000
	n	9	100	16838.000	10800.000	1500.000	115800.000	17789.554	5250.000	22200.000
f_2	1	9	86	191109.884	65325.000	2750.000	1523250.000	298018.973	15800.000	244550.000
	$n/2$	9	0	—	—	—	—	—	—	—
	n	9	0	—	—	—	—	—	—	—
f_3	1	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_4	1	9	0	—	—	—	—	—	—	—
	$n/2$	9	88	373156.818	192750.000	43800.000	1989200.000	453078.330	97250.000	459200.000
	n	9	100	28248.000	26650.000	17900.000	51100.000	7537.562	23000.000	32550.000
f_5	1	9	99	139900.505	10700.000	200.000	1871900.000	315419.062	4837.500	58387.500
	$n/2$	9	73	394686.301	132100.000	4650.000	1973200.000	529829.629	46150.000	564137.500
	n	9	100	71622.500	46775.000	5100.000	504650.000	69962.603	29925.000	94550.000
f_6	1	9	0	—	—	—	—	—	—	—
	$n/2$	9	0	—	—	—	—	—	—	—
	n	9	0	—	—	—	—	—	—	—
f_7	1	9	5	982600.000	964800.000	67800.000	1763500.000	616183.078	680625.000	1364875.000
	$n/2$	9	90	337923.333	131950.000	16300.000	1997000.000	461742.392	63000.000	415300.000
	n	9	93	29850.538	25300.000	8800.000	105100.000	18115.983	17350.000	35800.000
f_8	1	9	100	18.000	0.000	0.000	200.000	43.531	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_9	1	9	100	23.000	0.000	0.000	300.000	52.905	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{10}	1	9	100	20.000	0.000	0.000	500.000	63.564	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{11}	1	9	100	18.000	0.000	0.000	300.000	45.793	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{12}	1	9	100	18.000	0.000	0.000	200.000	47.948	0.000	0.000
	$n/2$	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	n	9	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; HITS - Number of hits on the function global minimum; MNE - Mean number of evaluations; Median - Median number of evaluations; Min - Minimum number of evaluations; Max - Maximum number of evaluations; Std - Standard deviation of evaluations; 1st Quartile - First quartile of evaluations; 3rd Quartile - Third quartile of evaluations.

Tables A11 and A12 show the results of Aligned Friedman’s and Iman-Davenport’s tests, along with the Holm’s procedure based on MG-PMA performance in the test set 2 for all values of k and d considered (Tables A6-A10).

Table A11: Average rankings of Aligned Friedman’s and Iman-Davenport’s tests for comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 2.

i	Algorithm	Rank
14	MG-PMA with $k = n/2$ and $d = 2$	130.000
13	MG-PMA with $k = 1$ and $d = 2$	122.400
12	MG-PMA with $k = 1$ and $d = 3$	107.700
11	MG-PMA with $k = n/2$ and $d = 3$	102.400
10	MG-PMA with $k = 1$ and $d = 5$	81.200
9	MG-PMA with $k = n$ and $d = 2$	80.9
8	MG-PMA with $k = 1$ and $d = 7$	77.650
7	MG-PMA with $k = n$ and $d = 3$	65.200
6	MG-PMA with $k = 1$ and $d = 9$	62.300
5	MG-PMA with $k = n/2$ and $d = 5$	57.100
4	MG-PMA with $k = n$ and $d = 5$	54.300
3	MG-PMA with $k = n/2$ and $d = 7$	50.000
2	MG-PMA with $k = n/2$ and $d = 9$	48.550
1	MG-PMA with $k = n$ and $d = 7$	46.950
0	MG-PMA with $k = n$ and $d = 9$	45.950

Table A12: Holm’s procedure for pairwise comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 2. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n$ and $d = 9$.

Test	p -value				
Aligned Friedman	0.839				
Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?	
MG-PMA with $k = n/2$ and $d = 2$	4.326	0.000	0.004	Yes	
MG-PMA with $k = 1$ and $d = 2$	3.935	0.000	0.004	Yes	
MG-PMA with $k = 1$ and $d = 3$	3.178	0.002	0.004	Yes	
MG-PMA with $k = n/2$ and $d = 3$	2.905	0.004	0.005	Yes	
MG-PMA with $k = 1$ and $d = 5$	1.814	0.070	0.005	No	
MG-PMA with $k = n$ and $d = 2$	1.794	0.073	0.006	No	
MG-PMA with $k = 1$ and $d = 7$	1.632	0.103	0.006	No	
MG-PMA with $k = n$ and $d = 3$	0.991	0.322	0.007	No	
MG-PMA with $k = 1$ and $d = 9$	0.842	0.400	0.008	No	
MG-PMA with $k = n/2$ and $d = 5$	0.574	0.566	0.010	No	
MG-PMA with $k = n$ and $d = 5$	0.430	0.667	0.013	No	
MG-PMA with $k = n/2$ and $d = 7$	0.208	0.835	0.017	No	
MG-PMA with $k = n/2$ and $d = 9$	0.134	0.894	0.025	No	
MG-PMA with $k = n$ and $d = 7$	0.052	0.959	0.050	No	

A2.3

Set 3

Table A13: Numerical results for the test set 3 with the tree depth (d) equal to 2.

Benchmark Function	k	d	MVBF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	2	1.523e+08	1.698e+08	1.131e+08	1.698e+08	2.265e+07	1.131e+08	1.131e+08
	$n/2$	2	2.166e+07	1.755e+07	1.071e+07	6.280e+07	1.205e+07	1.071e+07	1.071e+07
	n	2	9.614e+06	5.597e+06	1.035e+06	5.590e+07	1.180e+07	1.035e+06	1.035e+06
c_2	1	2	1.643e+10	1.836e+10	9.113e+09	2.204e+10	6.145e+09	9.113e+09	9.113e+09
	$n/2$	2	5.166e+09	5.007e+09	4.829e+09	6.659e+09	3.928e+08	4.829e+09	4.829e+09
	n	2	1.410e+06	2.424e+05	3.977e+04	5.520e+07	7.688e+06	3.977e+04	3.977e+04
c_3	1	2	3.206e+02	3.206e+02	3.203e+02	3.209e+02	1.477e-01	3.203e+02	3.203e+02
	$n/2$	2	3.201e+02	3.200e+02	3.200e+02	3.205e+02	7.588e-02	3.200e+02	3.200e+02
	n	2	3.200e+02	3.200e+02	3.200e+02	3.200e+02	1.646e-03	3.200e+02	3.200e+02
c_4	1	2	4.940e+02	4.934e+02	4.632e+02	5.108e+02	9.724e+00	4.632e+02	4.632e+02
	$n/2$	2	4.535e+02	4.510e+02	4.304e+02	4.716e+02	8.333e+00	4.304e+02	4.304e+02
	n	2	4.536e+02	4.550e+02	4.244e+02	4.637e+02	1.071e+01	4.244e+02	4.244e+02
c_5	1	2	2.187e+03	2.157e+03	2.098e+03	2.455e+03	6.971e+01	2.098e+03	2.098e+03
	$n/2$	2	1.365e+03	1.376e+03	9.595e+02	2.009e+03	2.337e+02	9.595e+02	9.595e+02
	n	2	1.116e+03	1.063e+03	8.445e+02	1.732e+03	1.904e+02	8.445e+02	8.445e+02
c_6	1	2	7.796e+07	1.251e+08	5.004e+06	1.251e+08	5.090e+07	5.004e+06	5.004e+06
	$n/2$	2	1.278e+06	1.506e+06	4.826e+04	2.688e+06	6.652e+05	4.826e+04	4.826e+04
	n	2	9.492e+05	6.698e+04	1.679e+03	4.299e+07	6.006e+06	1.679e+03	1.679e+03
c_7	1	2	1.024e+03	9.144e+02	8.536e+02	1.268e+03	1.669e+02	8.536e+02	8.536e+02
	$n/2$	2	7.072e+02	7.095e+02	7.020e+02	7.119e+02	3.860e+00	7.020e+02	7.020e+02
	n	2	7.011e+02	7.011e+02	7.003e+02	7.026e+02	4.414e-01	7.003e+02	7.003e+02
c_8	1	2	2.825e+06	1.957e+06	5.143e+03	9.757e+06	2.283e+06	5.143e+03	5.143e+03
	$n/2$	2	7.375e+05	2.789e+05	2.080e+03	8.545e+06	1.620e+06	2.080e+03	2.080e+03
	n	2	4.950e+05	2.079e+05	1.719e+03	3.088e+06	7.079e+05	1.719e+03	1.719e+03
c_9	1	2	1.121e+03	1.112e+03	1.054e+03	1.254e+03	6.594e+01	1.054e+03	1.054e+03
	$n/2$	2	1.013e+03	1.011e+03	1.005e+03	1.039e+03	9.473e+00	1.005e+03	1.005e+03
	n	2	1.001e+03	1.001e+03	1.000e+03	1.003e+03	5.119e-01	1.000e+03	1.000e+03
c_{10}	1	2	7.258e+05	6.255e+05	1.202e+05	3.818e+06	7.154e+05	1.202e+05	1.202e+05
	$n/2$	2	5.540e+04	4.254e+04	4.188e+03	2.674e+05	5.538e+04	4.188e+03	4.188e+03
	n	2	2.740e+04	1.354e+04	1.381e+03	2.134e+05	3.823e+04	1.381e+03	1.381e+03
c_{11}	1	2	1.661e+03	1.657e+03	1.638e+03	1.768e+03	2.105e+01	1.638e+03	1.638e+03
	$n/2$	2	1.408e+03	1.408e+03	1.402e+03	1.418e+03	3.375e+00	1.402e+03	1.402e+03
	n	2	1.386e+03	1.403e+03	1.115e+03	1.406e+03	6.687e+01	1.115e+03	1.115e+03
c_{12}	1	2	1.388e+03	1.400e+03	1.367e+03	1.400e+03	1.321e+01	1.367e+03	1.367e+03
	$n/2$	2	1.316e+03	1.311e+03	1.304e+03	1.400e+03	2.151e+01	1.304e+03	1.304e+03
	n	2	1.323e+03	1.308e+03	1.303e+03	1.400e+03	3.383e+01	1.303e+03	1.303e+03
c_{13}	1	2	1.300e+03	1.300e+03	1.300e+03	1.300e+03	6.436e-05	1.300e+03	1.300e+03
	$n/2$	2	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.337e-04	1.300e+03	1.300e+03
	n	2	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.429e-04	1.300e+03	1.300e+03
c_{14}	1	2	8.175e+03	6.117e+03	6.109e+03	1.937e+04	4.330e+03	6.109e+03	6.109e+03
	$n/2$	2	8.741e+03	8.707e+03	2.736e+03	1.473e+04	2.508e+03	2.736e+03	2.736e+03
	n	2	3.558e+03	1.555e+03	1.518e+03	1.263e+04	3.246e+03	1.518e+03	1.518e+03
c_{15}	1	2	2.698e+03	2.695e+03	2.553e+03	3.513e+03	2.177e+02	2.553e+03	2.553e+03
	$n/2$	2	2.265e+03	2.256e+03	1.816e+03	2.419e+03	9.600e+01	1.816e+03	1.816e+03
	n	2	1.600e+03	1.600e+03	1.601e+03	1.601e+03	1.653e-01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A14: Numerical results for the test set 3 with the tree depth (d) equal to 3.

Benchmark Function	k	d	MVBF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	3	1.083e+08	1.075e+08	3.063e+07	1.698e+08	2.854e+07	3.063e+07	3.063e+07
	$n/2$	3	1.603e+07	1.489e+07	4.936e+06	3.046e+07	6.017e+06	4.936e+06	4.936e+06
	n	3	6.001e+06	4.034e+06	9.295e+05	5.039e+07	7.839e+06	9.295e+05	9.295e+05
c_2	1	3	8.203e+09	8.152e+09	1.461e+09	1.760e+10	3.305e+09	1.461e+09	1.461e+09
	$n/2$	3	1.569e+09	1.504e+08	2.379e+06	6.121e+09	2.174e+09	2.379e+06	2.379e+06
	n	3	1.222e+05	7.122e+04	1.173e+04	6.470e+05	1.330e+05	1.173e+04	1.173e+04
c_3	1	3	3.205e+02	3.204e+02	3.203e+02	3.208e+02	1.368e-01	3.203e+02	3.203e+02
	$n/2$	3	3.200e+02	3.200e+02	3.200e+02	3.203e+02	4.723e-02	3.200e+02	3.200e+02
	n	3	3.200e+02	3.200e+02	3.200e+02	3.200e+02	1.446e-03	3.200e+02	3.200e+02
c_4	1	3	4.758e+02	4.736e+02	4.534e+02	5.085e+02	1.186e+01	4.534e+02	4.534e+02
	$n/2$	3	4.462e+02	4.473e+02	4.186e+02	4.647e+02	1.199e+01	4.186e+02	4.186e+02
	n	3	4.439e+02	4.438e+02	4.249e+02	4.637e+02	9.687e+00	4.249e+02	4.249e+02
c_5	1	3	2.010e+03	2.028e+03	1.558e+03	2.464e+03	2.214e+02	1.558e+03	1.558e+03
	$n/2$	3	1.247e+03	1.189e+03	8.817e+02	2.125e+03	2.388e+02	8.817e+02	8.817e+02
	n	3	1.040e+03	9.820e+02	8.433e+02	1.347e+03	1.483e+02	8.433e+02	8.433e+02
c_6	1	3	1.752e+07	7.310e+06	8.145e+04	8.713e+07	2.271e+07	8.145e+04	8.145e+04
	$n/2$	3	1.019e+06	8.800e+05	1.106e+04	2.758e+06	7.785e+05	1.106e+04	1.106e+04
	n	3	1.138e+05	5.412e+04	1.349e+03	7.756e+05	1.539e+05	1.349e+03	1.349e+03
c_7	1	3	7.706e+02	7.283e+02	7.066e+02	9.280e+02	7.171e+01	7.066e+02	7.066e+02
	$n/2$	3	7.033e+02	7.030e+02	7.018e+02	7.110e+02	1.643e+00	7.018e+02	7.018e+02
	n	3	7.011e+02	7.011e+02	7.002e+02	7.026e+02	5.005e-01	7.002e+02	7.002e+02
c_8	1	3	1.447e+06	7.564e+05	2.701e+03	4.481e+06	1.615e+06	2.701e+03	2.701e+03
	$n/2$	3	7.046e+05	1.878e+05	1.207e+03	5.198e+06	1.318e+06	1.207e+03	1.207e+03
	n	3	3.385e+05	1.496e+05	1.407e+03	1.891e+06	4.623e+05	1.407e+03	1.407e+03
c_9	1	3	1.056e+03	1.054e+03	1.023e+03	1.134e+03	2.288e+01	1.023e+03	1.023e+03
	$n/2$	3	1.003e+03	1.003e+03	1.001e+03	1.009e+03	2.522e+00	1.001e+03	1.001e+03
	n	3	1.001e+03	1.001e+03	1.000e+03	1.002e+03	3.249e-01	1.000e+03	1.000e+03
c_{10}	1	3	5.215e+05	4.473e+05	2.742e+04	8.006e+06	1.095e+06	2.742e+04	2.742e+04
	$n/2$	3	3.228e+04	2.143e+04	2.776e+03	2.698e+05	4.106e+04	2.776e+03	2.776e+03
	n	3	1.386e+04	6.445e+03	1.874e+03	1.435e+05	2.611e+04	1.874e+03	1.874e+03
c_{11}	1	3	1.507e+03	1.462e+03	1.405e+03	1.664e+03	9.959e+01	1.405e+03	1.405e+03
	$n/2$	3	1.405e+03	1.404e+03	1.402e+03	1.412e+03	2.378e+00	1.402e+03	1.402e+03
	n	3	1.403e+03	1.403e+03	1.401e+03	1.404e+03	6.813e-01	1.401e+03	1.401e+03
c_{12}	1	3	1.363e+03	1.360e+03	1.317e+03	1.400e+03	2.185e+01	1.317e+03	1.317e+03
	$n/2$	3	1.319e+03	1.310e+03	1.304e+03	1.400e+03	2.732e+01	1.304e+03	1.304e+03
	n	3	1.338e+03	1.309e+03	1.303e+03	1.400e+03	4.432e+01	1.303e+03	1.303e+03
c_{13}	1	3	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.133e-04	1.300e+03	1.300e+03
	$n/2$	3	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.058e-04	1.300e+03	1.300e+03
	n	3	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.291e-04	1.300e+03	1.300e+03
c_{14}	1	3	6.244e+03	6.061e+03	3.089e+03	1.583e+04	2.657e+03	3.089e+03	3.089e+03
	$n/2$	3	6.703e+03	7.614e+03	2.155e+03	1.504e+04	3.484e+03	2.155e+03	2.155e+03
	n	3	4.205e+03	1.535e+03	1.506e+03	1.312e+04	3.998e+03	1.506e+03	1.506e+03
c_{15}	1	3	2.332e+03	2.498e+03	1.691e+03	2.804e+03	3.430e+02	1.691e+03	1.691e+03
	$n/2$	3	1.884e+03	1.651e+03	1.600e+03	2.255e+03	2.997e+02	1.600e+03	1.600e+03
	n	3	1.600e+03	1.600e+03	1.600e+03	1.600e+03	1.117e-01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A15: Numerical results for the test set 3 with the tree depth (d) equal to 5.

Benchmark Function	k	d	MVBF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	5	3.917e+07	3.059e+07	4.292e+06	1.111e+08	2.753e+07	4.292e+06	4.292e+06
	$n/2$	5	7.554e+06	6.979e+06	2.678e+06	1.831e+07	3.562e+06	2.678e+06	2.678e+06
	n	5	1.811e+06	1.711e+06	7.500e+05	6.343e+06	9.447e+05	7.500e+05	7.500e+05
c_2	1	5	2.558e+09	1.362e+09	4.856e+07	1.114e+10	2.677e+09	4.856e+07	4.856e+07
	$n/2$	5	4.318e+06	1.415e+06	7.198e+03	3.271e+07	6.936e+06	7.198e+03	7.198e+03
	n	5	2.043e+04	1.015e+04	1.266e+03	1.181e+05	2.700e+04	1.266e+03	1.266e+03
c_3	1	5	3.204e+02	3.203e+02	3.201e+02	3.207e+02	1.401e-01	3.201e+02	3.201e+02
	$n/2$	5	3.200e+02	3.200e+02	3.200e+02	3.202e+02	4.437e-02	3.200e+02	3.200e+02
	n	5	3.200e+02	3.200e+02	3.200e+02	3.200e+02	2.378e-03	3.200e+02	3.200e+02
c_4	1	5	4.629e+02	4.627e+02	4.420e+02	4.926e+02	1.145e+01	4.420e+02	4.420e+02
	$n/2$	5	4.372e+02	4.369e+02	4.167e+02	4.638e+02	1.219e+01	4.167e+02	4.167e+02
	n	5	4.383e+02	4.378e+02	4.209e+02	4.597e+02	9.003e+00	4.209e+02	4.209e+02
c_5	1	5	1.804e+03	1.816e+03	1.291e+03	2.431e+03	2.837e+02	1.291e+03	1.291e+03
	$n/2$	5	1.089e+03	1.098e+03	6.375e+02	1.592e+03	1.546e+02	6.375e+02	6.375e+02
	n	5	1.011e+03	9.651e+02	8.432e+02	1.305e+03	1.284e+02	8.432e+02	8.432e+02
c_6	1	5	1.871e+06	2.376e+05	1.291e+04	5.441e+07	7.805e+06	1.291e+04	1.291e+04
	$n/2$	5	9.038e+05	7.928e+05	1.799e+04	2.557e+06	6.818e+05	1.799e+04	1.799e+04
	n	5	5.730e+04	3.130e+04	1.669e+03	2.756e+05	7.201e+04	1.669e+03	1.669e+03
c_7	1	5	7.142e+02	7.117e+02	7.040e+02	7.393e+02	8.075e+00	7.040e+02	7.040e+02
	$n/2$	5	7.025e+02	7.023e+02	7.012e+02	7.044e+02	7.352e-01	7.012e+02	7.012e+02
	n	5	7.009e+02	7.010e+02	7.003e+02	7.025e+02	4.022e-01	7.003e+02	7.003e+02
c_8	1	5	1.838e+05	8.154e+03	1.732e+03	3.894e+06	6.372e+05	1.732e+03	1.732e+03
	$n/2$	5	1.366e+05	5.708e+04	1.786e+03	9.096e+05	1.879e+05	1.786e+03	1.786e+03
	n	5	2.247e+05	7.191e+04	1.839e+03	1.320e+06	3.122e+05	1.839e+03	1.839e+03
c_9	1	5	1.020e+03	1.017e+03	1.000e+03	1.068e+03	1.600e+01	1.000e+03	1.000e+03
	$n/2$	5	1.001e+03	1.001e+03	1.000e+03	1.003e+03	5.578e-01	1.000e+03	1.000e+03
	n	5	1.001e+03	1.001e+03	1.000e+03	1.002e+03	3.608e-01	1.000e+03	1.000e+03
c_{10}	1	5	1.254e+05	9.264e+04	3.248e+03	6.258e+05	1.244e+05	3.248e+03	3.248e+03
	$n/2$	5	1.133e+04	8.354e+03	1.477e+03	5.332e+04	1.141e+04	1.477e+03	1.477e+03
	n	5	5.028e+03	3.029e+03	1.410e+03	3.675e+04	6.507e+03	1.410e+03	1.410e+03
c_{11}	1	5	1.428e+03	1.412e+03	1.403e+03	1.639e+03	5.295e+01	1.403e+03	1.403e+03
	$n/2$	5	1.402e+03	1.402e+03	1.401e+03	1.405e+03	8.876e-01	1.401e+03	1.401e+03
	n	5	1.402e+03	1.402e+03	1.401e+03	1.406e+03	8.092e-01	1.401e+03	1.401e+03
c_{12}	1	5	1.333e+03	1.328e+03	1.307e+03	1.400e+03	2.389e+01	1.307e+03	1.307e+03
	$n/2$	5	1.317e+03	1.309e+03	1.302e+03	1.400e+03	2.770e+01	1.302e+03	1.302e+03
	n	5	1.333e+03	1.310e+03	1.302e+03	1.400e+03	4.184e+01	1.302e+03	1.302e+03
c_{13}	1	5	1.300e+03	1.300e+03	1.300e+03	1.300e+03	7.098e-05	1.300e+03	1.300e+03
	$n/2$	5	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.883e-04	1.300e+03	1.300e+03
	n	5	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.305e-04	1.300e+03	1.300e+03
c_{14}	1	5	3.513e+03	3.241e+03	1.966e+03	9.733e+03	1.216e+03	1.966e+03	1.966e+03
	$n/2$	5	4.678e+03	2.212e+03	1.532e+03	1.309e+04	3.364e+03	1.532e+03	1.532e+03
	n	5	3.538e+03	1.510e+03	1.500e+03	1.263e+04	3.062e+03	1.500e+03	1.500e+03
c_{15}	1	5	1.721e+03	1.660e+03	1.616e+03	2.300e+03	1.714e+02	1.616e+03	1.616e+03
	$n/2$	5	1.613e+03	1.601e+03	1.600e+03	2.206e+03	8.467e+01	1.600e+03	1.600e+03
	n	5	1.600e+03	1.600e+03	1.600e+03	1.600e+03	5.365e-02	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A16: Numerical results for the test set 3 with the tree depth (d) equal to 7.

