CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD. THESIS

Mashar Cenk GENÇAL

A STUDY TO IMPROVE PERFORMANCE OF GENETIC ALGORITHMS

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2019

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

A STUDY TO IMPROVE PERFORMANCE OF GENETIC ALGORITHMS

Mashar Cenk GENÇAL

PhD THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of Doctor of Philosophy by the board of jury on 29/11/2019

……..…….. ……..….
…….....…….……
Assoc. Prof. Dr. Mustafa ORAL Prof. Dr. S.Ayşe ÖZEL Assist. Prof. Dr. Ahmet AYDIN
SUPERVISOR MEMBER MEMBER

……..…….. ……..…..……
Assist. Prof. Dr. Ersin ÖZDEMİR Assist. Prof. Dr. Ahmet GÖKÇEN
MEMBER MEMBER

This PhD Thesis is written at the Department of Institute of Natural and Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

This work was supported by the Scientific Research Project Unit of Çukurova University (Project number: 8463).

Note: The usage of the presented specific declarations, tables, figures, and photographs either in this thesis or in any other reference without citation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic.

ABSTRACT

PhD THESIS

A STUDY TO IMPROVE PERFORMANCE OF GENETIC ALGORITHMS

Mashar Cenk GENÇAL

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc. Prof. Dr. Mustafa ORAL
Year: 2019, Pages: 147
Juries : Assoc. Prof. Dr. Mustafa ORAL
: Prof. Dr. S. Ayşe ÖZEL
: Assist. Prof. Dr. Ahmet AYDIN
: Assist. Prof. Dr. Ersin ÖZDEMİR
: Assist. Prof. Dr. Ahmet GÖKÇEN

Selection is one of the most crucial steps of Genetic Algorithms (GAs) that commonly used in areas of robot applications, image and voice recognition, artificial intelligence applications, path finding problems, scheduling problems, etc. In GAs, lack of adjusting the balance between exploration and exploitation, and selecting appropriate parameter settings are main problems of most selection methods as they cause premature convergence and trapping in local optima. In order to overcome these problems, two common techniques have been utilized: presenting a new selection method, tuning the parameter of an existing algorithm.

In the first part of study, new selection methods, Aggressive, Non-Aggressive, Integrated Aggressive, Integrated Non-Aggressive, Outlander, Non-Aggressive Outlander, Bipolar Mating Tendency (BMT), were proposed to solve the problems. As most of the methods are based on Standard Tournament Selection (ST), their performances were compared with ST and prevalent selection methods that are also based on ST: Restricted Tournament, Unbiased Tournament, Fine-Grained Tournament and Cooperative Selections. Twenty-one well known test functions in the field of GAs were employed for the comparison. Furthermore, non-parametric statistical tests, Friedman and Wilcoxon Signed Rank, were applied to demonstrate the significance of the results.

In the second part of the study, meta search methods (Brute Force and Coarse to Fine) and meta optimization algorithms (GAs, Particle Swarm Optimization and BMT) were applied to tune standard GAs in order to achieve its best performance. Moreover, the second part contains a short survey in the literature of meta optimization.

Keywords: Diversity methods, genetic algorithm, meta-optimization, non-parametric tests, selection methods, tournament

ÖZ

DOKTORA TEZİ

GENETİK ALGORTİMALARIN PERFORMANSINI GELİŞTİRME ÜZERİNE BİR ÇALIŞMA

Mashar Cenk GENÇAL

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. Mustafa ORAL
Yıl: 2019, Sayfa: 147
Jüri : Doç. Dr. Mustafa ORAL
: Prof. Dr. S. Ayşe ÖZEL
: Dr. Öğr. Üyesi Ahmet AYDIN
: Dr. Öğr. Üyesi Ersin ÖZDEMİR
: Dr. Öğr. Üyesi Ahmet GÖKÇEN

Seçim işlemi, robot uygulamaları, görüntü ve ses tanıma, yapay zekâ uygulamaları, yol bulma problemleri, çizelgeleme problemleri vb. gibi alanlarda sıklıkla kullanılan Genetik Algoritmaların (GA) en önemli adımlarından biridir. GA'larda, arama ve didikleme arasındaki uyumun ayarlanamaması ve uygun parametre ayarlarının seçilememesi, erken yakınsamaya ve lokal optimuma sıkışmaya neden olabileceğinden, çoğu seçim yönteminin temel sorunlarındandır. Bu problemlerin üstesinden gelmek için iki yaygın yöntem kullanılmaktadır: yeni bir seçim yöntemi sunmak, mevcut bir algoritmanın parametrelerini ayarlamak.