Benchmark Function	k	d	MVBF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	7	1.424e+07	6.844e+06	6.794e+05	8.270e+07	1.945e+07	6.794e+05	6.794e+05
	$n/2$	7	2.903e+06	2.602e+06	5.634e+05	8.051e+06	1.657e+06	5.634e+05	5.634e+05
	n	7	1.298e+06	1.054e+06	4.880e+05	3.248e+06	6.578e+05	4.880e+05	4.880e+05
c_2	1	7	4.502e+08	2.219e+07	3.650e+05	6.042e+09	1.290e+09	3.650e+05	3.650e+05
	$n/2$	7	1.053e+06	6.524e+04	2.992e+02	2.973e+07	4.234e+06	2.992e+02	2.992e+02
	n	7	1.077e+04	6.855e+03	1.001e+03	6.743e+04	1.274e+04	1.001e+03	1.001e+03
c_3	1	7	3.202e+02	3.202e+02	3.200e+02	3.206e+02	1.636e-01	3.200e+02	3.200e+02
	$n/2$	7	3.200e+02	3.200e+02	3.200e+02	3.201e+02	1.828e-02	3.200e+02	3.200e+02
	n	7	3.200e+02	3.200e+02	3.200e+02	3.200e+02	6.597e-03	3.200e+02	3.200e+02
c_4	1	7	4.545e+02	4.538e+02	4.299e+02	4.880e+02	1.081e+01	4.299e+02	4.299e+02
	$n/2$	7	4.292e+02	4.279e+02	4.080e+02	4.537e+02	1.158e+01	4.080e+02	4.080e+02
	n	7	4.360e+02	4.328e+02	4.169e+02	4.537e+02	8.049e+00	4.169e+02	4.169e+02
c_5	1	7	1.598e+03	1.669e+03	8.739e+02	2.156e+03	3.017e+02	8.739e+02	8.739e+02
	$n/2$	7	1.059e+03	1.045e+03	6.393e+02	1.886e+03	2.002e+02	6.393e+02	6.393e+02
	n	7	1.040e+03	1.021e+03	7.404e+02	1.771e+03	1.760e+02	7.404e+02	7.404e+02
c_6	1	7	3.629e+05	1.223e+05	1.935e+03	2.113e+06	5.386e+05	1.935e+03	1.935e+03
	$n/2$	7	5.819e+05	3.617e+05	4.892e+03	2.523e+06	6.422e+05	4.892e+03	4.892e+03
	n	7	7.253e+04	3.954e+04	4.158e+03	2.843e+05	7.142e+04	4.158e+03	4.158e+03
c_7	1	7	7.074e+02	7.055e+02	7.016e+02	7.216e+02	5.155e+00	7.016e+02	7.016e+02
	$n/2$	7	7.020e+02	7.019e+02	7.006e+02	7.041e+02	5.708e-01	7.006e+02	7.006e+02
	n	7	7.009e+02	7.011e+02	7.001e+02	7.039e+02	5.904e-01	7.001e+02	7.001e+02
c_8	1	7	1.115e+04	5.101e+03	1.280e+03	1.312e+05	2.172e+04	1.280e+03	1.280e+03
	$n/2$	7	6.016e+04	3.364e+04	1.213e+03	2.989e+05	7.550e+04	1.213e+03	1.213e+03
	n	7	9.755e+04	3.331e+04	1.059e+03	6.407e+05	1.502e+05	1.059e+03	1.059e+03
c_9	1	7	1.006e+03	1.001e+03	1.000e+03	1.061e+03	1.145e+01	1.000e+03	1.000e+03
	$n/2$	7	1.001e+03	1.001e+03	1.000e+03	1.003e+03	5.499e-01	1.000e+03	1.000e+03
	n	7	1.001e+03	1.001e+03	1.000e+03	1.002e+03	2.821e-01	1.000e+03	1.000e+03
c_{10}	1	7	4.844e+04	2.793e+04	3.947e+03	3.617e+05	6.328e+04	3.947e+03	3.947e+03
	$n/2$	7	1.146e+04	5.810e+03	1.465e+03	6.964e+04	1.524e+04	1.465e+03	1.465e+03
	n	7	5.279e+03	2.053e+03	1.332e+03	4.870e+04	8.333e+03	1.332e+03	1.332e+03
c_{11}	1	7	1.403e+03	1.406e+03	1.149e+03	1.427e+03	3.660e+01	1.149e+03	1.149e+03
	$n/2$	7	1.402e+03	1.402e+03	1.401e+03	1.405e+03	8.371e-01	1.401e+03	1.401e+03
	n	7	1.402e+03	1.402e+03	1.401e+03	1.404e+03	7.043e-01	1.401e+03	1.401e+03
c_{12}	1	7	1.323e+03	1.315e+03	1.305e+03	1.400e+03	2.454e+01	1.305e+03	1.305e+03
	$n/2$	7	1.328e+03	1.310e+03	1.303e+03	1.400e+03	3.813e+01	1.303e+03	1.303e+03
	n	7	1.343e+03	1.308e+03	1.302e+03	1.400e+03	4.628e+01	1.302e+03	1.302e+03
c_{13}	1	7	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.059e-04	1.300e+03	1.300e+03
	$n/2$	7	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.010e-04	1.300e+03	1.300e+03
	n	7	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.993e-04	1.300e+03	1.300e+03
c_{14}	1	7	2.635e+03	2.343e+03	1.762e+03	9.155e+03	1.200e+03	1.762e+03	1.762e+03
	$n/2$	7	4.244e+03	1.930e+03	1.506e+03	1.528e+04	3.765e+03	1.506e+03	1.506e+03
	n	7	3.767e+03	1.518e+03	1.500e+03	1.285e+04	3.745e+03	1.500e+03	1.500e+03
c_{15}	1	7	1.629e+03	1.610e+03	1.601e+03	1.944e+03	5.350e+01	1.601e+03	1.601e+03
	$n/2$	7	1.600e+03	1.600e+03	1.600e+03	1.603e+03	4.979e-01	1.600e+03	1.600e+03
	n	7	1.606e+03	1.600e+03	1.600e+03	1.700e+03	2.376e+01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A17: Numerical results for the test set 3 with the tree depth (d) equal to 9.

Benchmark Function	k	d	MVBF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	9	3.299e+06	2.107e+06	3.262e+05	1.329e+07	3.291e+06	3.262e+05	3.262e+05
	$n/2$	9	2.084e+06	1.917e+06	5.154e+05	8.469e+06	1.332e+06	5.154e+05	5.154e+05
	n	9	1.561e+06	9.829e+05	4.036e+05	2.977e+07	4.045e+06	4.036e+05	4.036e+05
c_2	1	9	8.085e+07	3.525e+06	3.722e+03	6.134e+08	1.549e+08	3.722e+03	3.722e+03
	$n/2$	9	2.248e+06	3.467e+03	5.305e+02	1.113e+08	1.558e+07	5.305e+02	5.305e+02
	n	9	2.183e+04	6.481e+03	5.701e+02	4.412e+05	6.665e+04	5.701e+02	5.701e+02
c_3	1	9	3.201e+02	3.201e+02	3.200e+02	3.205e+02	1.282e-01	3.200e+02	3.200e+02
	$n/2$	9	3.200e+02	3.200e+02	3.200e+02	3.201e+02	2.639e-02	3.200e+02	3.200e+02
	n	9	3.200e+02	3.200e+02	3.200e+02	3.200e+02	3.625e-03	3.200e+02	3.200e+02
c_4	1	9	4.477e+02	4.478e+02	4.292e+02	4.772e+02	9.606e+00	4.292e+02	4.292e+02
	$n/2$	9	4.276e+02	4.279e+02	4.080e+02	4.458e+02	1.064e+01	4.080e+02	4.080e+02
	n	9	4.323e+02	4.328e+02	4.179e+02	4.527e+02	7.137e+00	4.179e+02	4.179e+02
c_5	1	9	1.433e+03	1.446e+03	9.338e+02	1.952e+03	2.981e+02	9.338e+02	9.338e+02
	$n/2$	9	1.131e+03	1.102e+03	8.433e+02	1.713e+03	1.869e+02	8.433e+02	8.433e+02
	n	9	1.033e+03	1.005e+03	7.520e+02	1.538e+03	1.594e+02	7.520e+02	7.520e+02
c_6	1	9	1.319e+05	3.142e+04	1.246e+03	1.276e+06	2.667e+05	1.246e+03	1.246e+03
	$n/2$	9	2.139e+05	1.263e+05	5.358e+03	2.438e+06	3.585e+05	5.358e+03	5.358e+03
	n	9	6.483e+04	3.913e+04	2.214e+03	6.115e+05	9.384e+04	2.214e+03	2.214e+03
c_7	1	9	7.061e+02	7.045e+02	7.016e+02	7.214e+02	4.912e+00	7.016e+02	7.016e+02
	$n/2$	9	7.018e+02	7.017e+02	7.009e+02	7.034e+02	4.239e-01	7.009e+02	7.009e+02
	n	9	7.009e+02	7.010e+02	7.002e+02	7.029e+02	4.653e-01	7.002e+02	7.002e+02
c_8	1	9	8.801e+03	3.322e+03	1.376e+03	1.495e+05	2.082e+04	1.376e+03	1.376e+03
	$n/2$	9	4.600e+04	2.600e+04	1.137e+03	6.303e+05	9.114e+04	1.137e+03	1.137e+03
	n	9	6.285e+04	2.591e+04	1.713e+03	4.459e+05	8.513e+04	1.713e+03	1.713e+03
c_9	1	9	1.006e+03	1.001e+03	1.000e+03	1.157e+03	2.246e+01	1.000e+03	1.000e+03
	$n/2$	9	1.001e+03	1.001e+03	1.000e+03	1.004e+03	6.121e-01	1.000e+03	1.000e+03
	n	9	1.001e+03	1.001e+03	1.000e+03	1.002e+03	5.321e-01	1.000e+03	1.000e+03
c_{10}	1	9	4.261e+04	1.036e+04	1.927e+03	5.052e+05	9.047e+04	1.927e+03	1.927e+03
	$n/2$	9	7.895e+03	3.746e+03	1.319e+03	5.751e+04	1.158e+04	1.319e+03	1.319e+03
	n	9	2.905e+03	1.752e+03	1.326e+03	8.974e+03	2.181e+03	1.326e+03	1.326e+03
c_{11}	1	9	1.405e+03	1.403e+03	1.400e+03	1.432e+03	5.974e+00	1.400e+03	1.400e+03
	$n/2$	9	1.402e+03	1.402e+03	1.401e+03	1.404e+03	7.779e-01	1.401e+03	1.401e+03
	n	9	1.402e+03	1.402e+03	1.400e+03	1.405e+03	8.131e-01	1.400e+03	1.400e+03
c_{12}	1	9	1.324e+03	1.311e+03	1.304e+03	1.400e+03	3.112e+01	1.304e+03	1.304e+03
	$n/2$	9	1.328e+03	1.309e+03	1.303e+03	1.400e+03	3.852e+01	1.303e+03	1.303e+03
	n	9	1.328e+03	1.307e+03	1.302e+03	1.400e+03	4.023e+01	1.302e+03	1.302e+03
c_{13}	1	9	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.669e-04	1.300e+03	1.300e+03
	$n/2$	9	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.829e-04	1.300e+03	1.300e+03
	n	9	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.102e-04	1.300e+03	1.300e+03
c_{14}	1	9	2.128e+03	1.919e+03	1.718e+03	4.365e+03	4.928e+02	1.718e+03	1.718e+03
	$n/2$	9	3.182e+03	1.859e+03	1.500e+03	1.318e+04	3.016e+03	1.500e+03	1.500e+03
	n	9	4.477e+03	1.879e+03	1.500e+03	1.375e+04	3.992e+03	1.500e+03	1.500e+03
c_{15}	1	9	1.618e+03	1.607e+03	1.600e+03	1.718e+03	3.067e+01	1.600e+03	1.600e+03
	$n/2$	9	1.603e+03	1.600e+03	1.600e+03	1.700e+03	1.437e+01	1.600e+03	1.600e+03
	n	9	1.600e+03	1.600e+03	1.600e+03	1.601e+03	8.130e-02	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; d - tree depth; MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Tables A18 and A19 show the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on MG-PMA performance in the test set 3 for all values of k and d considered (Tables A13-A17).

Table A18: Average rankings of Aligned Friedman's and Iman-Davenport's tests for comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 3.

i	Algorithm	Rank
14	MG-PMA with $k = 1$ and $d = 2$	201.400
13	MG-PMA with $k = 1$ and $d = 3$	198.600
12	MG-PMA with $k = 1$ and $d = 5$	145.800
11	MG-PMA with $k = n/2$ and $d = 2$	142.333
10	MG-PMA with $k = n/2$ and $d = 3$	115.533
9	MG-PMA with $k = 1$ and $d = 7$	100.200
8	MG-PMA with $k = n$ and $d = 2$	97.933
7	MG-PMA with $k = 1$ and $d = 9$	95.533
6	MG-PMA with $k = n/2$ and $d = 5$	92.333
5	MG-PMA with $k = n$ and $d = 3$	87.800
4	MG-PMA with $k = n$ and $d = 7$	85.533
3	MG-PMA with $k = n/2$ and $d = 7$	83.400
2	MG-PMA with $k = n$ and $d = 5$	83.200
1	MG-PMA with $k = n$ and $d = 9$	82.733
0	MG-PMA with $k = n/2$ and $d = 9$	82.667

Table A19: Holm's procedure for pairwise comparison between MG-PMAs with different values of the number of trees k and the tree depths d for the test set 3. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n/2$ and $d = 9$.

Test	p -value			
Aligned Friedman	0.535			
Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
MG-PMA with $k = 1$ and $d = 2$	4.995	0.000	0.004	Yes
MG-PMA with $k = 1$ and $d = 3$	4.877	0.000	0.004	Yes
MG-PMA with $k = 1$ and $d = 5$	2.656	0.008	0.004	No
MG-PMA with $k = n/2$ and $d = 2$	2.510	0.012	0.005	No
MG-PMA with $k = n/2$ and $d = 3$	1.383	0.167	0.005	No
MG-PMA with $k = 1$ and $d = 7$	0.738	0.461	0.006	No
MG-PMA with $k = n$ and $d = 2$	0.642	0.521	0.006	No
MG-PMA with $k = 1$ and $d = 9$	0.541	0.588	0.007	No
MG-PMA with $k = n/2$ and $d = 5$	0.407	0.684	0.008	No
MG-PMA with $k = n$ and $d = 3$	0.216	0.829	0.010	No
MG-PMA with $k = n$ and $d = 7$	0.121	0.904	0.013	No
MG-PMA with $k = n/2$ and $d = 7$	0.031	0.975	0.017	No
MG-PMA with $k = n$ and $d = 5$	0.023	0.982	0.025	No
MG-PMA with $k = n$ and $d = 9$	0.003	0.998	0.050	No

A3 Numerical Results for MG-PMA

This section presents the numerical results for benchmark-based analysis of MG-PMA to compare with the feedback approach and the analysis of the tournament size and the high-level crossover operator. The results shown in Table A20 presents the results for MG-PMA with tree depth equal to 7, tournament size equal to 100 and high-level crossover operator.

Table A20 shows the mean of the best fitness values (MBFV), the median of the best fitness values (Median), the minimum of the best fitness values (Min), the maximum of the best fitness values (Max), the standard deviation of the best fitness values (Std), the first quartile of the best fitness values (1st Quartile) and the third quartile of the best fitness values (3rd Quartile).

Table A20: Numerical results for MG-PMA with tree depth equal to 7, tournament size equal to 100 and high-level crossover operator.

Benchmark Function	k	MBFV	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	1.058e+07	5.740e+06	1.647e+06	5.935e+07	1.398e+07	4.303e+06	9.182e+06
	$n/2$	2.715e+06	2.555e+06	2.972e+05	6.100e+06	1.200e+06	1.904e+06	3.484e+06
	n	1.236e+06	1.165e+06	6.055e+05	2.794e+06	4.714e+05	8.448e+05	1.557e+06
c_2	1	7.139e+08	1.220e+08	3.754e+05	9.113e+09	1.673e+09	2.266e+07	3.742e+08
	$n/2$	3.613e+05	4.948e+04	8.336e+02	4.727e+06	9.638e+05	6.956e+03	1.917e+05
	n	1.487e+04	8.088e+03	1.181e+03	1.330e+05	2.472e+04	4.998e+03	1.147e+04
c_3	1	3.202e+02	3.201e+02	3.200e+02	3.207e+02	1.772e-01	3.200e+02	3.203e+02
	$n/2$	3.200e+02	3.200e+02	3.200e+02	3.203e+02	6.048e-02	3.200e+02	3.200e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.200e+02	1.694e-04	3.200e+02	3.200e+02
c_4	1	4.515e+02	4.522e+02	4.320e+02	4.795e+02	1.062e+01	4.428e+02	4.579e+02
	$n/2$	4.329e+02	4.331e+02	4.090e+02	4.497e+02	1.025e+01	4.279e+02	4.405e+02
	n	4.338e+02	4.328e+02	4.129e+02	4.527e+02	7.343e+00	4.298e+02	4.408e+02
c_5	1	1.604e+03	1.544e+03	9.686e+02	2.254e+03	2.944e+02	1.403e+03	1.842e+03
	$n/2$	1.093e+03	1.086e+03	7.889e+02	1.584e+03	1.753e+02	9.619e+02	1.179e+03
	n	9.985e+02	9.617e+02	6.220e+02	1.370e+03	1.405e+02	9.617e+02	1.080e+03
c_6	1	3.558e+05	1.007e+05	8.338e+03	2.451e+06	5.198e+05	3.422e+04	4.527e+05
	$n/2$	6.416e+05	3.616e+05	3.499e+03	5.275e+06	1.006e+06	1.655e+05	7.264e+05
	n	3.379e+04	2.062e+04	1.303e+03	1.816e+05	3.726e+04	1.071e+04	4.663e+04
c_7	1	7.085e+02	7.061e+02	7.021e+02	7.291e+02	6.180e+00	7.043e+02	7.112e+02
	$n/2$	7.020e+02	7.018e+02	7.012e+02	7.041e+02	5.833e-01	7.016e+02	7.021e+02
	n	7.009e+02	7.011e+02	7.002e+02	7.015e+02	3.358e-01	7.008e+02	7.011e+02
c_8	1	3.936e+04	3.677e+03	1.503e+03	1.462e+06	2.039e+05	2.232e+03	1.119e+04
	$n/2$	1.068e+05	3.922e+04	1.269e+03	8.356e+05	1.732e+05	1.968e+04	8.176e+04
	n	8.318e+04	3.662e+04	1.617e+03	4.512e+05	9.766e+04	1.027e+04	1.381e+05
c_9	1	1.004e+03	1.001e+03	1.000e+03	1.053e+03	8.740e+00	1.001e+03	1.004e+03
	$n/2$	1.001e+03	1.001e+03	1.000e+03	1.004e+03	5.717e-01	1.000e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	2.790e-01	1.001e+03	1.001e+03
c_{10}	1	4.036e+04	2.379e+04	1.950e+03	3.816e+05	6.179e+04	7.841e+03	4.123e+04
	$n/2$	1.105e+04	5.581e+03	1.356e+03	4.524e+04	1.163e+04	2.061e+03	2.088e+04
	n	5.123e+03	3.478e+03	1.385e+03	4.419e+04	6.935e+03	1.532e+03	5.954e+03
c_{11}	1	1.403e+03	1.405e+03	1.149e+03	1.441e+03	3.691e+01	1.403e+03	1.408e+03
	$n/2$	1.402e+03	1.402e+03	1.401e+03	1.404e+03	8.516e-01	1.401e+03	1.403e+03
	n	1.402e+03	1.402e+03	1.401e+03	1.404e+03	6.668e-01	1.401e+03	1.402e+03
c_{12}	1	1.322e+03	1.314e+03	1.304e+03	1.400e+03	2.310e+01	1.311e+03	1.319e+03
	$n/2$	1.319e+03	1.308e+03	1.303e+03	1.400e+03	3.006e+01	1.306e+03	1.313e+03
	n	1.330e+03	1.307e+03	1.303e+03	1.400e+03	4.137e+01	1.304e+03	1.379e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.340e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.181e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.892e-04	1.300e+03	1.300e+03
c_{14}	1	2.644e+03	2.408e+03	1.757e+03	1.032e+04	1.270e+03	2.074e+03	2.681e+03
	$n/2$	3.789e+03	1.925e+03	1.504e+03	1.308e+04	3.305e+03	1.534e+03	7.520e+03
	n	4.484e+03	1.859e+03	1.500e+03	1.375e+04	4.198e+03	1.502e+03	7.540e+03
c_{15}	1	1.634e+03	1.613e+03	1.602e+03	1.881e+03	4.796e+01	1.606e+03	1.642e+03
	$n/2$	1.607e+03	1.600e+03	1.600e+03	1.700e+03	2.378e+01	1.600e+03	1.600e+03
	n	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.400e+01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

A4 Numerical Results for MG-PMA with Feedback

This section presents the numerical results for benchmark-based analysis of MG-PMA with feedback. The results shown in Table A20 presents the results for MG-PMA with tree depth equal to 7, tournament size equal to 100 and high-level crossover operator. The numerical results of MG-PMA with feedback are presented in Table A21.

Tables A20 and A21 show the mean of the best fitness values (MBFV), the median of the best fitness values (Median), the minimum of the best fitness values (Min), the maximum of the best fitness values (Max), the standard deviation of the best fitness values (Std), the first quartile of the best fitness values (1st Quartile) and the third quartile of the best fitness values (3rd Quartile).

Table A21: Numerical results for MG-PMA with feedback using the test set 3 with the tree depth (d) equal to 7.

Benchmark Function	k	MBFV	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	5.481e+07	1.241e+07	2.027e+06	1.698e+08	6.865e+07	7.668e+06	1.183e+08
	$n/2$	5.713e+07	5.962e+07	5.770e+06	1.163e+08	2.206e+07	4.307e+07	7.216e+07
	n	8.297e+07	7.287e+07	1.288e+07	1.698e+08	4.075e+07	5.875e+07	1.118e+08
c_2	1	3.834e+09	6.935e+08	2.979e+04	2.204e+10	6.867e+09	1.683e+08	3.420e+09
	$n/2$	8.753e+09	6.714e+09	1.279e+09	1.973e+10	5.745e+09	3.406e+09	1.368e+10
	n	1.219e+10	1.190e+10	4.273e+09	2.204e+10	3.102e+09	1.037e+10	1.405e+10
c_3	1	3.204e+02	3.203e+02	3.202e+02	3.206e+02	7.782e-02	3.203e+02	3.204e+02
	$n/2$	3.204e+02	3.204e+02	3.201e+02	3.207e+02	1.292e-01	3.203e+02	3.205e+02
	n	3.202e+02	3.202e+02	3.200e+02	3.205e+02	8.386e-02	3.202e+02	3.202e+02
c_4	1	4.611e+02	4.590e+02	4.293e+02	5.108e+02	1.825e+01	4.461e+02	4.682e+02
	$n/2$	4.644e+02	4.641e+02	4.442e+02	4.835e+02	1.100e+01	4.566e+02	4.733e+02
	n	4.658e+02	4.665e+02	4.445e+02	4.875e+02	1.060e+01	4.570e+02	4.731e+02
c_5	1	1.563e+03	1.525e+03	8.173e+02	2.379e+03	4.060e+02	1.272e+03	1.822e+03
	$n/2$	1.788e+03	1.813e+03	1.277e+03	2.356e+03	2.747e+02	1.593e+03	1.965e+03
	n	1.815e+03	1.809e+03	1.236e+03	2.320e+03	2.538e+02	1.621e+03	2.008e+03
c_6	1	1.341e+07	2.671e+05	2.002e+03	1.251e+08	3.751e+07	4.523e+04	1.226e+06
	$n/2$	2.294e+06	1.143e+06	1.447e+04	2.025e+07	3.924e+06	3.176e+05	1.806e+06
	n	2.167e+07	1.134e+07	4.761e+04	1.251e+08	2.698e+07	2.595e+06	3.075e+07
c_7	1	7.175e+02	7.092e+02	7.021e+02	7.827e+02	1.974e+01	7.044e+02	7.255e+02
	$n/2$	7.148e+02	7.138e+02	7.045e+02	7.489e+02	9.718e+00	7.073e+02	7.165e+02
	n	7.508e+02	7.323e+02	7.059e+02	8.972e+02	4.254e+01	7.229e+02	7.717e+02
c_8	1	9.438e+03	5.184e+03	1.620e+03	4.589e+04	1.039e+04	2.529e+03	1.246e+04
	$n/2$	8.310e+04	7.551e+03	1.620e+03	1.940e+06	3.198e+05	2.505e+03	1.898e+04
	n	3.401e+06	2.145e+06	7.209e+04	1.303e+07	2.684e+06	1.871e+06	3.866e+06
c_9	1	1.033e+03	1.015e+03	1.000e+03	1.254e+03	5.346e+01	1.001e+03	1.045e+03
	$n/2$	1.036e+03	1.038e+03	1.008e+03	1.091e+03	1.814e+01	1.021e+03	1.047e+03
	n	1.075e+03	1.072e+03	1.025e+03	1.206e+03	3.441e+01	1.051e+03	1.092e+03
c_{10}	1	2.479e+05	2.175e+05	2.819e+03	6.254e+05	2.216e+05	4.021e+04	4.370e+05
	$n/2$	4.475e+05	2.880e+05	1.363e+04	2.221e+06	4.758e+05	1.490e+05	6.254e+05
	n	4.780e+05	3.597e+05	5.682e+03	4.200e+06	6.652e+05	1.468e+05	6.254e+05
c_{11}	1	1.436e+03	1.411e+03	1.403e+03	1.767e+03	8.211e+01	1.406e+03	1.419e+03
	$n/2$	1.439e+03	1.429e+03	1.404e+03	1.539e+03	2.957e+01	1.417e+03	1.452e+03
	n	1.439e+03	1.427e+03	1.406e+03	1.554e+03	3.196e+01	1.418e+03	1.450e+03
c_{12}	1	1.344e+03	1.328e+03	1.304e+03	1.400e+03	3.859e+01	1.311e+03	1.400e+03
	$n/2$	1.332e+03	1.329e+03	1.310e+03	1.400e+03	2.041e+01	1.318e+03	1.332e+03
	n	1.364e+03	1.355e+03	1.317e+03	1.400e+03	2.642e+01	1.344e+03	1.400e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.377e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.112e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	6.863e-05	1.300e+03	1.300e+03
c_{14}	1	4.974e+03	3.761e+03	2.058e+03	1.786e+04	3.028e+03	3.032e+03	5.981e+03
	$n/2$	8.114e+03	7.290e+03	2.264e+03	1.587e+04	2.950e+03	6.305e+03	9.904e+03
	n	1.528e+04	1.566e+04	8.092e+03	1.839e+04	2.504e+03	1.419e+04	1.764e+04
c_{15}	1	1.706e+03	1.700e+03	1.600e+03	2.557e+03	1.717e+02	1.636e+03	1.700e+03
	$n/2$	2.000e+03	1.796e+03	1.652e+03	3.877e+03	5.367e+02	1.700e+03	1.995e+03
	n	2.015e+03	1.988e+03	1.700e+03	2.628e+03	2.175e+02	1.867e+03	2.074e+03

k - Number of functions incorporated in an MG-PMA individual; MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A22 shows the results of Aligned Friedman’s and Iman-Davenport’s tests, along with the Holm’s procedure based on the performance of MG-PMA and MG-PMA with feedback methods (Tables A20 and A21).

Table A22: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between MG-PMA and MG-PMA with feedback. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n$.

i	Algorithm	Rank
5	MG-PMA with feedback with $k = 1$	59.500
4	MG-PMA with feedback with $k = n/2$	58.567
3	MG-PMA with feedback with $k = n$	49.100
2	MG-PMA with $k = 1$	45.033
1	MG-PMA with $k = n/2$	30.633
0	MG-PMA with $k = n$	30.167

Test	p -value
Aligned Friedman	0.025

Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
MG-PMA with $k = 1$ with feedback	3.075	0.002	0.010	Yes
MG-PMA with $k = n/2$ with feedback	2.977	0.003	0.013	Yes
MG-PMA with $k = n$ with feedback	1.985	0.047	0.017	No
MG-PMA with $k = 1$	1.559	0.119	0.025	No
MG-PMA with $k = n/2$	0.049	0.961	0.050	No

A5

Numerical Results for Analysis of Tournament Size and Crossover

This section presents the numerical results for benchmark-based analysis of the tournament size (Tables A23-A25) and the high-level crossover operator (Tables A26-A29).

Tables A23-A29 show the mean of the best fitness values (MBFV), the median of the best fitness values (Median), the minimum of the best fitness values (Min), the maximum of the best fitness values (Max), the standard deviation of the best fitness values (Std), the first quartile of the best fitness values (1st Quartile) and the third quartile of the best fitness values (3rd Quartile).

Table A23: Numerical results for MG-PMA with tournament size equal to 2 and the tree depth (d) equal to 7 using the test set 3.

Benchmark Function	k	MBVF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	6.878e+06	6.822e+06	1.092e+06	1.710e+07	3.492e+06	4.211e+06	9.122e+06
	$n/2$	9.698e+06	9.818e+06	3.309e+06	1.677e+07	3.042e+06	7.549e+06	1.171e+07
	n	5.000e+06	5.081e+06	1.044e+06	1.164e+07	2.355e+06	3.437e+06	6.632e+06
c_2	1	1.502e+08	9.567e+07	2.137e+06	1.457e+09	2.226e+08	3.047e+07	1.977e+08
	$n/2$	1.143e+08	2.449e+07	1.005e+05	2.767e+09	4.002e+08	8.709e+06	5.529e+07
	n	2.338e+06	5.078e+05	2.746e+03	4.663e+07	6.768e+06	1.134e+05	2.066e+06
c_3	1	3.201e+02	3.201e+02	3.200e+02	3.204e+02	7.232e-02	3.200e+02	3.201e+02
	$n/2$	3.201e+02	3.200e+02	3.200e+02	3.203e+02	5.254e-02	3.200e+02	3.201e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.201e+02	1.876e-02	3.200e+02	3.200e+02
c_4	1	4.462e+02	4.451e+02	4.349e+02	4.653e+02	8.135e+00	4.402e+02	4.520e+02
	$n/2$	4.285e+02	4.266e+02	4.152e+02	4.518e+02	7.721e+00	4.231e+02	4.322e+02
	n	4.347e+02	4.342e+02	4.163e+02	4.528e+02	8.251e+00	4.281e+02	4.416e+02
c_5	1	1.332e+03	1.352e+03	8.349e+02	1.723e+03	1.913e+02	1.186e+03	1.441e+03
	$n/2$	1.068e+03	1.059e+03	7.487e+02	1.323e+03	1.194e+02	9.723e+02	1.179e+03
	n	9.875e+02	9.639e+02	7.481e+02	1.208e+03	9.433e+01	9.619e+02	1.063e+03
c_6	1	6.044e+04	2.085e+04	3.041e+03	6.279e+05	1.052e+05	9.549e+03	7.420e+04
	$n/2$	9.001e+04	3.492e+04	1.570e+03	1.114e+06	2.112e+05	1.721e+04	4.258e+04
	n	2.983e+04	3.097e+04	2.192e+03	1.812e+05	2.572e+04	1.430e+04	3.663e+04
c_7	1	7.041e+02	7.036e+02	7.019e+02	7.122e+02	1.944e+00	7.028e+02	7.049e+02
	$n/2$	7.025e+02	7.023e+02	7.013e+02	7.052e+02	8.139e-01	7.019e+02	7.033e+02
	n	7.015e+02	7.015e+02	7.006e+02	7.033e+02	4.955e-01	7.012e+02	7.018e+02
c_8	1	5.068e+03	2.770e+03	1.194e+03	1.930e+04	4.406e+03	1.916e+03	8.239e+03
	$n/2$	3.046e+03	1.703e+03	1.004e+03	3.475e+04	4.928e+03	1.519e+03	2.507e+03
	n	2.917e+03	2.207e+03	1.114e+03	1.086e+04	2.185e+03	1.553e+03	3.151e+03
c_9	1	1.002e+03	1.001e+03	1.000e+03	1.015e+03	2.574e+00	1.001e+03	1.002e+03
	$n/2$	1.001e+03	1.001e+03	1.000e+03	1.005e+03	1.232e+00	1.001e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	3.590e-01	1.001e+03	1.001e+03
c_{10}	1	2.080e+04	1.373e+04	1.771e+03	9.250e+04	2.026e+04	6.871e+03	2.701e+04
	$n/2$	6.871e+03	2.766e+03	1.659e+03	3.048e+04	7.699e+03	2.123e+03	8.343e+03
	n	3.778e+03	1.687e+03	1.340e+03	1.208e+04	3.211e+03	1.449e+03	5.646e+03
c_{11}	1	1.405e+03	1.405e+03	1.400e+03	1.413e+03	2.499e+00	1.403e+03	1.406e+03
	$n/2$	1.404e+03	1.404e+03	1.401e+03	1.407e+03	1.357e+00	1.403e+03	1.405e+03
	n	1.370e+03	1.402e+03	1.111e+03	1.406e+03	9.065e+01	1.401e+03	1.403e+03
c_{12}	1	1.311e+03	1.309e+03	1.303e+03	1.400e+03	1.372e+01	1.306e+03	1.311e+03
	$n/2$	1.308e+03	1.308e+03	1.304e+03	1.313e+03	2.468e+00	1.306e+03	1.310e+03
	n	1.306e+03	1.306e+03	1.302e+03	1.310e+03	1.925e+00	1.304e+03	1.308e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.687e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.794e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.874e-04	1.300e+03	1.300e+03
c_{14}	1	2.289e+03	2.194e+03	1.730e+03	3.320e+03	3.733e+02	2.005e+03	2.458e+03
	$n/2$	2.908e+03	2.240e+03	1.769e+03	1.261e+04	2.126e+03	2.091e+03	2.510e+03
	n	2.839e+03	2.000e+03	1.525e+03	1.261e+04	2.319e+03	1.666e+03	2.374e+03
c_{15}	1	1.619e+03	1.614e+03	1.601e+03	1.700e+03	1.940e+01	1.608e+03	1.624e+03
	$n/2$	1.609e+03	1.604e+03	1.601e+03	1.700e+03	1.926e+01	1.602e+03	1.607e+03
	n	1.603e+03	1.601e+03	1.600e+03	1.700e+03	1.399e+01	1.600e+03	1.602e+03

k - Number of functions incorporated in an MG-PMA individual; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A24: Numerical results for MG-PMA with tournament size equal to 25 and the tree depth (d) equal to 7 using the test set 3.

Benchmark Function	k	MBVF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	7.575e+06	6.699e+06	5.626e+05	2.429e+07	4.869e+06	3.932e+06	9.847e+06
	$n/2$	3.431e+06	2.483e+06	7.397e+05	1.565e+07	2.748e+06	1.904e+06	4.033e+06
	n	1.227e+06	1.081e+06	5.864e+05	2.647e+06	5.076e+05	8.379e+05	1.634e+06
c_2	1	3.209e+08	6.469e+07	5.018e+05	4.703e+09	8.580e+08	1.479e+07	2.529e+08
	$n/2$	6.629e+05	1.415e+04	1.214e+03	7.697e+06	1.680e+06	4.394e+03	2.536e+05
	n	2.513e+04	9.262e+03	1.063e+03	7.180e+05	9.970e+04	3.421e+03	1.526e+04
c_3	1	3.202e+02	3.201e+02	3.200e+02	3.206e+02	1.465e-01	3.200e+02	3.203e+02
	$n/2$	3.200e+02	3.200e+02	3.200e+02	3.201e+02	2.059e-02	3.200e+02	3.200e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.200e+02	3.678e-03	3.200e+02	3.200e+02
c_4	1	4.503e+02	4.470e+02	4.239e+02	4.721e+02	1.088e+01	4.435e+02	4.583e+02
	$n/2$	4.317e+02	4.328e+02	4.090e+02	4.497e+02	1.049e+01	4.271e+02	4.408e+02
	n	4.338e+02	4.319e+02	4.129e+02	4.597e+02	1.026e+01	4.269e+02	4.408e+02
c_5	1	1.525e+03	1.464e+03	9.377e+02	2.036e+03	2.592e+02	1.343e+03	1.720e+03
	$n/2$	1.082e+03	1.088e+03	6.230e+02	1.712e+03	2.152e+02	9.618e+02	1.179e+03
	n	1.006e+03	9.618e+02	7.404e+02	1.366e+03	1.548e+02	8.433e+02	1.143e+03
c_6	1	2.289e+05	9.448e+04	2.304e+03	1.404e+06	3.298e+05	4.488e+04	2.371e+05
	$n/2$	3.952e+05	2.848e+05	5.325e+03	1.509e+06	3.496e+05	1.423e+05	5.726e+05
	n	5.660e+04	3.271e+04	3.496e+03	3.581e+05	7.165e+04	1.986e+04	6.309e+04
c_7	1	7.062e+02	7.050e+02	7.015e+02	7.212e+02	4.639e+00	7.037e+02	7.063e+02
	$n/2$	7.019e+02	7.019e+02	7.012e+02	7.034e+02	4.184e-01	7.017e+02	7.021e+02
	n	7.009e+02	7.009e+02	7.001e+02	7.046e+02	7.036e-01	7.004e+02	7.011e+02
c_8	1	6.729e+03	5.983e+03	1.448e+03	1.959e+04	4.501e+03	2.759e+03	9.237e+03
	$n/2$	2.774e+04	1.583e+04	9.904e+02	1.439e+05	3.195e+04	6.573e+03	3.378e+04
	n	5.934e+04	3.222e+04	1.476e+03	4.290e+05	7.619e+04	1.219e+04	7.900e+04
c_9	1	1.004e+03	1.001e+03	1.000e+03	1.045e+03	8.063e+00	1.001e+03	1.002e+03
	$n/2$	1.001e+03	1.001e+03	1.000e+03	1.003e+03	4.616e-01	1.000e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	3.093e-01	1.000e+03	1.001e+03
c_{10}	1	2.835e+04	1.190e+04	2.719e+03	1.852e+05	3.717e+04	7.022e+03	3.514e+04
	$n/2$	1.083e+04	5.705e+03	1.384e+03	4.931e+04	1.075e+04	2.169e+03	1.732e+04
	n	7.443e+03	1.773e+03	1.330e+03	1.994e+05	2.767e+04	1.489e+03	5.527e+03
c_{11}	1	1.401e+03	1.406e+03	1.117e+03	1.421e+03	4.100e+01	1.403e+03	1.410e+03
	$n/2$	1.402e+03	1.402e+03	1.401e+03	1.404e+03	7.217e-01	1.401e+03	1.402e+03
	n	1.402e+03	1.402e+03	1.401e+03	1.405e+03	8.438e-01	1.401e+03	1.403e+03
c_{12}	1	1.327e+03	1.313e+03	1.303e+03	1.400e+03	3.236e+01	1.308e+03	1.323e+03
	$n/2$	1.317e+03	1.308e+03	1.303e+03	1.400e+03	2.783e+01	1.306e+03	1.310e+03
	n	1.321e+03	1.306e+03	1.302e+03	1.400e+03	3.468e+01	1.304e+03	1.309e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.393e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.869e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.214e-04	1.300e+03	1.300e+03
c_{14}	1	2.672e+03	2.304e+03	1.751e+03	9.141e+03	1.342e+03	2.144e+03	2.618e+03
	$n/2$	2.913e+03	1.897e+03	1.504e+03	1.301e+04	2.766e+03	1.615e+03	1.975e+03
	n	4.323e+03	1.920e+03	1.500e+03	1.263e+04	3.770e+03	1.503e+03	7.429e+03
c_{15}	1	1.630e+03	1.613e+03	1.601e+03	1.847e+03	4.324e+01	1.607e+03	1.630e+03
	$n/2$	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.396e+01	1.600e+03	1.601e+03
	n	1.600e+03	1.600e+03	1.600e+03	1.600e+03	6.044e-02	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A25: Numerical results for MG-PMA with tournament size equal to 50 and the tree depth (d) equal to 7 using the test set 3.

Benchmark Function	k	MBVF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	9.000e+06	5.655e+06	1.235e+06	6.011e+07	1.132e+07	5.655e+06	8.379e+06
	$n/2$	3.215e+06	3.091e+06	9.358e+05	6.998e+06	1.341e+06	3.091e+06	4.284e+06
	n	1.339e+06	1.090e+06	5.437e+05	4.526e+06	7.372e+05	1.090e+06	1.546e+06
c_2	1	1.054e+09	9.925e+07	5.371e+05	1.014e+10	2.311e+09	9.925e+07	4.539e+08
	$n/2$	7.897e+05	2.966e+04	9.667e+02	1.937e+07	2.810e+06	2.966e+04	2.834e+05
	n	2.165e+04	8.414e+03	1.210e+03	5.035e+05	7.187e+04	8.414e+03	1.106e+04
c_3	1	3.202e+02	3.201e+02	3.200e+02	3.206e+02	1.400e-01	3.201e+02	3.203e+02
	$n/2$	3.200e+02	3.200e+02	3.200e+02	3.201e+02	1.359e-02	3.200e+02	3.200e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.200e+02	1.153e-03	3.200e+02	3.200e+02
c_4	1	4.523e+02	4.518e+02	4.367e+02	4.777e+02	9.895e+00	4.518e+02	4.598e+02
	$n/2$	4.337e+02	4.338e+02	4.090e+02	4.523e+02	1.044e+01	4.338e+02	4.428e+02
	n	4.311e+02	4.328e+02	4.129e+02	4.438e+02	8.024e+00	4.328e+02	4.358e+02
c_5	1	1.602e+03	1.542e+03	7.851e+02	2.367e+03	3.133e+02	1.542e+03	1.811e+03
	$n/2$	1.077e+03	1.064e+03	7.557e+02	1.619e+03	2.005e+02	1.064e+03	1.182e+03
	n	1.013e+03	1.007e+03	7.404e+02	1.469e+03	1.379e+02	1.007e+03	1.095e+03
c_6	1	3.087e+05	1.645e+05	2.934e+03	1.277e+06	3.549e+05	1.645e+05	4.740e+05
	$n/2$	5.833e+05	3.937e+05	1.085e+04	3.721e+06	6.890e+05	3.937e+05	6.689e+05
	n	6.401e+04	2.267e+04	2.100e+03	4.712e+05	9.996e+04	2.267e+04	6.221e+04
c_7	1	7.080e+02	7.055e+02	7.022e+02	7.274e+02	6.388e+00	7.055e+02	7.076e+02
	$n/2$	7.019e+02	7.018e+02	7.009e+02	7.039e+02	5.396e-01	7.018e+02	7.020e+02
	n	7.009e+02	7.010e+02	7.001e+02	7.014e+02	3.331e-01	7.010e+02	7.012e+02
c_8	1	3.543e+04	4.494e+03	1.477e+03	1.406e+06	1.960e+05	4.494e+03	1.044e+04
	$n/2$	7.736e+04	2.379e+04	1.921e+03	8.428e+05	1.624e+05	2.379e+04	5.334e+04
	n	9.720e+04	3.452e+04	1.606e+03	1.074e+06	1.802e+05	3.452e+04	8.877e+04
c_9	1	1.006e+03	1.001e+03	1.000e+03	1.057e+03	1.053e+01	1.001e+03	1.006e+03
	$n/2$	1.001e+03	1.001e+03	1.000e+03	1.004e+03	5.282e-01	1.001e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	3.033e-01	1.001e+03	1.001e+03
c_{10}	1	3.364e+04	1.299e+04	2.260e+03	3.374e+05	5.261e+04	1.299e+04	4.161e+04
	$n/2$	1.034e+04	5.092e+03	1.481e+03	3.938e+04	1.051e+04	5.092e+03	1.687e+04
	n	3.572e+03	1.889e+03	1.287e+03	1.055e+04	2.697e+03	1.889e+03	5.843e+03
c_{11}	1	1.408e+03	1.406e+03	1.401e+03	1.434e+03	6.153e+00	1.406e+03	1.410e+03
	$n/2$	1.402e+03	1.402e+03	1.401e+03	1.404e+03	6.960e-01	1.402e+03	1.402e+03
	n	1.402e+03	1.402e+03	1.400e+03	1.405e+03	8.359e-01	1.402e+03	1.402e+03
c_{12}	1	1.325e+03	1.312e+03	1.303e+03	1.400e+03	3.112e+01	1.312e+03	1.317e+03
	$n/2$	1.317e+03	1.308e+03	1.302e+03	1.400e+03	2.779e+01	1.308e+03	1.310e+03
	n	1.326e+03	1.307e+03	1.302e+03	1.400e+03	3.917e+01	1.307e+03	1.311e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.401e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.882e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.065e-04	1.300e+03	1.300e+03
c_{14}	1	2.711e+03	2.443e+03	1.772e+03	1.245e+04	1.487e+03	2.443e+03	2.907e+03
	$n/2$	3.775e+03	1.941e+03	1.504e+03	1.313e+04	3.241e+03	1.941e+03	6.642e+03
	n	4.119e+03	1.868e+03	1.500e+03	1.285e+04	3.641e+03	1.868e+03	7.413e+03
c_{15}	1	1.631e+03	1.614e+03	1.601e+03	1.700e+03	3.513e+01	1.614e+03	1.635e+03
	$n/2$	1.601e+03	1.600e+03	1.600e+03	1.626e+03	3.582e+00	1.600e+03	1.600e+03
	n	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.400e+01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A26: Numerical results for MG-PMA with restricted high-level crossover and the tree depth (d) equal to 7 using the test set 3.