Çalışmanın ilk bölümünde, GA'ların problemlerini çözmek için yeni seçim yöntemleri önerilmiştir: Saldırgan, Saldırgan Olmayan, Bütünleşik Saldırgan, Bütünleşik Saldırgan Olmayan, Yabancı, Saldırgan Olmayan Yabancı, Bipolar Eşleşme Eğilimi (BMT). Yöntemlerin çoğu Standart Turnuva (ST) seçimine dayandığından, performansları ST ve ST'ye dayanan yaygın seçim yöntemleri olan Sınırlı Turnuva, Tarafsız Turnuva, İnce Taneli Turnuva ve Kooperatif Turnuva yöntemleri ile karşılaştırılmıştır. Karşılaştırma için GA alanında iyi bilinen yirmi bir test fonksiyonu kullanılmıştır. Ayrıca, sonuçların anlamlı olduğunu göstermek için parametrik olmayan istatistiksel testler, Friedman ve Wilcoxon İşaretli Sıra, uygulanmıştır.

İkinci bölümde ise, standart GA'ların en iyi performansını elde etmek için, meta arama yöntemleri (Kaba Kuvvet ve Kabadan İnceye Arama) ve meta optimizasyon algoritmaları (GA, Parçacık Sürü Optimizasyonu ve BMT) uygulanarak standart GA'ların parametre ayarları belirlenmiştir. Ayrıca, ikinci kısım, meta optimizasyon literatürü için kısa bir araştırma içermektedir.

Anahtar kelimeler: Çeşitlilik yöntemleri, genetik algoritma, meta-optimizasyon, parametrik olmayan testler, seçim yöntemleri, turnuva

EXTENDED ABSTRACT

Genetic Algorithms (GAs) were introduced by John Holland in 1975. The origin of these algorithms is based on Darwin's natural selection theory. The idea has inspired many studies, and the efficient search of GAs in search space has made them the most common methods in optimization problems for finding the optimal solution.

GAs consist of four main parts: selection, crossover, mutation and the creation of new population. Generally, the selection process choses the best individual in the population (the individual whose fitness value is best, according to the criteria) based on the value found by a fitness function. The crossover process starts immediately after the selection process to obtain new individual or individuals. The goal of mutation is to prevent early convergence to a local optimum and to maintain genetic diversity. For this purpose, it modifies the genetic structure of the selected individuals. After these steps are completed, the next generation of the population is created by combining a proportion of new offspring and the members of the current population. Then, the algorithm returns to the selection process and repeats the steps until the termination criteria are met; either optimum solution is found or maximum number of generations are reached. However, since the best individuals are favored for mating, the selection process is expected to improve the population step by step. Therefore, it should have an adequate selection method in order to meet this expectation.

A good selection method should present decent results in all function types such as unimodal, multimodal, noisy or deceptive. Improvements in selection methods that can give good results in different types of functions is significantly important task in areas where GAs are commonly used (parameter and system diagnostics, control systems, robot applications, image and sound recognition, engineering designs, planning, artificial intelligence applications, expert systems, function and combinational optimization problems, network design problems, routing problems, scheduling problems, social and economic planning).

The aim of our study is to improve the performance of GAs by introducing new selection methods. For this purpose, in the first part of the thesis, Aggressive (ASM), Non-Aggressive (NASM), Integrated Aggressive (IASM), Integrated Non-Aggressive (INASM), Outlander (OA), Non-Aggressive Outlander (N-OA), Bipolar Mating Tendency (BMT) were proposed.