Benchmark Function	k	MBVF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	9.561e+06	7.207e+06	6.654e+05	7.261e+07	1.207e+07	3.387e+06	1.015e+07
	$n/2$	2.716e+06	2.313e+06	5.277e+05	6.592e+06	1.393e+06	1.517e+06	3.619e+06
	n	1.739e+06	1.205e+06	2.642e+05	1.760e+07	2.431e+06	9.580e+05	1.707e+06
c_2	1	8.220e+08	4.435e+07	3.058e+05	8.007e+09	1.912e+09	1.355e+07	2.538e+08
	$n/2$	3.438e+05	2.469e+04	3.982e+02	2.790e+06	7.259e+05	5.190e+03	1.580e+05
	n	1.422e+04	6.126e+03	5.149e+02	1.640e+05	2.595e+04	3.117e+03	1.182e+04
c_3	1	3.202e+02	3.202e+02	3.200e+02	3.206e+02	1.742e-01	3.200e+02	3.203e+02
	$n/2$	3.200e+02	3.200e+02	3.200e+02	3.202e+02	3.798e-02	3.200e+02	3.200e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.200e+02	2.044e-03	3.200e+02	3.200e+02
c_4	1	4.520e+02	4.522e+02	4.181e+02	4.829e+02	1.299e+01	4.443e+02	4.632e+02
	$n/2$	4.311e+02	4.305e+02	4.090e+02	4.497e+02	1.029e+01	4.241e+02	4.408e+02
	n	4.352e+02	4.328e+02	4.129e+02	4.527e+02	8.842e+00	4.298e+02	4.433e+02
c_5	1	1.600e+03	1.565e+03	1.009e+03	2.172e+03	2.817e+02	1.400e+03	1.756e+03
	$n/2$	1.060e+03	1.043e+03	6.325e+02	1.752e+03	2.032e+02	9.617e+02	1.144e+03
	n	1.039e+03	9.655e+02	8.433e+02	1.512e+03	1.708e+02	9.617e+02	1.117e+03
c_6	1	3.936e+05	2.001e+05	2.035e+03	2.010e+06	4.635e+05	7.218e+04	5.672e+05
	$n/2$	4.431e+05	2.633e+05	1.309e+04	2.303e+06	4.876e+05	1.466e+05	5.807e+05
	n	6.538e+04	3.422e+04	9.422e+02	6.159e+05	1.072e+05	1.008e+04	7.879e+04
c_7	1	7.070e+02	7.058e+02	7.016e+02	7.212e+02	4.397e+00	7.038e+02	7.082e+02
	$n/2$	7.021e+02	7.019e+02	7.011e+02	7.045e+02	6.816e-01	7.017e+02	7.022e+02
	n	7.009e+02	7.010e+02	7.001e+02	7.025e+02	4.328e-01	7.006e+02	7.011e+02
c_8	1	1.895e+04	7.389e+03	1.597e+03	2.436e+05	4.239e+04	2.474e+03	1.311e+04
	$n/2$	1.495e+05	3.918e+04	1.574e+03	3.760e+06	5.240e+05	1.578e+04	1.064e+05
	n	8.270e+04	3.603e+04	2.435e+03	6.610e+05	1.230e+05	1.125e+04	1.279e+05
c_9	1	1.005e+03	1.001e+03	1.000e+03	1.044e+03	9.457e+00	1.001e+03	1.005e+03
	$n/2$	1.004e+03	1.001e+03	1.000e+03	1.153e+03	2.140e+01	1.000e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	3.709e-01	1.000e+03	1.001e+03
c_{10}	1	5.281e+04	4.079e+04	2.077e+03	1.765e+05	4.546e+04	1.157e+04	9.190e+04
	$n/2$	1.127e+04	3.877e+03	1.381e+03	8.197e+04	1.563e+04	2.243e+03	1.246e+04
	n	4.270e+03	2.008e+03	1.354e+03	2.398e+04	4.364e+03	1.498e+03	5.702e+03
c_{11}	1	1.408e+03	1.406e+03	1.401e+03	1.485e+03	1.202e+01	1.403e+03	1.409e+03
	$n/2$	1.402e+03	1.402e+03	1.401e+03	1.404e+03	6.908e-01	1.401e+03	1.403e+03
	n	1.402e+03	1.402e+03	1.401e+03	1.404e+03	6.282e-01	1.401e+03	1.402e+03
c_{12}	1	1.322e+03	1.314e+03	1.305e+03	1.400e+03	2.676e+01	1.309e+03	1.317e+03
	$n/2$	1.320e+03	1.308e+03	1.302e+03	1.400e+03	3.218e+01	1.306e+03	1.310e+03
	n	1.334e+03	1.307e+03	1.302e+03	1.400e+03	4.321e+01	1.305e+03	1.400e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.574e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.151e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.724e-04	1.300e+03	1.300e+03
c_{14}	1	2.405e+03	2.375e+03	1.773e+03	5.726e+03	5.789e+02	2.072e+03	2.515e+03
	$n/2$	4.321e+03	1.917e+03	1.502e+03	1.261e+04	3.685e+03	1.830e+03	7.577e+03
	n	3.413e+03	1.511e+03	1.500e+03	1.263e+04	3.446e+03	1.502e+03	2.026e+03
c_{15}	1	1.635e+03	1.613e+03	1.601e+03	1.840e+03	4.569e+01	1.608e+03	1.651e+03
	$n/2$	1.606e+03	1.600e+03	1.600e+03	1.700e+03	2.371e+01	1.600e+03	1.600e+03
	n	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.400e+01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A27: Numerical results for MG-PMA with high-level single-point crossover and the tree depth (d) equal to 7 using the test set 3.

Benchmark Function	k	MBVF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	9.980e+06	7.623e+06	1.341e+06	4.748e+07	8.754e+06	5.161e+06	1.150e+07
	$n/2$	3.407e+06	2.827e+06	6.111e+05	8.450e+06	1.730e+06	2.154e+06	4.467e+06
	n	1.275e+06	1.181e+06	4.469e+05	2.673e+06	5.266e+05	8.585e+05	1.655e+06
c_2	1	7.010e+08	6.745e+07	4.391e+05	9.113e+09	1.823e+09	1.264e+07	5.018e+08
	$n/2$	1.033e+06	5.047e+04	4.411e+02	1.950e+07	3.051e+06	1.104e+04	3.192e+05
	n	1.134e+04	8.470e+03	8.342e+02	1.135e+05	1.842e+04	3.364e+03	1.165e+04
c_3	1	3.202e+02	3.202e+02	3.200e+02	3.206e+02	1.587e-01	3.201e+02	3.203e+02
	$n/2$	3.200e+02	3.200e+02	3.200e+02	3.201e+02	3.270e-02	3.200e+02	3.200e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.200e+02	8.716e-04	3.200e+02	3.200e+02
c_4	1	4.495e+02	4.480e+02	4.273e+02	4.755e+02	1.038e+01	4.431e+02	4.562e+02
	$n/2$	4.325e+02	4.328e+02	4.060e+02	4.472e+02	9.967e+00	4.269e+02	4.408e+02
	n	4.317e+02	4.328e+02	4.129e+02	4.507e+02	8.063e+00	4.279e+02	4.358e+02
c_5	1	1.628e+03	1.592e+03	9.917e+02	2.266e+03	2.712e+02	1.427e+03	1.786e+03
	$n/2$	1.068e+03	9.905e+02	7.472e+02	1.556e+03	2.008e+02	9.617e+02	1.200e+03
	n	1.024e+03	1.036e+03	6.221e+02	1.611e+03	1.748e+02	8.635e+02	1.090e+03
c_6	1	3.287e+05	1.544e+05	8.962e+03	1.803e+06	4.095e+05	4.515e+04	5.216e+05
	$n/2$	4.731e+05	3.104e+05	2.493e+04	2.605e+06	5.082e+05	1.238e+05	7.070e+05
	n	1.098e+05	3.462e+04	2.525e+03	1.778e+06	2.981e+05	1.265e+04	6.522e+04
c_7	1	7.077e+02	7.054e+02	7.021e+02	7.236e+02	5.931e+00	7.039e+02	7.085e+02
	$n/2$	7.021e+02	7.019e+02	7.011e+02	7.052e+02	7.799e-01	7.017e+02	7.024e+02
	n	7.009e+02	7.010e+02	7.002e+02	7.021e+02	3.615e-01	7.006e+02	7.011e+02
c_8	1	1.104e+04	3.836e+03	1.187e+03	1.124e+05	2.080e+04	2.161e+03	1.020e+04
	$n/2$	1.559e+05	1.066e+05	5.783e+03	6.226e+05	1.444e+05	6.206e+04	1.961e+05
	n	5.793e+05	3.374e+05	1.415e+03	9.586e+06	1.315e+06	2.191e+05	5.124e+05
c_9	1	1.005e+03	1.001e+03	1.000e+03	1.034e+03	7.624e+00	1.001e+03	1.004e+03
	$n/2$	1.004e+03	1.001e+03	1.000e+03	1.146e+03	2.036e+01	1.001e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	2.984e-01	1.000e+03	1.001e+03
c_{10}	1	4.850e+04	2.598e+04	2.769e+03	2.383e+05	5.352e+04	9.368e+03	7.249e+04
	$n/2$	9.619e+03	3.462e+03	1.341e+03	6.247e+04	1.285e+04	2.047e+03	1.458e+04
	n	1.408e+04	3.735e+03	1.363e+03	1.980e+05	3.865e+04	1.967e+03	6.177e+03
c_{11}	1	1.408e+03	1.406e+03	1.302e+03	1.497e+03	2.047e+01	1.404e+03	1.409e+03
	$n/2$	1.402e+03	1.402e+03	1.400e+03	1.405e+03	8.351e-01	1.402e+03	1.403e+03
	n	1.402e+03	1.402e+03	1.401e+03	1.404e+03	8.893e-01	1.401e+03	1.402e+03
c_{12}	1	1.324e+03	1.314e+03	1.304e+03	1.400e+03	2.744e+01	1.309e+03	1.320e+03
	$n/2$	1.331e+03	1.310e+03	1.304e+03	1.400e+03	4.053e+01	1.307e+03	1.378e+03
	n	1.346e+03	1.310e+03	1.301e+03	1.400e+03	4.715e+01	1.305e+03	1.400e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.474e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.089e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.139e-04	1.300e+03	1.300e+03
c_{14}	1	2.792e+03	2.355e+03	1.839e+03	9.655e+03	1.486e+03	2.179e+03	2.658e+03
	$n/2$	3.995e+03	1.982e+03	1.406e+03	1.378e+04	3.205e+03	1.837e+03	7.583e+03
	n	5.021e+03	1.946e+03	1.405e+03	1.404e+04	4.146e+03	1.503e+03	7.429e+03
c_{15}	1	1.620e+03	1.609e+03	1.601e+03	1.700e+03	2.662e+01	1.604e+03	1.623e+03
	$n/2$	1.604e+03	1.600e+03	1.600e+03	1.700e+03	1.955e+01	1.600e+03	1.600e+03
	n	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.400e+01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A28: Numerical results for MG-PMA with high-level two-point crossover and the tree depth (d) equal to 7 using the test set 3.

Benchmark Function	k	MBVF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	1.217e+07	6.457e+06	2.238e+05	5.725e+07	1.434e+07	4.817e+06	1.131e+07
	$n/2$	3.095e+06	2.695e+06	6.477e+05	1.076e+07	1.805e+06	1.693e+06	4.031e+06
	n	1.289e+06	1.090e+06	3.795e+05	3.641e+06	6.554e+05	8.285e+05	1.646e+06
c_2	1	2.664e+08	6.766e+07	3.570e+05	5.109e+09	7.425e+08	2.247e+07	2.508e+08
	$n/2$	1.231e+06	2.312e+04	1.612e+03	1.976e+07	3.639e+06	6.102e+03	3.576e+05
	n	8.528e+03	3.891e+03	2.134e+02	7.074e+04	1.136e+04	1.494e+03	1.173e+04
c_3	1	3.202e+02	3.202e+02	3.200e+02	3.206e+02	1.541e-01	3.201e+02	3.203e+02
	$n/2$	3.200e+02	3.200e+02	3.200e+02	3.202e+02	2.671e-02	3.200e+02	3.200e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.200e+02	2.883e-03	3.200e+02	3.200e+02
c_4	1	4.543e+02	4.542e+02	4.334e+02	4.744e+02	8.386e+00	4.499e+02	4.580e+02
	$n/2$	4.306e+02	4.318e+02	4.100e+02	4.488e+02	1.024e+01	4.220e+02	4.388e+02
	n	4.325e+02	4.318e+02	4.179e+02	4.527e+02	8.687e+00	4.259e+02	4.405e+02
c_5	1	1.607e+03	1.590e+03	1.113e+03	2.336e+03	2.861e+02	1.403e+03	1.759e+03
	$n/2$	1.090e+03	1.060e+03	7.445e+02	1.649e+03	2.208e+02	9.622e+02	1.204e+03
	n	9.936e+02	9.618e+02	6.469e+02	1.325e+03	1.345e+02	9.617e+02	1.079e+03
c_6	1	2.889e+05	1.139e+05	2.251e+03	2.017e+06	4.380e+05	4.104e+04	2.693e+05
	$n/2$	5.111e+05	3.351e+05	1.253e+04	2.475e+06	4.899e+05	2.071e+05	7.410e+05
	n	7.609e+04	2.678e+04	2.609e+03	1.069e+06	1.757e+05	9.939e+03	5.648e+04
c_7	1	7.076e+02	7.057e+02	7.022e+02	7.276e+02	5.540e+00	7.043e+02	7.078e+02
	$n/2$	7.022e+02	7.019e+02	7.011e+02	7.055e+02	9.022e-01	7.017e+02	7.022e+02
	n	7.008e+02	7.009e+02	7.002e+02	7.014e+02	3.118e-01	7.005e+02	7.011e+02
c_8	1	9.029e+04	6.579e+03	1.649e+03	3.885e+06	5.424e+05	3.065e+03	1.365e+04
	$n/2$	1.731e+05	4.576e+04	1.448e+03	1.978e+06	3.726e+05	1.831e+04	1.179e+05
	n	4.531e+05	2.732e+05	2.780e+03	4.024e+06	6.446e+05	1.518e+05	3.836e+05
c_9	1	1.004e+03	1.001e+03	1.000e+03	1.031e+03	6.985e+00	1.001e+03	1.003e+03
	$n/2$	1.001e+03	1.001e+03	1.000e+03	1.001e+03	1.721e-01	1.001e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	4.449e-01	1.000e+03	1.001e+03
c_{10}	1	6.365e+04	2.784e+04	3.462e+03	6.254e+05	9.877e+04	1.111e+04	7.006e+04
	$n/2$	6.749e+03	3.135e+03	1.409e+03	2.258e+04	6.597e+03	2.022e+03	8.021e+03
	n	5.742e+03	3.315e+03	1.352e+03	7.280e+04	1.029e+04	1.580e+03	5.639e+03
c_{11}	1	1.408e+03	1.406e+03	1.401e+03	1.429e+03	6.826e+00	1.403e+03	1.412e+03
	$n/2$	1.402e+03	1.402e+03	1.401e+03	1.404e+03	7.449e-01	1.402e+03	1.403e+03
	n	1.402e+03	1.402e+03	1.401e+03	1.405e+03	6.985e-01	1.401e+03	1.402e+03
c_{12}	1	1.329e+03	1.314e+03	1.305e+03	1.400e+03	2.941e+01	1.311e+03	1.337e+03
	$n/2$	1.322e+03	1.308e+03	1.302e+03	1.400e+03	3.408e+01	1.306e+03	1.311e+03
	n	1.329e+03	1.306e+03	1.302e+03	1.400e+03	4.177e+01	1.304e+03	1.377e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.497e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.422e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.533e-04	1.300e+03	1.300e+03
c_{14}	1	2.787e+03	2.263e+03	1.578e+03	1.386e+04	1.920e+03	2.083e+03	2.838e+03
	$n/2$	4.666e+03	1.957e+03	1.501e+03	1.354e+04	3.775e+03	1.873e+03	8.118e+03
	n	5.666e+03	7.377e+03	1.500e+03	1.347e+04	4.358e+03	1.504e+03	8.118e+03
c_{15}	1	1.631e+03	1.617e+03	1.601e+03	1.917e+03	4.895e+01	1.606e+03	1.638e+03
	$n/2$	1.605e+03	1.600e+03	1.600e+03	1.700e+03	1.984e+01	1.600e+03	1.600e+03
	n	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.400e+01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A29: Numerical results for MG-PMA with high-level uniform crossover and the tree depth (d) equal to 7 using the test set 3.

Benchmark Function	k	MBVF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1	7.060e+06	5.810e+06	1.082e+06	4.358e+07	6.391e+06	3.490e+06	9.274e+06
	$n/2$	2.916e+06	2.424e+06	7.768e+05	7.165e+06	1.449e+06	1.873e+06	4.146e+06
	n	1.366e+06	1.288e+06	5.729e+05	2.871e+06	5.387e+05	9.713e+05	1.788e+06
c_2	1	5.779e+08	2.075e+08	2.907e+04	6.552e+09	1.260e+09	3.003e+07	5.427e+08
	$n/2$	2.171e+05	1.740e+04	2.271e+02	4.463e+06	7.121e+05	4.238e+03	7.361e+04
	n	1.707e+04	8.092e+03	4.572e+02	3.096e+05	4.440e+04	4.598e+03	1.171e+04
c_3	1	3.202e+02	3.202e+02	3.200e+02	3.206e+02	1.855e-01	3.200e+02	3.203e+02
	$n/2$	3.200e+02	3.200e+02	3.200e+02	3.202e+02	2.736e-02	3.200e+02	3.200e+02
	n	3.200e+02	3.200e+02	3.200e+02	3.200e+02	9.427e-04	3.200e+02	3.200e+02
c_4	1	4.533e+02	4.525e+02	4.329e+02	4.811e+02	1.030e+01	4.460e+02	4.605e+02
	$n/2$	4.322e+02	4.339e+02	4.120e+02	4.459e+02	9.677e+00	4.236e+02	4.408e+02
	n	4.319e+02	4.320e+02	4.060e+02	4.517e+02	8.706e+00	4.281e+02	4.358e+02
c_5	1	1.628e+03	1.651e+03	1.077e+03	2.167e+03	2.643e+02	1.425e+03	1.843e+03
	$n/2$	1.062e+03	1.059e+03	6.259e+02	1.640e+03	2.107e+02	9.618e+02	1.179e+03
	n	1.016e+03	9.695e+02	8.433e+02	1.327e+03	1.304e+02	9.586e+02	1.101e+03
c_6	1	1.885e+05	1.162e+05	3.967e+03	1.108e+06	2.060e+05	4.931e+04	2.902e+05
	$n/2$	5.031e+05	3.309e+05	1.207e+04	2.237e+06	5.016e+05	1.540e+05	7.475e+05
	n	2.252e+05	2.694e+04	1.866e+03	9.321e+06	1.300e+06	1.185e+04	6.116e+04
c_7	1	7.069e+02	7.054e+02	7.014e+02	7.207e+02	4.519e+00	7.040e+02	7.075e+02
	$n/2$	7.020e+02	7.019e+02	7.011e+02	7.039e+02	5.281e-01	7.017e+02	7.021e+02
	n	7.009e+02	7.010e+02	7.002e+02	7.018e+02	3.679e-01	7.006e+02	7.011e+02
c_8	1	5.737e+04	4.825e+03	1.501e+03	1.617e+06	2.390e+05	2.523e+03	9.908e+03
	$n/2$	3.753e+05	1.217e+05	2.232e+03	4.379e+06	7.753e+05	6.322e+04	3.601e+05
	n	3.989e+05	2.953e+05	2.441e+03	2.772e+06	4.295e+05	1.437e+05	4.985e+05
c_9	1	1.006e+03	1.001e+03	1.000e+03	1.030e+03	8.611e+00	1.001e+03	1.008e+03
	$n/2$	1.001e+03	1.001e+03	1.000e+03	1.003e+03	6.894e-01	1.001e+03	1.001e+03
	n	1.001e+03	1.001e+03	1.000e+03	1.002e+03	3.252e-01	1.001e+03	1.001e+03
c_{10}	1	6.229e+04	4.250e+04	2.959e+03	2.983e+05	5.935e+04	1.426e+04	9.636e+04
	$n/2$	9.609e+03	3.861e+03	1.453e+03	1.093e+05	1.674e+04	2.409e+03	8.486e+03
	n	6.813e+03	3.045e+03	1.347e+03	6.401e+04	1.273e+04	1.537e+03	5.455e+03
c_{11}	1	1.407e+03	1.404e+03	1.401e+03	1.445e+03	7.050e+00	1.403e+03	1.408e+03
	$n/2$	1.402e+03	1.402e+03	1.401e+03	1.404e+03	7.766e-01	1.402e+03	1.403e+03
	n	1.402e+03	1.402e+03	1.400e+03	1.404e+03	8.990e-01	1.401e+03	1.402e+03
c_{12}	1	1.318e+03	1.312e+03	1.303e+03	1.400e+03	2.216e+01	1.308e+03	1.316e+03
	$n/2$	1.343e+03	1.312e+03	1.304e+03	1.400e+03	4.444e+01	1.308e+03	1.400e+03
	n	1.354e+03	1.400e+03	1.303e+03	1.400e+03	4.760e+01	1.306e+03	1.400e+03
c_{13}	1	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.371e-04	1.300e+03	1.300e+03
	$n/2$	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.989e-04	1.300e+03	1.300e+03
	n	1.300e+03	1.300e+03	1.300e+03	1.300e+03	2.181e-04	1.300e+03	1.300e+03
c_{14}	1	2.577e+03	2.430e+03	1.622e+03	9.162e+03	1.044e+03	2.180e+03	2.717e+03
	$n/2$	3.958e+03	1.947e+03	1.503e+03	1.291e+04	3.104e+03	1.866e+03	7.448e+03
	n	3.903e+03	1.882e+03	1.500e+03	1.261e+04	3.438e+03	1.501e+03	7.428e+03
c_{15}	1	1.633e+03	1.611e+03	1.601e+03	2.227e+03	9.279e+01	1.605e+03	1.626e+03
	$n/2$	1.604e+03	1.600e+03	1.600e+03	1.700e+03	1.954e+01	1.600e+03	1.601e+03
	n	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.400e+01	1.600e+03	1.600e+03

k - Number of functions incorporated in an MG-PMA individual; MBVF - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Tables A30 and A31 show the results of Aligned Friedman’s and Iman-Davenport’s tests, along with the Holm’s procedure based on the accuracy of MG-PMA with variations of k (Table A20), tournament size (Tables A23-A25) and high-level crossover operators (Tables A26-A29).

Table A30: Average rankings of Aligned Friedman’s and Iman-Davenport’s tests for comparison between MG-PMAs with different tournament sizes and crossover operators.

i	Algorithm	k	Tournament Size	Crossover Operator	Rank
23	MG-PMA	1	50	high-level	257.733
22	MG-PMA	1	25	high-level	252.733
21	MG-PMA	1	100	restricted high-level	243.833
20	MG-PMA	1	100	high-level two-point	238.367
19	MG-PMA	1	100	high-level	233.500
18	MG-PMA	1	100	high-level single-point	230.700
17	MG-PMA	1	100	high-level uniform	217.833
16	MG-PMA	$n/2$	100	high-level uniform	201.800
15	MG-PMA	$n/2$	100	high-level single-point	193.733
14	MG-PMA	1	2	high-level	191.800
13	MG-PMA	$n/2$	100	restricted high-level	186.567
12	MG-PMA	$n/2$	100	high-level two-point	170.267
11	MG-PMA	$n/2$	50	high-level	169.100
10	MG-PMA	$n/2$	100	high-level	163.100
9	MG-PMA	n	100	high-level single-point	162.400
8	MG-PMA	n	100	high-level uniform	161.700
7	MG-PMA	n	100	high-level	144.867
6	MG-PMA	n	100	high-level two-point	141.667
5	MG-PMA	n	50	high-level	140.767
4	MG-PMA	n	100	restricted high-level	140.400
3	MG-PMA	$n/2$	25	high-level	138.433
2	MG-PMA	n	25	high-level	133.200
1	MG-PMA	$n/2$	2	high-level	117.067
0	MG-PMA	n	2	high-level	100.433

Table A31: Holm’s procedure for pairwise comparison between MG-PMAs with different tournament sizes and crossover operators. The reference value R_0 corresponds to the rank of the best algorithm ($i = 0$); in this case MG-PMA with $k = n$, tournament size equals 2 and high-level crossover operator.

Test							p -value
Aligned Friedman							0.925
Algorithm	k	Tournament Size	Crossover Operator	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
MG-PMA	1	50	high-level	4.139	0.000	0.002	Yes
MG-PMA	1	25	high-level	4.008	0.000	0.002	Yes
MG-PMA	1	100	restricted high-level	3.774	0.000	0.002	Yes
MG-PMA	1	100	high-level two-point	3.630	0.000	0.003	Yes
MG-PMA	1	100	high-level	3.502	0.000	0.003	Yes
MG-PMA	1	100	high-level single-point	3.428	0.001	0.003	Yes
MG-PMA	1	100	high-level uniform	3.089	0.002	0.003	Yes
MG-PMA	$n/2$	100	high-level uniform	2.668	0.008	0.003	No
MG-PMA	$n/2$	100	high-level single-point	2.455	0.014	0.003	No
MG-PMA	1	2	high-level	2.404	0.016	0.004	No
MG-PMA	$n/2$	100	restricted high-level	2.267	0.023	0.004	No
MG-PMA	$n/2$	100	high-level two-point	1.838	0.066	0.004	No
MG-PMA	$n/2$	50	high-level	1.807	0.071	0.005	No
MG-PMA	$n/2$	100	high-level	1.649	0.099	0.005	No
MG-PMA	n	100	high-level single-point	1.631	0.103	0.006	No
MG-PMA	n	100	high-level uniform	1.612	0.107	0.006	No
MG-PMA	n	100	high-level	1.169	0.242	0.007	No
MG-PMA	n	100	high-level two-point	1.085	0.278	0.008	No
MG-PMA	n	50	high-level	1.061	0.289	0.010	No
MG-PMA	n	100	restricted high-level	1.052	0.293	0.013	No
MG-PMA	$n/2$	25	high-level	1.000	0.317	0.017	No
MG-PMA	n	25	high-level	0.862	0.389	0.025	No
MG-PMA	$n/2$	2	high-level	0.438	0.662	0.050	No

A6

Numerical Results for Comparison with Known Methods

Table A32: Numerical results for MG-PMA with number of functions (k) equal to n ($n = 10$), tournament size equal to 2, high-level crossover, the tree depth (d) equal to 7 and local minimization using the test set 3.

Benchmark Function	MVBF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1.000e+02	1.000e+02	1.000e+02	1.000e+02	3.855e-03	1.000e+02	1.000e+02
c_2	2.004e+02	2.000e+02	2.000e+02	2.198e+02	2.776e+00	2.000e+02	2.000e+02
c_3	3.200e+02	3.200e+02	3.200e+02	3.200e+02	4.806e-05	3.200e+02	3.200e+02
c_4	4.303e+02	4.298e+02	4.129e+02	4.448e+02	7.615e+00	4.249e+02	4.376e+02
c_5	9.606e+02	9.617e+02	6.369e+02	1.193e+03	1.116e+02	9.584e+02	1.049e+03
c_6	8.731e+02	8.753e+02	6.119e+02	1.345e+03	2.058e+02	6.644e+02	9.804e+02
c_7	7.015e+02	7.015e+02	7.006e+02	7.033e+02	4.963e-01	7.012e+02	7.017e+02
c_8	9.007e+02	8.199e+02	8.067e+02	1.275e+03	1.244e+02	8.179e+02	9.914e+02
c_9	1.001e+03	1.001e+03	1.000e+03	1.001e+03	1.649e-01	1.000e+03	1.001e+03
c_{10}	1.446e+03	1.418e+03	1.293e+03	1.679e+03	1.049e+02	1.365e+03	1.523e+03
c_{11}	1.369e+03	1.402e+03	1.108e+03	1.406e+03	9.198e+01	1.400e+03	1.403e+03
c_{12}	1.303e+03	1.303e+03	1.301e+03	1.308e+03	1.493e+00	1.302e+03	1.304e+03
c_{13}	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.880e-04	1.300e+03	1.300e+03
c_{14}	2.400e+03	1.500e+03	1.500e+03	1.261e+04	2.291e+03	1.500e+03	1.716e+03
c_{15}	1.602e+03	1.600e+03	1.600e+03	1.700e+03	1.401e+01	1.600e+03	1.600e+03

MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A33: Numerical results for PSO with local minimization using the test set 3.

Benchmark Function	MVBF	Median	Min	Max	Std	1st Quartile	3rd Quartile
c_1	1.328e+02	1.000e+02	1.000e+02	1.107e+03	1.530e+02	1.000e+02	1.000e+02
c_2	2.017e+02	2.000e+02	2.000e+02	2.698e+02	9.928e+00	2.000e+02	2.001e+02
c_3	3.196e+02	3.200e+02	3.000e+02	3.200e+02	2.800e+00	3.200e+02	3.200e+02
c_4	4.050e+02	4.050e+02	4.020e+02	4.090e+02	1.554e+00	4.040e+02	4.060e+02
c_5	7.640e+02	7.540e+02	5.152e+02	1.339e+03	1.909e+02	6.258e+02	8.783e+02
c_6	8.103e+02	7.448e+02	6.002e+02	2.210e+03	2.379e+02	7.232e+02	8.576e+02
c_7	7.010e+02	7.011e+02	7.001e+02	7.028e+02	6.092e-01	7.010e+02	7.011e+02
c_8	9.130e+02	9.251e+02	8.010e+02	1.346e+03	1.088e+02	8.192e+02	9.437e+02
c_9	1.000e+03	1.000e+03	1.000e+03	1.000e+03	5.151e-02	1.000e+03	1.000e+03
c_{10}	1.444e+03	1.392e+03	1.267e+03	2.097e+03	1.679e+02	1.331e+03	1.486e+03
c_{11}	1.379e+03	1.400e+03	1.103e+03	1.500e+03	8.215e+01	1.400e+03	1.401e+03
c_{12}	1.301e+03	1.301e+03	1.301e+03	1.303e+03	3.943e-01	1.301e+03	1.302e+03
c_{13}	1.300e+03	1.300e+03	1.300e+03	1.300e+03	1.404e-02	1.300e+03	1.300e+03
c_{14}	2.793e+03	1.716e+03	1.500e+03	1.025e+04	1.940e+03	1.716e+03	2.910e+03
c_{15}	1.600e+03	1.600e+03	1.600e+03	1.600e+03	3.232e-13	1.600e+03	1.600e+03

MBFV - Mean of the best fitness values; Median - Median of the best fitness values; Min - Minimum of the best fitness values; Max - Maximum of the best fitness values; Std - Standard deviation of the best fitness values; 1st Quartile - First quartile of the best fitness values; 3rd Quartile - Third quartile of the best fitness values.

Table A34 shows the results of Aligned Friedman's and Iman-Davenport's tests, along with the Holm's procedure based on the accuracy PSO, SPS-L-SHADE-EIG, DEsPA, LSHADE-ND, MVMO-SH and MG-PMA (Table 4.7).

Table A34: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between MG-PMA, PSO, SPS-L-SHADE-EIG, DEsPA, LSHADE-ND and MVMO-SH. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case SPS-L-SHADE-EIG.

i	Algorithm	Rank
5	PSO	63.433
4	MG-PMA	63.200
3	DEsPA	45.933
2	LSHADE-ND	37.933
1	MVMO-SH	31.800
0	SPS-L-SHADE-EIG	30.700

Test	p -value
Aligned Friedman	0.032

Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
PSO	3.431	0.000	0.010	Yes
MG-PMA	3.407	0.001	0.0125	Yes
DEsPA	1.597	0.110	0.0167	No
LSHADE-ND	0.758	0.448	0.025	No
MVMO-SH	0.115	0.908	0.050	No

B CEC'2015 Benchmark Functions

B1 Introduction

This appendix presents the properties of the CEC'2015 benchmark functions (c_1 - c_{15}) and their 3-D maps for the 2-D functions available in the problem definition for the CEC'2015 competition (Liang, 2014). More details about the benchmark functions can also be found in the rules of the competition (Liang, 2014).

B1.1 Unimodal Functions

Figure B1 presents the 3-D map for 2-D Rotated High Conditioned Elliptic Function (c_1).

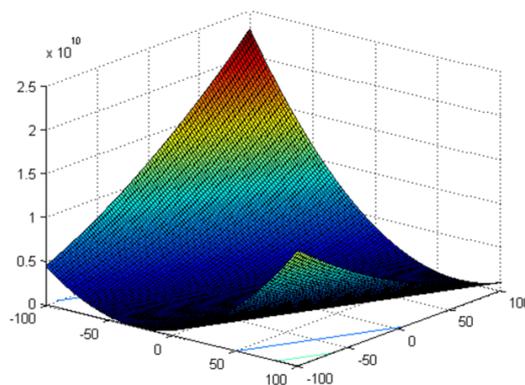


Figure B1: 3-D map for 2-D Rotated High Conditioned Elliptic Function (c_1) (Liang, 2014).

Properties of c_1 :

- Unimodal;
- Non-separable;
- Quadratic ill-conditioned.

Figure B2 presents the 3-D map for 2-D the Rotated Cigar Function (c_2).

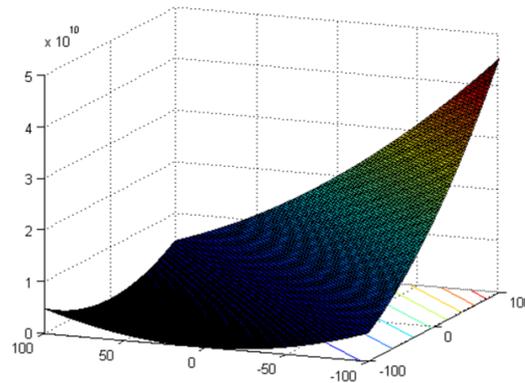


Figure B2: 3-D map for 2-D Rotated Cigar Function (c_2) (Liang, 2014).

Properties of c_2 :

- Unimodal;
- Non-separable;
- Smooth but narrow ridge.

B1.2 Multimodal Functions

Figure B3 presents the 3-D map for 2-D the Shifted and Rotated Ackley's Function (c_3).

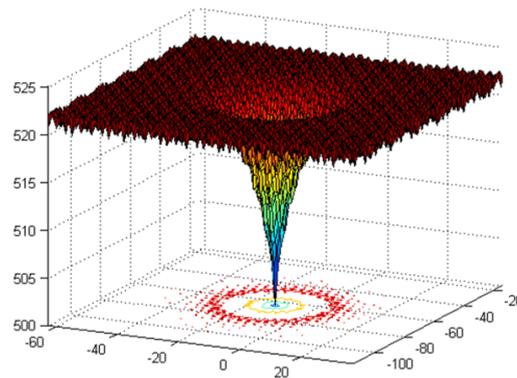


Figure B3: 3-D map for 2-D Shifted and Rotated Ackley's Function (c_3) (Liang, 2014).

Properties of c_3 :

- Multimodal;
- Non-separable.

Figure B4 presents the 3-D map for 2-D the Shifted and Rotated Rastrigin's Function (c_4).

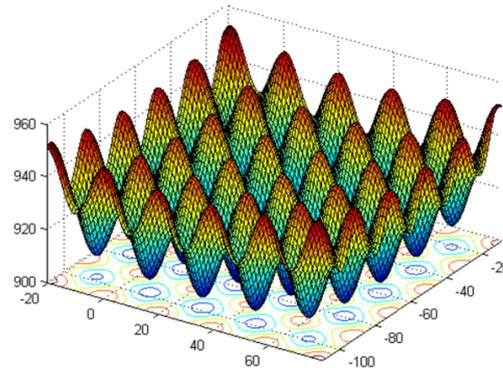


Figure B4: 3-D map for 2-D Shifted and Rotated Rastrigin's Function (c_4) (Liang, 2014).

Properties of c_4 :

- Multimodal;
- Non-separable;
- Local optima's number is huge.

Figure B5.a presents the 3-D map for 2-D the Shifted and Rotated Schwefel's Function (c_5) and Figure B5.b its contour map 2-D function.

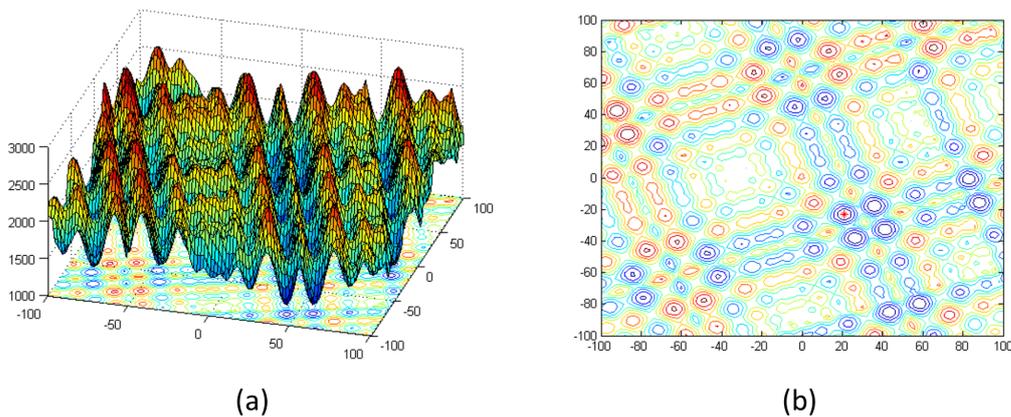


Figure B5: Shifted and Rotated Schwefel's Function (c_5): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).

Properties of c_5 :

- Multimodal;
- Non-separable;
- Local optima's number is huge and second better local optimum is far from the global optimum.

B1.3 Hybrid Functions

The hybrid functions (c_6 - c_8) are divided into some subcomponents and then different basic functions (Liang, 2014) are used for different subcomponents. The properties of hybrid functions are:

- Multimodal or unimodal, depending on the basic function;
- Non-separable subcomponents;
- Different properties for different variables subcomponents.

B1.4 Composition Functions

Figure B6.a presents the 3-D map for 2-D the Composition Function I (c_9) and Figure B6.b its contour map 2-D function.

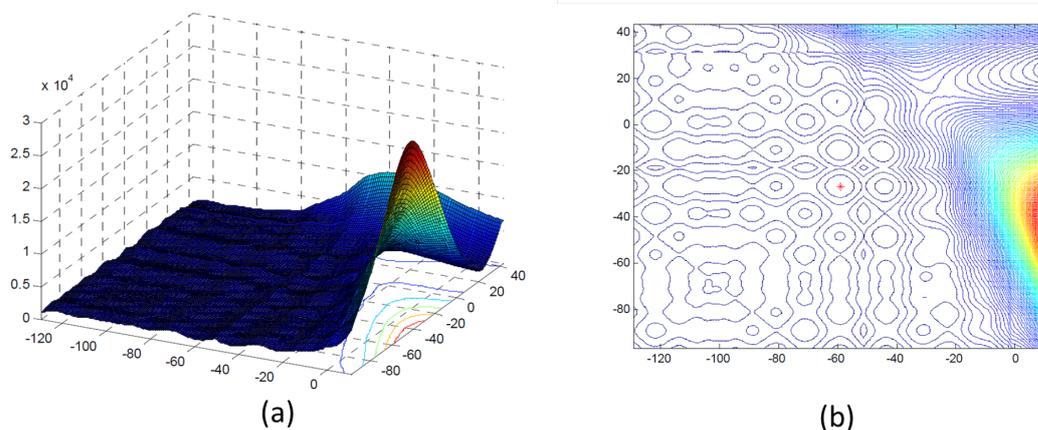


Figure B6: Composition Function I (c_9): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).

Properties of c_9 :

- Multimodal;
- Non-separable;

- Different properties around different local optima;
- The basic function of which the global optimum belongs to is fixed. The sequence of the other basic functions can be randomly generated.

Properties of c_{10} :

- Multimodal;
- Non-separable;
- Asymmetrical;
- Different properties around different local optima;
- Different properties for different variables subcomponents;
- The sequence of the basic functions can be randomly generated.

Figure B7.a presents the 3-D map for 2-D the Composition Function 3 (c_{11}) and Figure B7.b its contour map 2-D function.

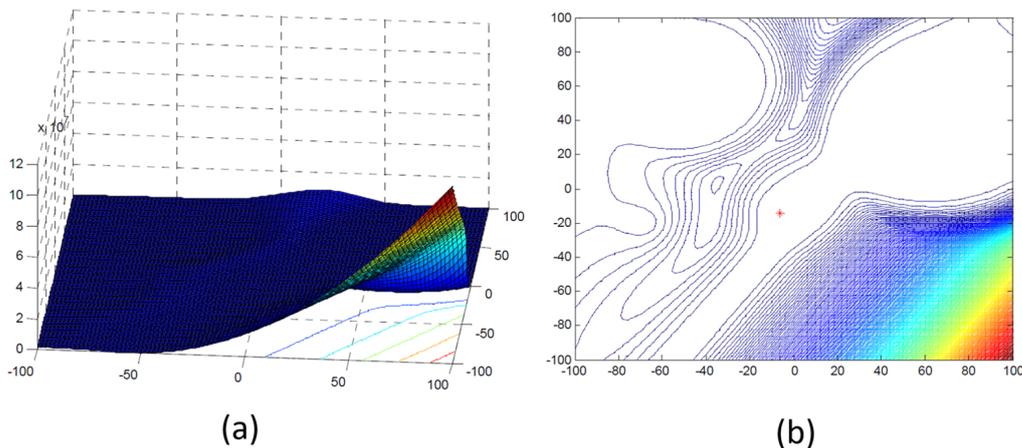


Figure B7: Composition Function 3 (c_{11}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).

Properties of c_{11} :

- Multimodal;
- Non-separable;
- Asymmetrical;
- Different properties around different local optima;
- The basic function of which the global optimum belongs to is fixed. The sequence of the other basic functions can be randomly generated.

Figure B8.a presents the 3-D map for 2-D the Composition Function 4 (c_{12}) and Figure B8.b its contour map 2-D function.

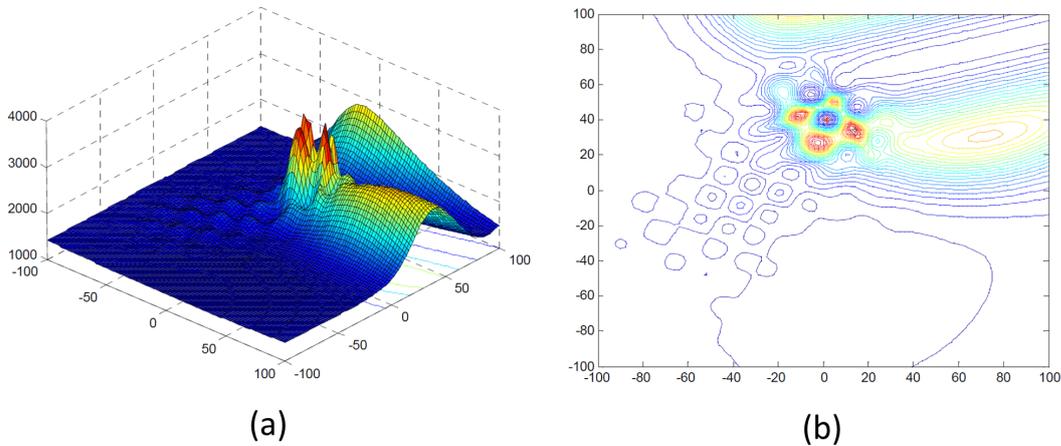


Figure B8: Composition Function 4 (c_{12}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).

Properties of c_{12} :

- Multimodal;
- Non-separable;
- Asymmetrical;
- Different properties around different local optima;
- Different properties for different variables subcomponents;
- The sequence of the basic functions can be randomly generated.

Properties of c_{13} :

- Multimodal;
- Non-separable;
- Asymmetrical;
- Different properties around different local optima;
- The sequence of the basic functions can be randomly generated.

Figure B9.a presents the 3-D map for 2-D the Composition Function 6 (c_{14}) and Figure B9.b its contour map 2-D function.

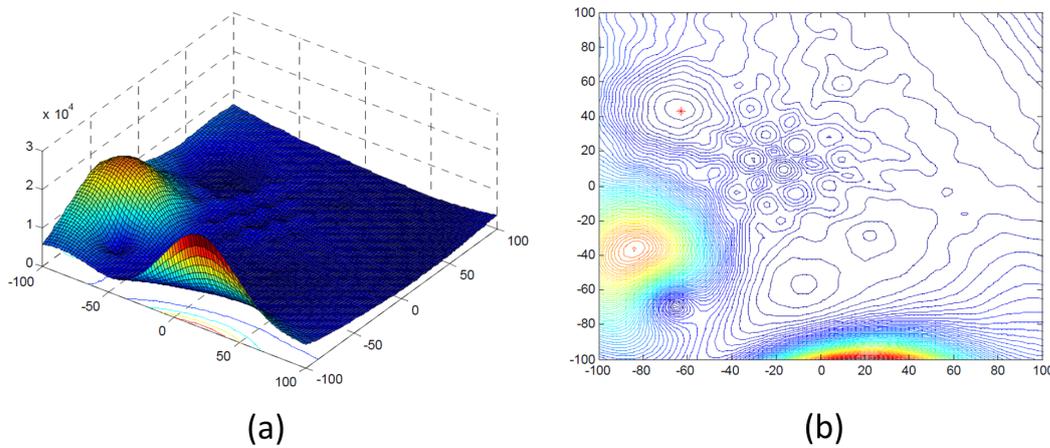


Figure B9: Composition Function 6 (c_{14}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).

Properties of c_{14} :

- Multimodal;
- Non-separable;
- Asymmetrical;
- Different properties around different local optima;
- Different properties for different variables subcomponents;
- The sequence of the basic functions can be randomly generated.

Figure B10.a presents the 3-D map for 2-D the Composition Function 7 (c_{15}) and Figure B10.b its contour map 2-D function.

Properties of c_{15} :

- Multimodal;
- Non-separable;
- Asymmetrical;
- Different properties around different local optima;
- The sequence of the basic functions can be randomly generated.

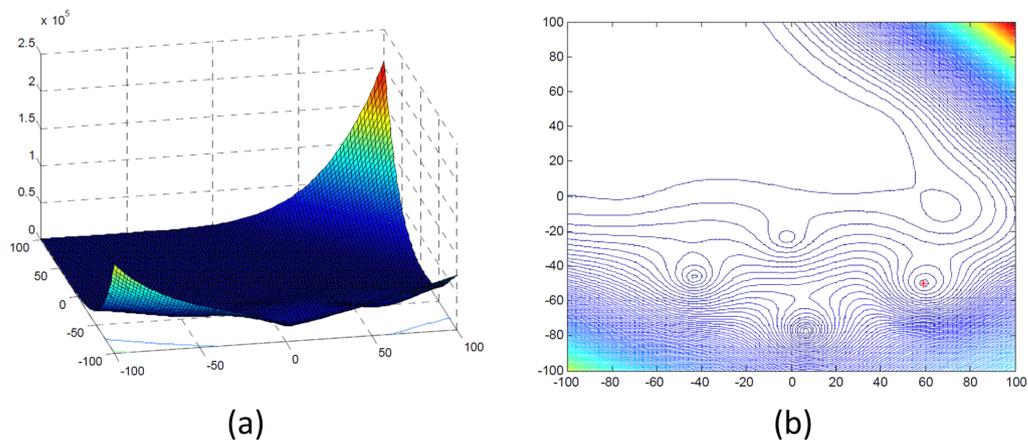


Figure B10: Composition Function 7 (c_{15}): (a) 3-D map for 2-D function; and (b) Contour map for 2-D function (Liang, 2014).

C

Analyses of Niching MG-PMA – Numerical Results

C1

Introduction

This appendix presents all the numerical results obtained in the experiments of the Chapter 5 of this thesis.

C2

Numerical Results for High-Level Crossover Operator Analysis

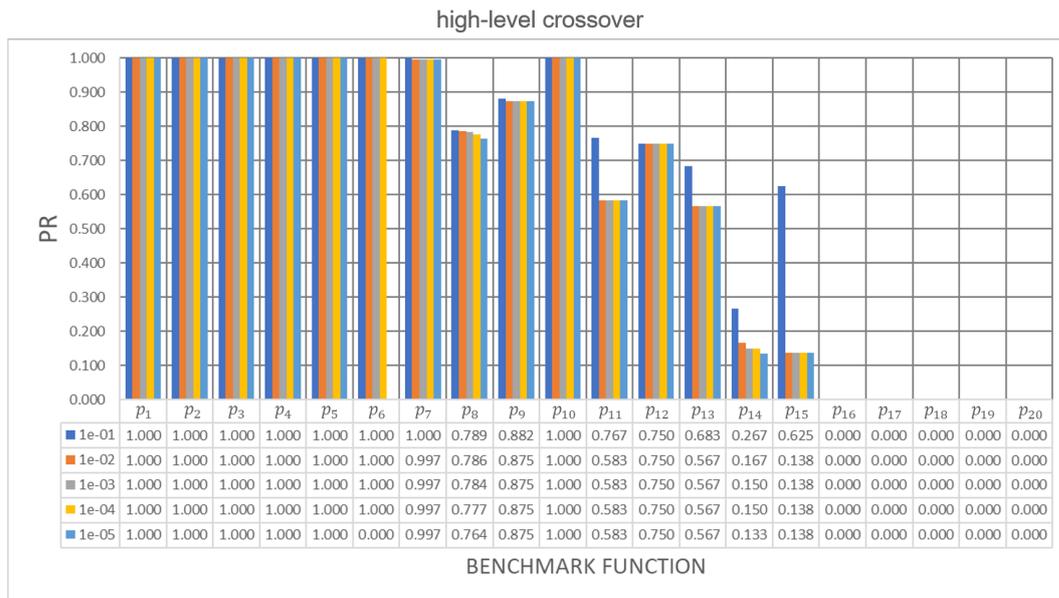


Figure C1: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

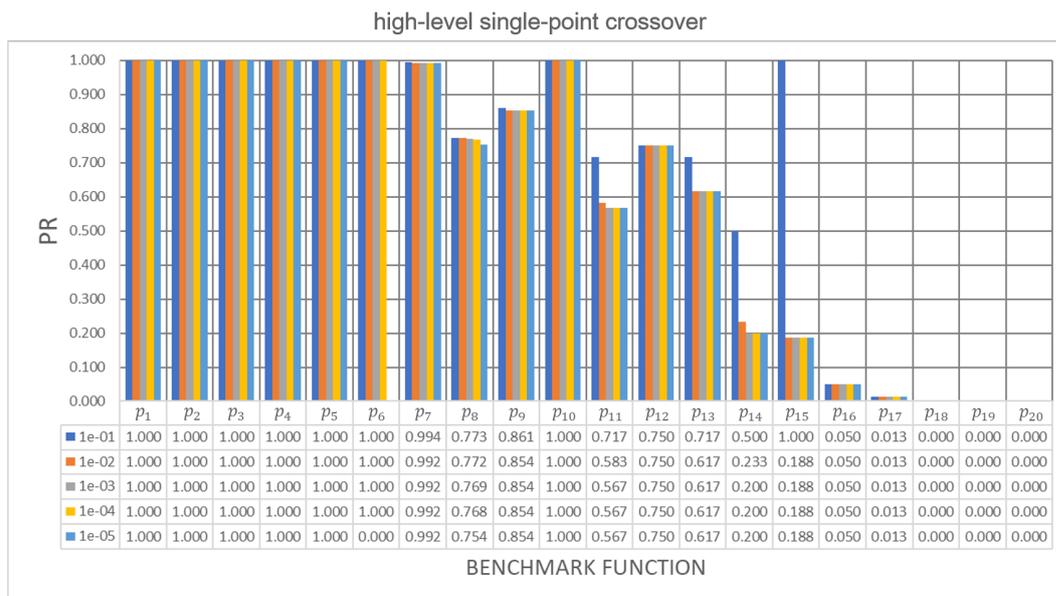


Figure C2: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

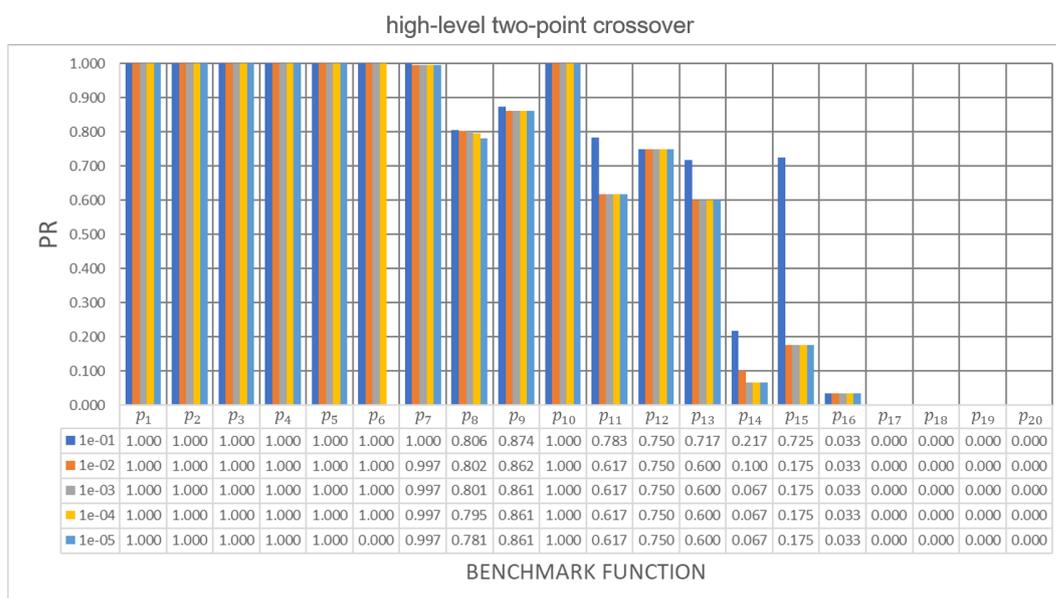


Figure C3: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level two-point crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

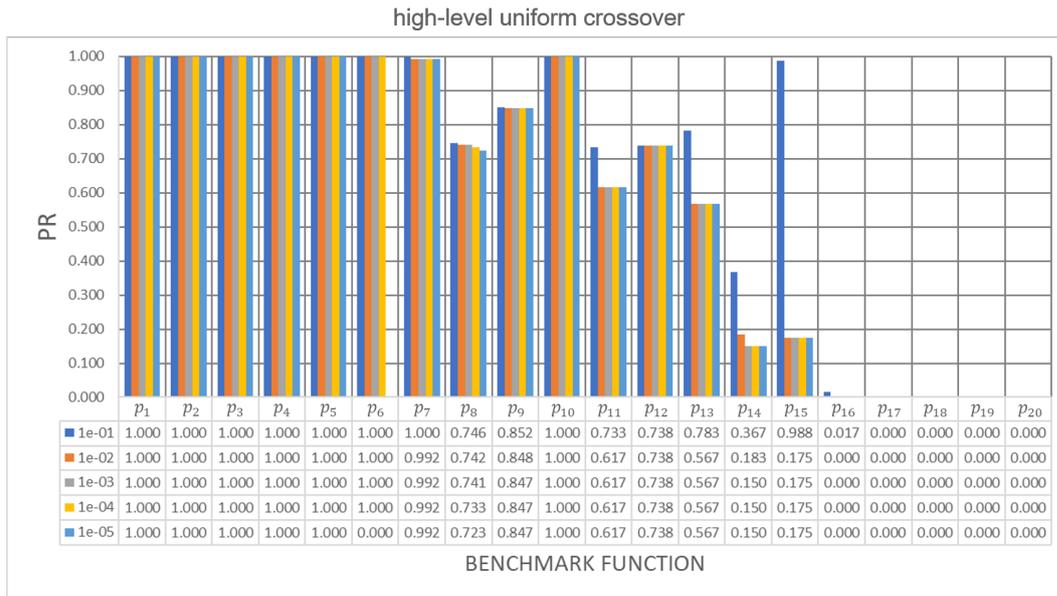


Figure C4: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level uniform crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

Tables C1-C5 the Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with five level of accuracy: $\{1e-1, 1e-2, 1e-3, 1e-4, 1e-5\}$.

Table C1: Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-1$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover.

i	Genetic Operator	Rank
3	high-level uniform crossover	42.750
2	high-level crossover	41.900
1	high-level single-point crossover	39.075
0	high-level two-point crossover	38.275

Test	p -value
Aligned Friedman	0.017

Genetic Operator	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
high-level uniform crossover	0.609	0.543	0.017	No
high-level crossover	0.493	0.622	0.025	No
high-level single-point crossover	0.109	0.913	0.050	No

Table C2: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-2$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover.

i	Genetic Operator	Rank
3	high-level uniform crossover	47.225
2	high-level crossover	45.300
1	high-level single-point crossover	36.350
0	high-level two-point crossover	33.125

Test	p -value
Aligned Friedman	0.017

Genetic Operator	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
high-level uniform crossover	1.919	0.055	0.017	No
high-level crossover	1.657	0.098	0.025	No
high-level single-point crossover	0.439	0.661	0.050	No

Table C3: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-3$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover.

i	Genetic Operator	Rank
3	high-level uniform crossover	47.900
2	high-level crossover	43.050
1	high-level single-point crossover	37.275
0	high-level two-point crossover	33.775

Test	p -value
Aligned Friedman	0.017

Genetic Operator	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
high-level uniform crossover	1.922	0.055	0.017	No
high-level crossover	1.262	0.207	0.025	No
high-level single-point crossover	0.476	0.634	0.050	No

Table C4: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-4$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover.

i	Genetic Operator	Rank
3	high-level uniform crossover	47.950
2	high-level crossover	43.200
1	high-level single-point crossover	37.025
0	high-level two-point crossover	33.825

Test	p -value
Aligned Friedman	0.017

Genetic Operator	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
high-level uniform crossover	1.922	0.055	0.017	No
high-level crossover	1.276	0.202	0.025	No
high-level single-point crossover	0.436	0.663	0.050	No

Table C5: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between high-level crossover operators for Niching MG-PMA with $1e-5$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case high-level two-point crossover.

i	Genetic Operator	Rank
3	high-level uniform crossover	47.125
2	high-level crossover	45.275
1	high-level single-point crossover	36.425
0	high-level two-point crossover	33.175

Test	p -value
Aligned Friedman	0.017

Genetic Operator	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
high-level uniform crossover	1.898	0.058	0.017	No
high-level crossover	1.647	0.100	0.025	No
high-level single-point crossover	0.442	0.658	0.050	No

C3

Numerical Results for Local Optimization Frequency Analysis

This section presents the numerical results for local optimization frequency analysis. Figures C5-C13 present the bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator or high-level two-point crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. Each figure uses a different local optimization frequency and the values used are: 10, 50, 100, 150 and 200.

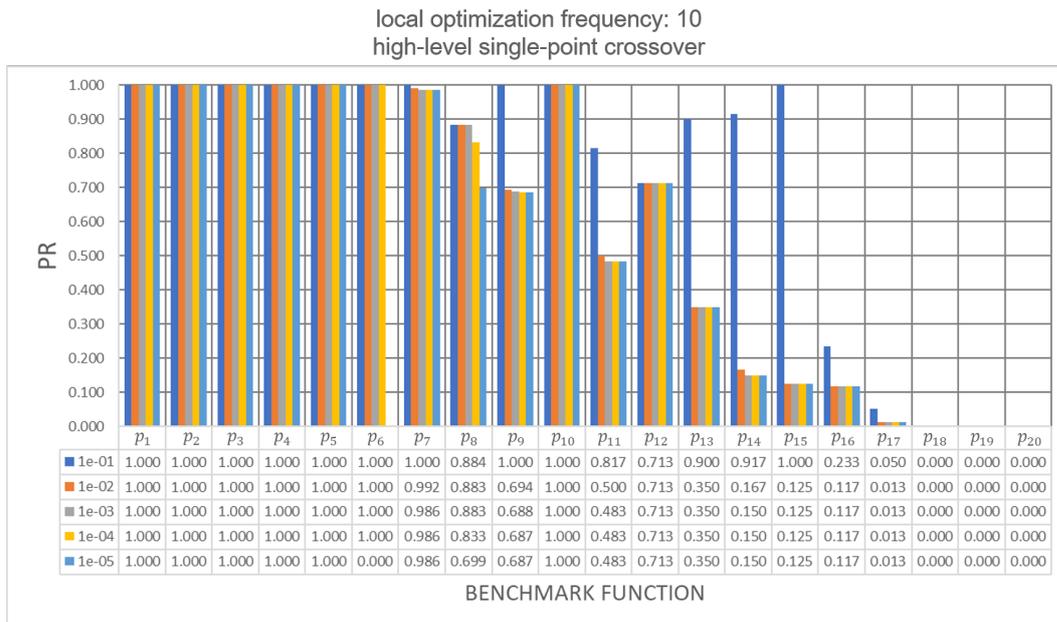


Figure C5: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 10, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

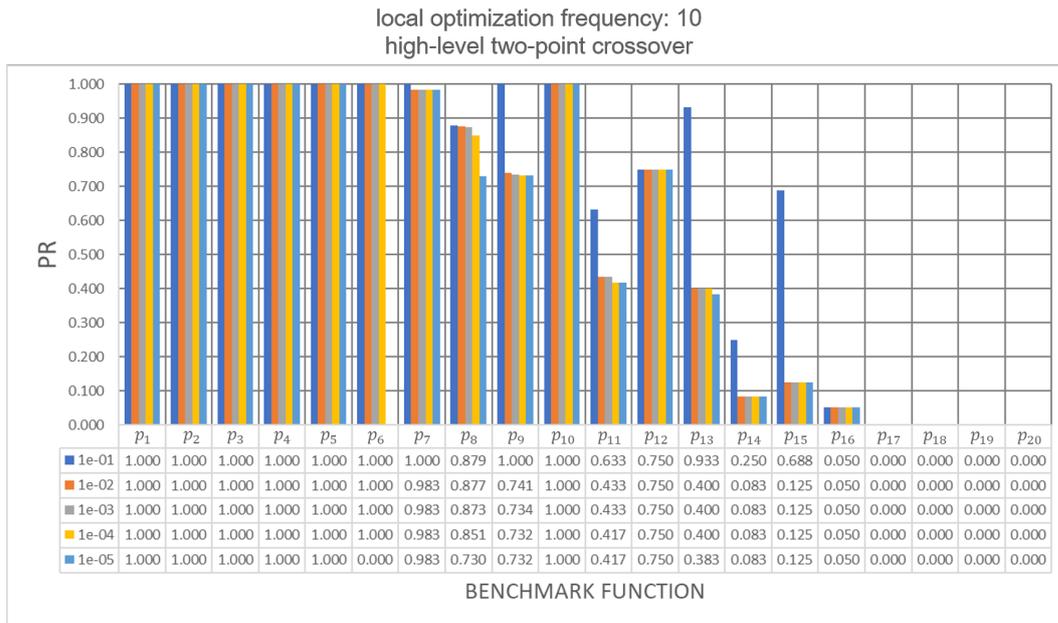


Figure C6: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 10, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

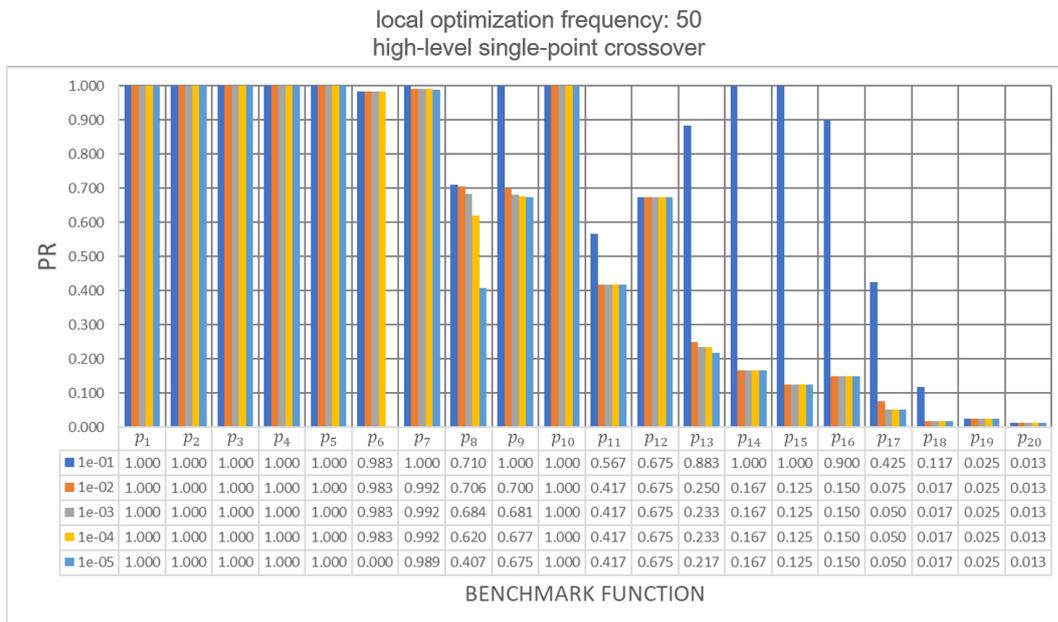


Figure C7: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 50, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

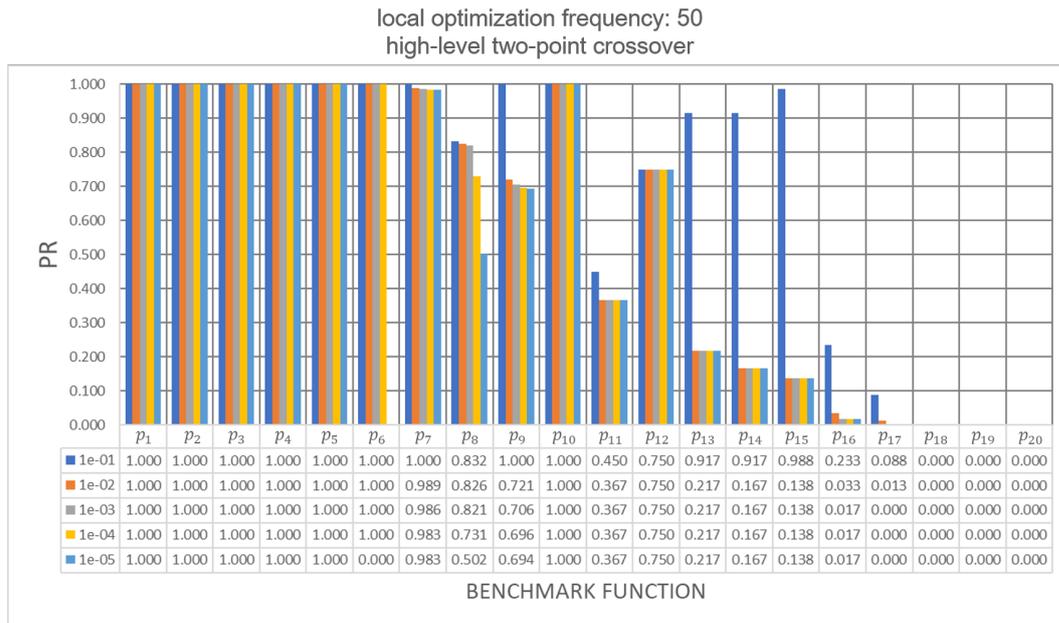


Figure C8: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 50, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

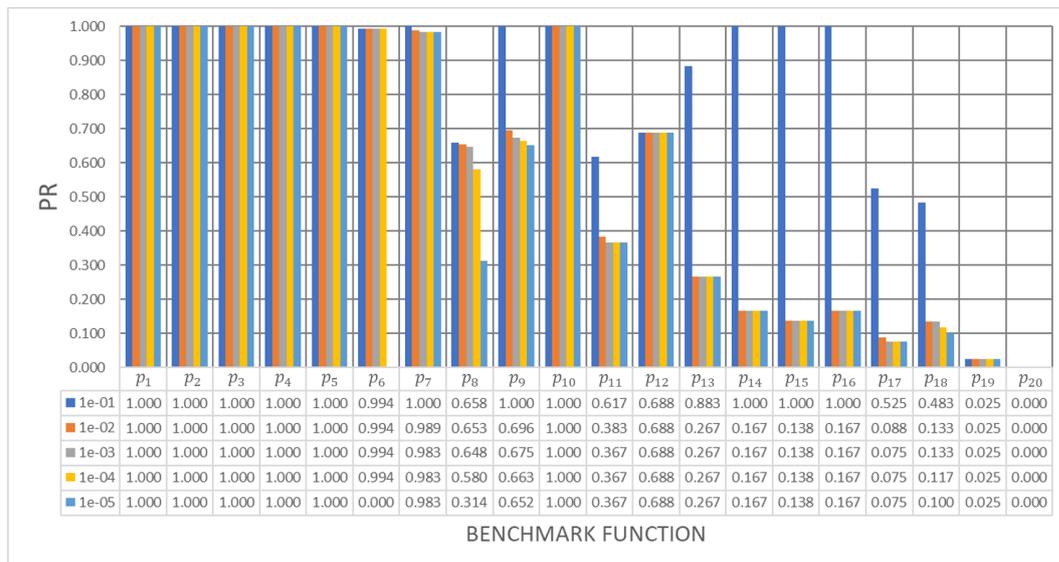


Figure C9: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 100, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

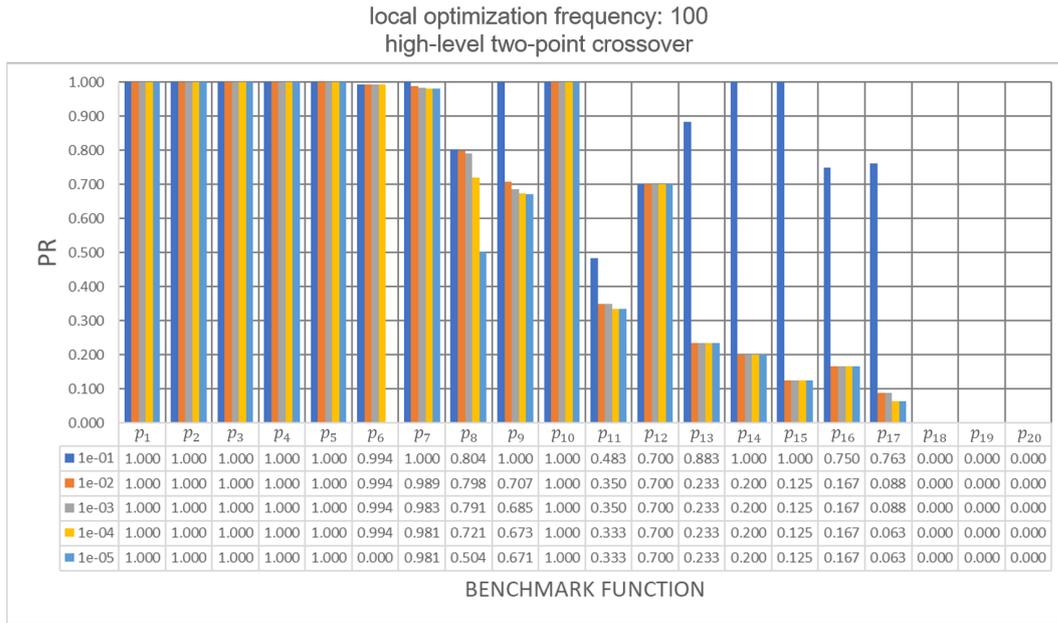


Figure C10: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 100, for 20 benchmark multimodal functions (p_1-p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

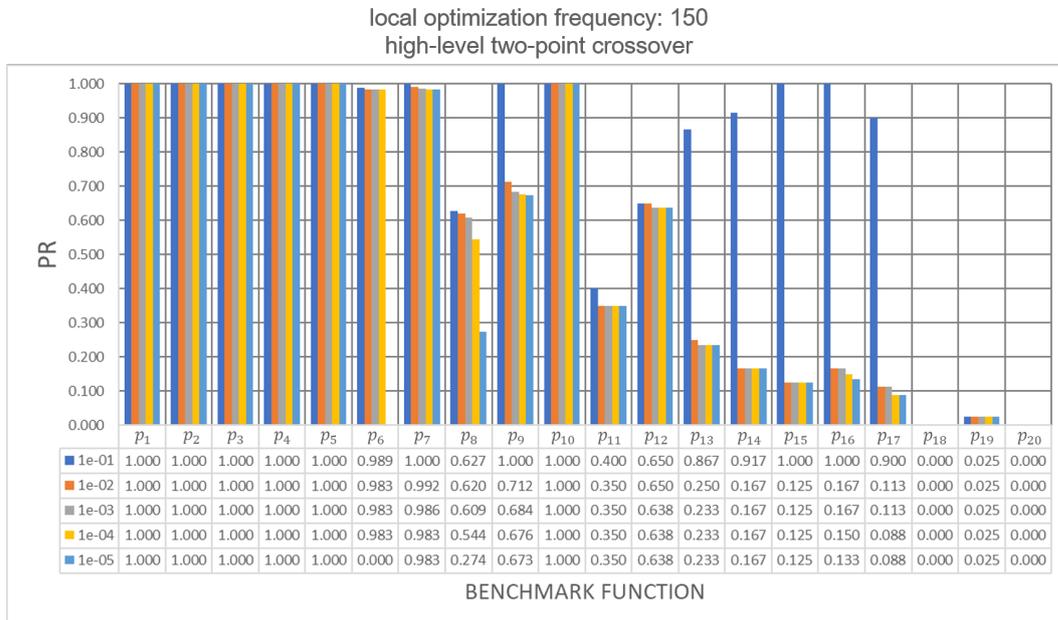


Figure C11: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 150, for 20 benchmark multimodal functions (p_1-p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

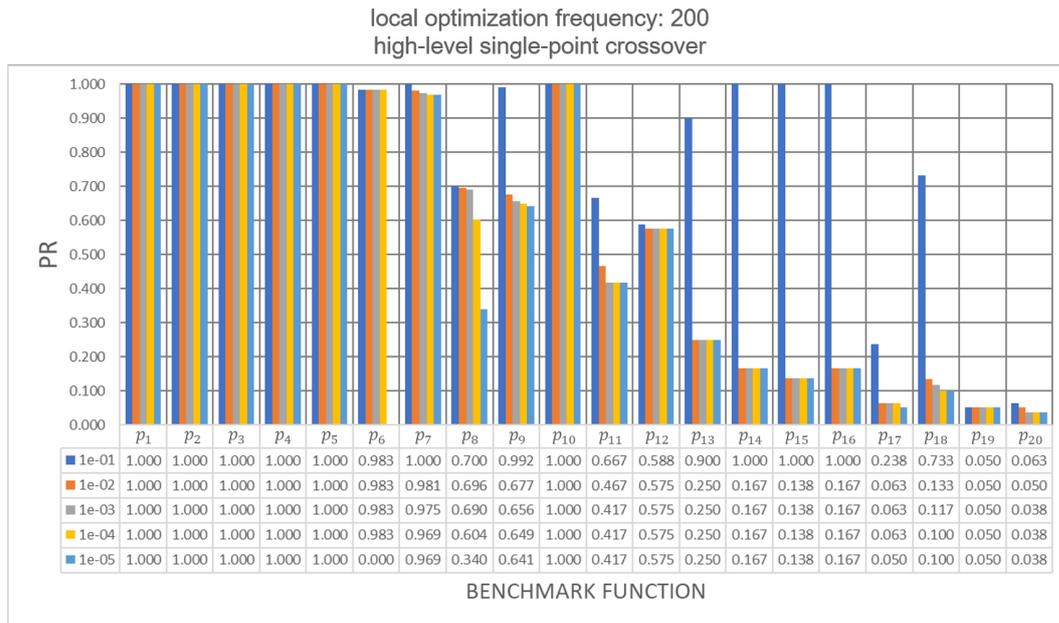


Figure C12: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and local optimization frequency equal to 200, for 20 benchmark multimodal functions (p_1-p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

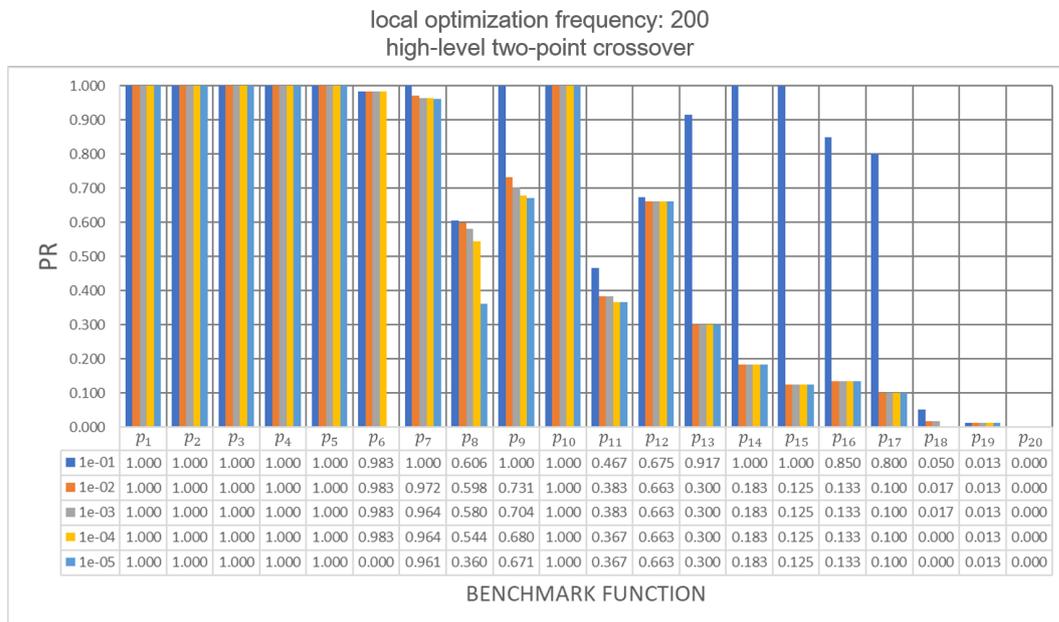


Figure C13: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level two-point crossover operator, and local optimization frequency equal to 200, for 20 benchmark multimodal functions (p_1-p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

Tables C6-C10 present the Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

Table C6: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-1$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L150 sp.

i	Niching MG-PMA	Rank
11	L1 tp	152.400
10	L1 sp	148.650
9	L10 tp	137.150
8	L50 tp	126.100
7	L200 tp	123.550
6	L150 tp	122.050
5	L10 sp	117.625
4	L50 sp	114.625
3	L100 tp	108.475
2	L200 sp	101.825
1	L100 sp	98.075
0	L150 sp	95.475

Test	p -value
Aligned Friedman	0.005

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
L1 tp	2.593	0.010	0.005	No
L1 sp	2.422	0.154	0.005	No
L10 tp	1.898	0.058	0.006	No
L50 tp	1.395	0.163	0.006	No
L200 tp	1.279	0.201	0.007	No
L150 tp	1.211	0.226	0.008	No
L10 sp	1.009	0.313	0.010	No
L50 sp	0.872	0.383	0.013	No
L100 tp	0.592	0.554	0.017	No
L200 sp	0.289	0.772	0.025	No
L100 sp	0.118	0.906	0.050	No

Table C7: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-2$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp.

i	Niching MG-PMA	Rank
11	L200 tp	143.050
10	L150 tp	136.775
9	L50 tp	131.725
8	L50 sp	129.050
7	L10 tp	127.850
6	L100 tp	120.400
5	L150 sp	117.025
4	L100 sp	115.625
3	L200 sp	112.950
2	L10 sp	110.650
1	L1 tp	105.200
0	L1 sp	95.700

Test	p -value
Aligned Friedman	0.005

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
L200 tp	2.157	0.031	0.005	No
L150 tp	1.871	0.061	0.005	No
L50 tp	1.641	0.101	0.006	No
L50 sp	1.519	0.129	0.006	No
L10 tp	1.464	0.143	0.007	No
L100 tp	1.125	0.261	0.008	No
L150 sp	0.971	0.331	0.010	No
L100 sp	0.908	0.364	0.013	No
L200 sp	0.786	0.432	0.017	No
L10 sp	0.681	0.496	0.025	No
L1 tp	0.433	0.665	0.050	No

Table C8: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-3$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp.

i	Niching MG-PMA	Rank
11	L200 tp	142.550
10	L150 tp	135.725
9	L50 tp	129.600
8	L50 sp	128.850
7	L100 tp	122.650
6	L10 tp	121.425
5	L200 sp	118.300
4	L100 sp	117.800
3	L150 sp	115.800
2	L10 sp	113.850
1	L1 tp	104.200
0	L1 sp	95.250

Test	p -value
Aligned Friedman	0.005

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
L200 tp	2.155	0.031	0.005	No
L150 tp	1.844	0.065	0.005	No
L50 tp	1.565	0.118	0.006	No
L50 sp	1.530	0.126	0.006	No
L100 tp	1.248	0.212	0.007	No
L10 tp	1.192	0.233	0.008	No
L200 sp	1.050	0.294	0.010	No
L100 sp	1.027	0.304	0.013	No
L150 sp	0.936	0.349	0.017	No
L10 sp	0.847	0.397	0.025	No
L1 tp	0.408	0.684	0.050	No

Table C9: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-4$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp.

i	Niching MG-PMA	Rank
11	L200 tp	143.650
10	L150 tp	137.050
9	L50 tp	130.525
8	L50 sp	125.825
7	L100 tp	123.950
6	L150 sp	123.175
5	L10 tp	121.125
4	L200 sp	118.175
3	L100 sp	113.750
2	L10 sp	112.675
1	L1 tp	103.275
0	L1 sp	92.825

Test	p -value
Aligned Friedman	0.005

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
L200 tp	2.315	0.021	0.005	No
L150 tp	2.014	0.044	0.005	No
L50 tp	1.717	0.086	0.006	No
L50 sp	1.503	0.133	0.006	No
L100 tp	1.418	0.156	0.007	No
L150 sp	1.382	0.169	0.008	No
L10 tp	1.289	0.197	0.010	No
L200 sp	1.155	0.248	0.013	No
L100 sp	0.953	0.341	0.017	No
L10 sp	0.904	0.366	0.025	No
L1 tp	0.476	0.634	0.050	No

Table C10: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different local optimization frequency values (L1, L10, L50, L100, L150 and L200) and high-level crossover operators (sp – single-point, tp – two-point) for Niching MG-PMA with $1e-5$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration L1 sp.

i	Niching MG-PMA	Rank
11	L200 tp	141.075
10	L150 tp	133.375
9	L50 tp	132.925
8	L100 tp	126.850
7	L10 tp	123.100
6	L50 sp	122.500
5	L150 sp	121.625
4	L200 sp	115.525
3	L100 sp	114.900
2	L10 sp	114.100
1	L1 tp	105.300
0	L1 sp	94.725

Test	p -value
Aligned Friedman	0.005

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
L200 tp	2.112	0.035	0.005	No
L150 tp	1.761	0.078	0.005	No
L50 tp	1.740	0.082	0.006	No
L100 tp	1.463	0.143	0.006	No
L10 tp	1.293	0.196	0.007	No
L50 sp	1.265	0.206	0.008	No
L150 sp	1.225	0.221	0.010	No
L200 sp	0.947	0.343	0.013	No
L100 sp	0.919	0.358	0.017	No
L10 sp	0.883	0.378	0.025	No
L1 tp	0.482	0.630	0.050	No

C4

Numerical Results for Feedback Frequency Analysis

This section presents the numerical results for feedback frequency analysis. Figures C14-C18 present the bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$. Each figure uses a different feedback frequency and the values used are: 0, 10, 50, 100, 150 and 200.

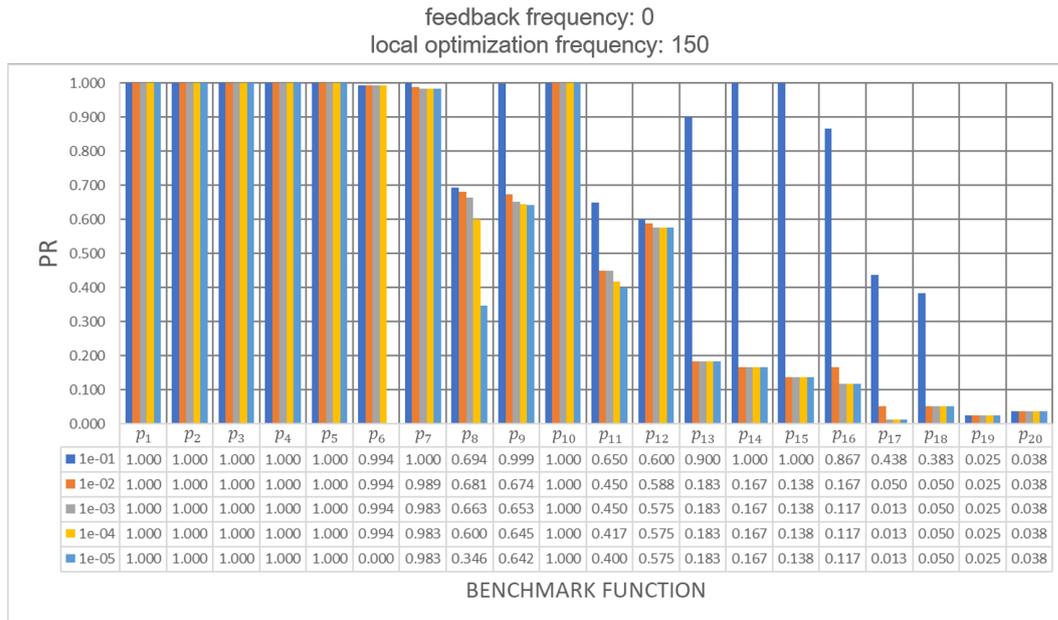


Figure C14: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 0, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

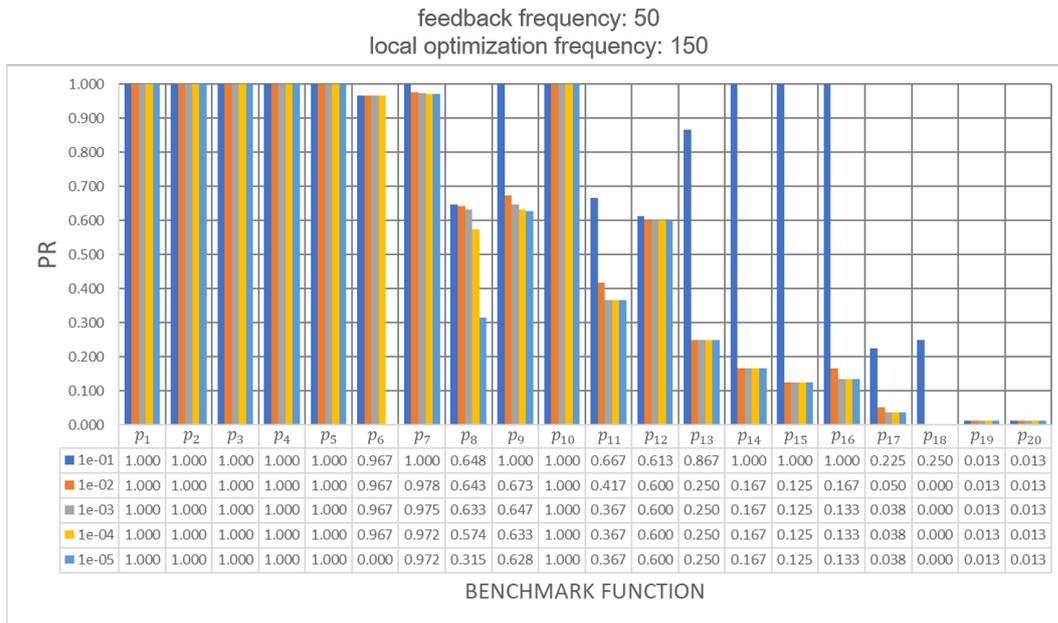


Figure C15: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 50, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

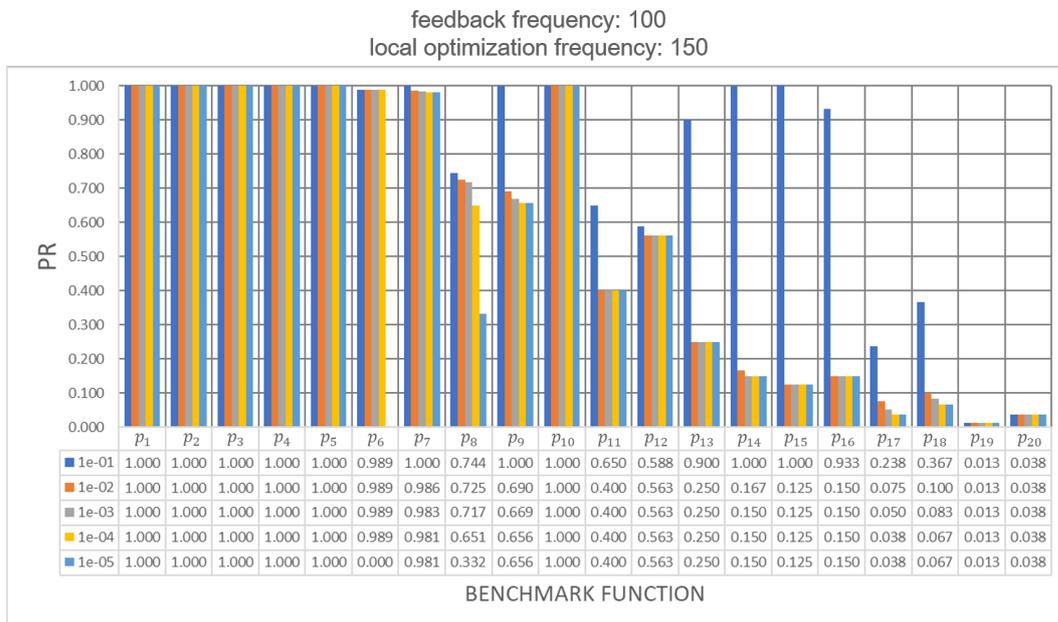


Figure C16: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 100, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

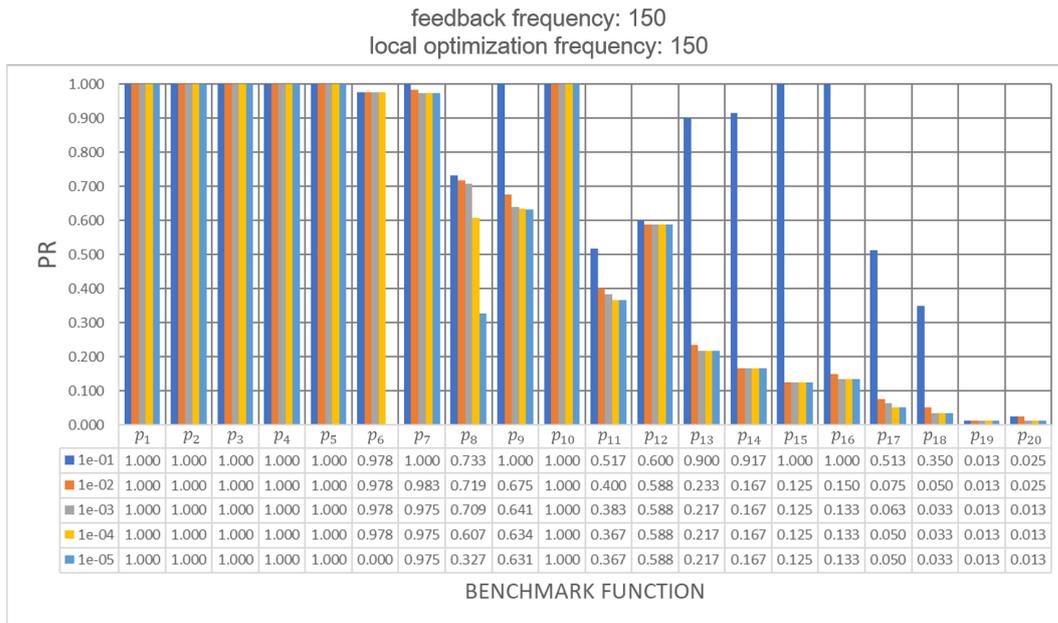


Figure C17: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 150, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

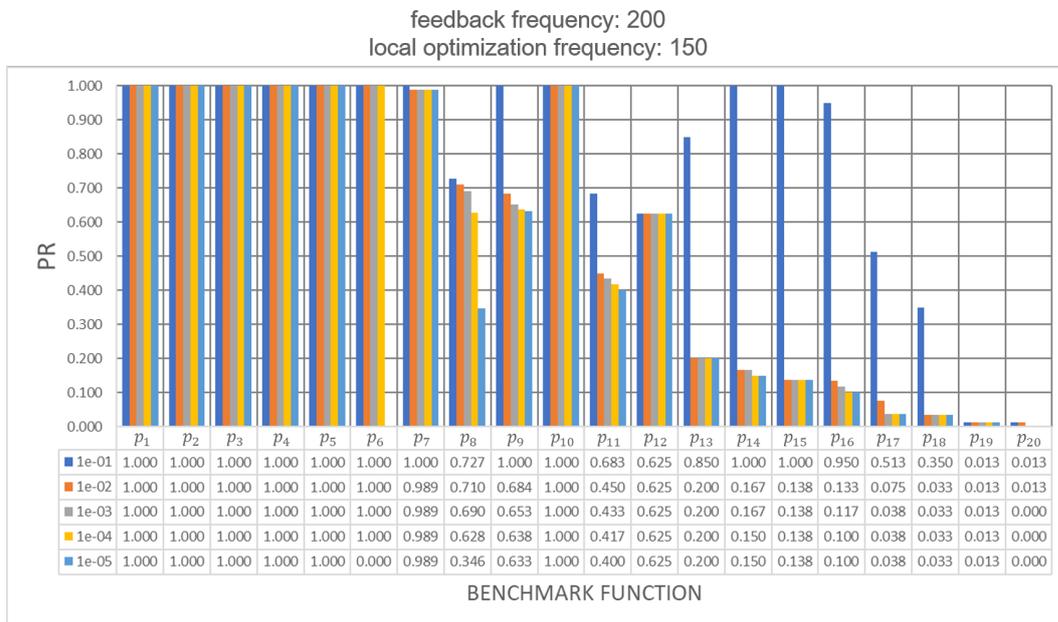


Figure C18: Bar chart with peak ratio (PR) results from Niching MG-PMA using high-level single-point crossover operator, and feedback frequency equal to 200, for 20 benchmark multimodal functions (p_1 - p_{20}) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

Tables C11-C15 present the Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

Table C11: Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-1$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1.

i	Niching MG-PMA	Rank
6	F50	83.750
5	F10	74.425
4	F150	73.100
3	F100	70.675
2	F0	68.375
1	F200	68.050
0	F1	55.125

Test	p -value
Aligned Friedman	0.008

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
F50	2.233	0.026	0.008	No
F10	1.505	0.132	0.010	No
F150	1.402	0.161	0.013	No
F100	1.212	0.225	0.017	No
F0	1.033	0.302	0.025	No
F200	1.008	0.314	0.050	No

Table C12: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-2$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1.

i	Niching MG-PMA	Rank
6	F50	88.525
5	F150	80.450
4	F0	70.275
3	F200	68.700
2	F10	65.300
1	F100	63.750
0	F1	56.500

Test	p -value
Aligned Friedman	0.008

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
F50	2.497	0.013	0.008	No
F150	1.867	0.062	0.010	No
F0	1.074	0.283	0.013	No
F200	0.951	0.342	0.017	No
F10	0.686	0.493	0.025	No
F100	0.565	0.572	0.050	No

Table C13: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-3$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1.

i	Niching MG-PMA	Rank
6	F50	92.375
5	F150	84.800
4	F200	74.075
3	F0	71.125
2	F10	59.525
1	F100	59.150
0	F1	52.450

Test	p -value
Aligned Friedman	0.010

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
F50	3.113	0.002	0.008	Yes
F150	2.522	0.012	0.010	No
F200	1.686	0.092	0.013	No
F0	1.456	0.145	0.017	No
F10	0.552	0.581	0.025	No
F100	0.522	0.601	0.050	No

Table C14: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-4$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F1.

i	Niching MG-PMA	Rank
6	F50	91.200
5	F150	88.975
4	F200	74.750
3	F0	70.425
2	F100	59.025
1	F10	57.150
0	F1	51.975

Test	p -value
Aligned Friedman	0.013

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
F50	3.058	0.002	0.008	Yes
F150	2.885	0.004	0.010	Yes
F200	1.776	0.076	0.013	No
F0	1.439	0.150	0.017	No
F100	0.550	0.583	0.025	No
F10	0.404	0.687	0.050	No

Table C15: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between different feedback frequency values (F0, F1, F10, F50, F100, F150 and F200) for Niching MG-PMA with $1e-5$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case the configuration F10.

i	Niching MG-PMA	Rank
6	F50	89.475
5	F150	88.475
4	F200	78.475
3	F0	67.100
2	F100	62.225
1	F1	54.475
0	F10	53.275

Test	p -value
Aligned Friedman	0.013

Niching MG-PMA	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
F50	2.822	0.005	0.008	Yes
F150	2.775	0.006	0.010	Yes
F200	1.965	0.049	0.013	No
F0	1.078	0.281	0.017	No
F100	0.698	0.485	0.025	No
F1	0.094	0.926	0.050	No

C5

Numerical Results for Comparison with Well-Established Algorithms

Tables C16-C20 present the Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between two configuration of Niching MG-PMA (Niching MG-PMA L1 and Niching MG-PMA L150) and two algorithms recommended by 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization (Li, 2013) (DE/nrand/1/bin and Crowding DE/rand/1/bin) with five level of accuracy: $\{1.0e-01, 1.0e-02, 1.0e-03, 1.0e-04, 1.0e-05\}$.

Table C16: Aligned Friedman's and Iman-Davenport's tests, and Holm's procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-01$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case Niching MG-PMA L150.

i	Algorithm	Rank
3	DE/nrand/1/bin	50.200
2	Niching MG-PMA L1	42.800
1	Crowding DE/rand/1/bin	37.775
0	Niching MG-PMA L150	31.225

Test	p -value
Aligned Friedman	0.025

Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
DE/nrand/1/bin	2.582	0.010	0.017	Yes
Niching MG-PMA L1	1.575	0.115	0.025	No
Crowding DE/rand/1/bin	0.891	0.373	0.050	No

Table C17: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-02$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.

i	Algorithm	Rank
3	Niching MG-PMA L150	47.075
2	Niching MG-PMA L1	41.750
1	Crowding DE/rand/1/bin	36.875
0	DE/nrand/1/bin	36.300

Test	p -value
Aligned Friedman	0.017

Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
Niching MG-PMA L150	1.466	0.143	0.017	No
Niching MG-PMA L1	0.742	0.458	0.025	No
Crowding DE/rand/1/bin	0.078	0.938	0.050	No

Table C18: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-03$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.

i	Algorithm	Rank
3	Niching MG-PMA L150	45.850
2	Niching MG-PMA L1	40.300
1	Crowding DE/rand/1/bin	39.500
0	DE/nrand/1/bin	36.350

Test	p -value
Aligned Friedman	0.017

Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
Niching MG-PMA L150	1.293	0.196	0.017	No
Niching MG-PMA L1	0.538	0.591	0.025	No
Crowding DE/rand/1/bin	0.429	0.668	0.050	No

Table C19: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-04$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.

i	Algorithm	Rank
3	Crowding DE/rand/1/bin	46.450
2	Niching MG-PMA L150	43.300
1	Niching MG-PMA L1	37.850
0	DE/nrand/1/bin	34.400

Test	p -value
Aligned Friedman	0.017

Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
Crowding DE/rand/1/bin	1.640	0.101	0.017	No
Niching MG-PMA L150	1.211	0.226	0.025	No
Niching MG-PMA L1	0.470	0.639	0.050	No

Table C20: Aligned Friedman’s and Iman-Davenport’s tests, and Holm’s procedure for pairwise comparison between Niching MG-PMA L1, Niching MG-PMA L150, DE/nrand/1/bin and Crowding DE/rand/1/bin for $1e-05$ accuracy. The reference rank (R_0) corresponds to the rank of the best algorithm ($i = 0$); in this case DE/nrand/1/bin.

i	Algorithm	Rank
3	Crowding DE/rand/1/bin	47.125
2	Niching MG-PMA L150	44.625
1	Niching MG-PMA L1	39.025
0	DE/nrand/1/bin	31.225

Test	p -value
Aligned Friedman	0.017

Algorithm	$z = (R_0 - R_i)/SE$	p -value	Holm	Reject?
Crowding DE/rand/1/bin	2.164	0.031	0.017	No
Niching MG-PMA L150	1.824	0.068	0.025	No
Niching MG-PMA L1	1.062	0.289	0.050	No

D

CEC'2017 Benchmark Multimodal Functions

D1

Introduction

This appendix presents the properties of the benchmark functions from 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization. The benchmark functions (p_1 - p_{20}) are based on 12 test functions (F_1 - F_{12}) with different dimensions and number of global optima. More details about the benchmark and test functions can also be found in the technical report of the competition (Li, 2013).

D1.1

Test Functions

Test functions from 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization are scalable to dimension and the number of global optima can be adjusted freely by the user. Table D1 presents the test functions of this competition and their properties.

Table D1: Test functions from 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization and their properties

Function	Name	Properties
F_1	Five-Uneven-Peak Trap	Variables ranges: $x \in [0, 30]$ Number of global optima: 2 Number of local optima: 3
F_2	Equal Maxima	Variables ranges: $x \in [0, 1]$ Number of global optima: 5 Number of local optima: 0
F_3	Uneven Decreasing Maxima	Variables ranges: $x \in [0, 1]$ Number of global optima: 1 Number of local optima: 4
F_4	Himmelblau	Variables ranges: $x, y \in [-6, 6]$ Number of global optima: 4 Number of local optima: 0
F_5	Six-Hump Camel Back	Variables ranges: $x \in [-1.9, 1.9]; y \in [-1.1, 1.1]$ Number of global optima: 2 Number of local optima: 2
F_6	Shubert	Variables ranges: $x_i \in [-10, 10]^D, i = 1, 2, \dots, D$ Number of global optima: $D \cdot 3^D$ Number of local optima: many
F_7	Vincent	Variables ranges: $x_i \in [0.25, 10]^D, i = 1, 2, \dots, D$ Number of global optima: 6^D Number of local optima: 0
F_8	Modified Rastrigin - All Global Optima	Variables ranges: $x_i \in [0, 1]^D, i = 1, 2, \dots, D$ Number of global optima: 12 Number of local optima: 0
F_9	Composition Function 1	Multimodal, Shifted, Non-Rotated, Non-symmetric, Separable near the global optima, Scalable, Numerous local optima, Different function's properties are mixed together, Number of global optima: 6
F_{10}	Composition Function 2	Multimodal, Shifted, Non-Rotated, Non-symmetric, Separable near the global optima, Scalable, Numerous local optima, Different function's properties are mixed together, Number of global optima: 8
F_{11}	Composition Function 3	Multimodal, Shifted, Rotated, Non-symmetric, Non-separable, Scalable, A huge numerous of local optima, Different function's properties are mixed together, Number of global optima: 6
F_{12}	Composition Function 4	Multimodal, Shifted, Rotated, Non-symmetric, Non-separable, Scalable, A huge numerous of local optima, Different function's properties are mixed together, Number of global optima: 8

Figure D1 presents the 3-D map for 2-D test functions.

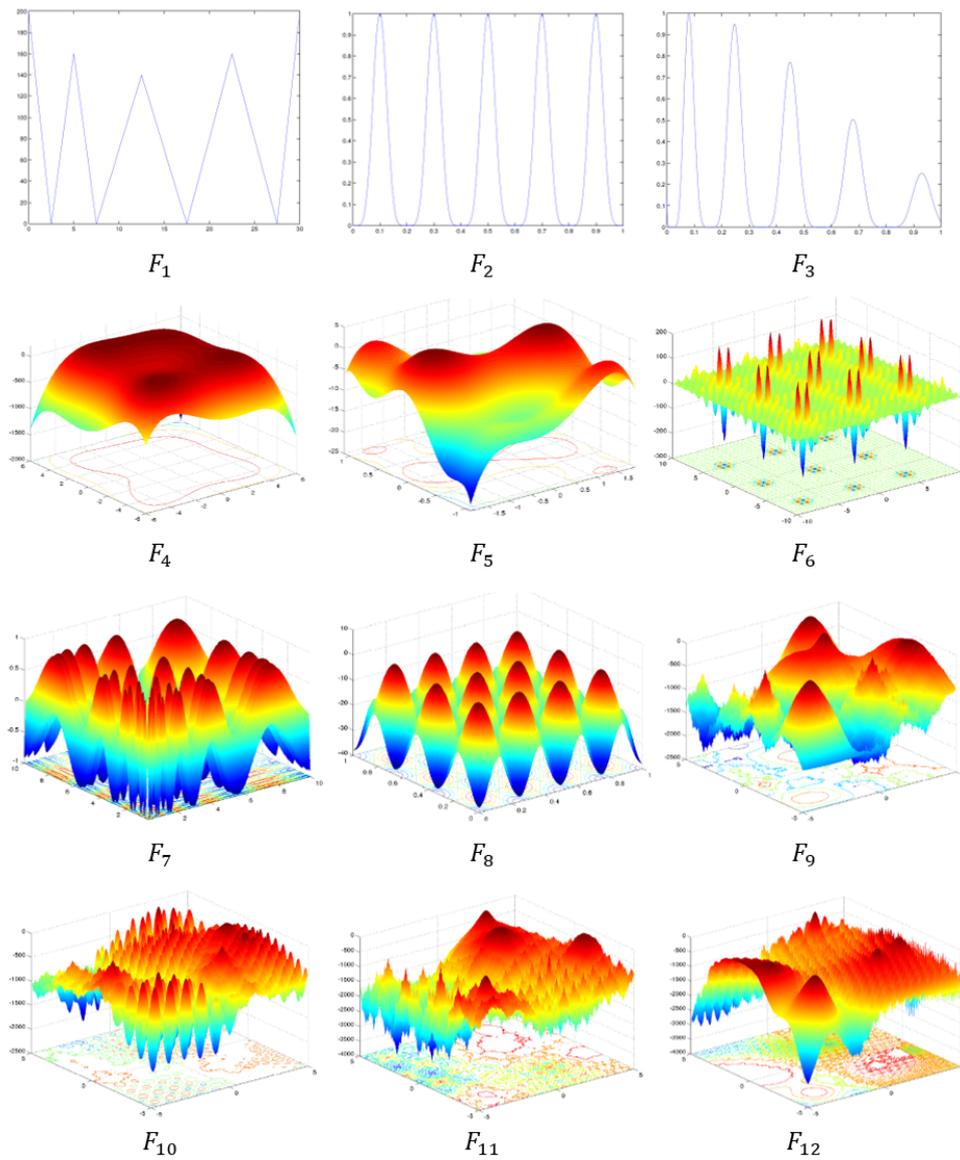


Figure D1: 3-D map for 2-D test functions (F_1 - F_{12}) (Li, 2013).

D1.2 Benchmark Functions

Table D2 presents the 20 multimodal benchmark functions (p_1 - p_{20}).

Table D2: Benchmark functions from 2017 IEEE CEC Special Session on Niching Methods for Multimodal Optimization

Benchmark Function	Test Function
p_1	F_1 (1D)
p_2	F_2 (1D)
p_3	F_3 (1D)
p_4	F_4 (2D)
p_5	F_5 (2D)
p_6	F_6 (2D)
p_7	F_7 (2D)
p_8	F_6 (3D)
p_9	F_7 (3D)
p_{10}	F_8 (2D)
p_{11}	F_9 (2D)
p_{12}	F_{10} (2D)
p_{13}	F_{11} (2D)
p_{14}	F_{11} (3D)
p_{15}	F_{12} (3D)
p_{16}	F_{11} (5D)
p_{17}	F_{12} (5D)
p_{18}	F_{11} (10D)
p_{19}	F_{12} (10D)
p_{20}	F_{12} (20D)