The performances of proposed methods are comparatively analyzed with Standard Tournament Selection (ST) and prevalent selection methods that are based on ST; Restricted Tournament Selection (RTS), Unbiased Tournament Selection (UTS), Fine-Grained Tournament Selection (FGS) and Cooperative Selection (CS). Twenty-one test functions, well known in the field of GAs, were used for the comparison. Identical parameter settings are applied to all selection methods such that crossover probability = 0.7, tournament size = 4, mutation probability = 0.05, and population sizes = {50, 100, 200}. The tests are repeated with 25 runs that have 25 different random seeds, in order to minimize the effects of randomization. The results presented in the thesis are the median and quartile values of the best results from these randomized runs.

In the second part of the thesis, meta optimization methods are used to find the most suitable parameter settings for a standard GA. Due to its performance in the first part of our study, the proposed method, BMT, was applied as a meta optimizer and the results were compared with the standard GA, Particle Swarm Optimization (PSO) and Coarse to Fine Search under equal conditions:

Meta optimization used in optimization problems refers that improving performance of an algorithm by tuning parameters of the algorithm by another algorithm. Besides improving efficiency of the algorithm, it also assists us to understand how the algorithm behaves in different types of problems. Meta optimization techniques can be classified in two main groups: off-line and on-line.

In off-line tuning methods, parameter settings are identified before the algorithm starts. Initially, an off-line method analyzes some number of parameter configurations on a training set of instances. Then, the chosen configuration is

applied to all instances that are struggled. This technique works well if selected instances have equal structure to other instances in the training set. However, it might not work when the class of instances is more heterogeneous. Since the instances have diversity structures, choosing appropriate parameter settings takes large computational time.

On-line tuning methods adjust parameter configuration of an algorithm while the algorithm is working to solve one of the problem instances. They collect feedbacks during the process of the algorithm to decide the best parameter configuration for that instance. This principle is applied to all problem instances to find suitable parameter configuration of each instances. Hence, it is obvious to notice that on-line methods provide saving the computational time while off-line tuning methods takes more time.

In order to work a standard GA effectively, its parameters need to be well selected. For this purpose, the Brute Force Search method was first applied using the given parameter settings: population size = {25, 50, 100, 200}, tournament size = {3, 4, 5, 6}, cross over probability = 0.25 to 1 in 0.05 steps, mutation probability = 0.025 to 0.1, in 0.005 steps, over 300 iterations. Each test was repeated 25 times with 25 different random start points to reduce the effect of randomness, and to increase the reliability and accuracy of the tests.

Although Brute Force Search provides the desired results to be achieved, it generates a large computational budget since it analyzes about 4000 parameters. For this reason, with only 400 parameter configurations, Coarse to Fine Search has been applied to reduce the computational budget. Additionally, Standard GA, PSO and BMT were used with the same number of parameter configurations and the results were compared with Coarse to Fine Search Algorithm.

When the results obtained from the studies were examined, it was observed that the proposed selection methods; especially INASM, N-OA and BMT, yielded more successful results than ST and the commonly used selection methods that are based on ST; RTS, UTS, FGS and CS. Furthermore, BMT showed a decent

performance in the second part of the study as in the first part of the thesis and gave the best results among the compared methods.

GENİŞLETİLMİŞ ÖZET

Genetik Algoritmalar ilk olarak John Holland tarafından 1975 yılında sunulmuştur. Bu algoritmaların temeli Darwin'in doğal seçim teorisine dayanır. Holland'ın bu fikri, birçok çalışmaya ilham vermiş ve Genetik Algoritmaların arama uzayında etkili bir arama yapması, bu algoritmaları, optimizasyon problemlerinde en iyi çözümü bulmak için sık kullanılan yöntemler haline getirmiştir. Genetik Algoritmalar 4 ana bölümden oluşur: seçim, çaprazlama, mutasyon ve yeni popülasyon oluşturma. Seçim işlemi, genellikle, popülasyon içindeki en iyi bireyi (uygunluk değeri aranan kriterlere göre en iyi olan bireyi), bir uygunluk fonksiyonu aracılığıyla bulunan değere göre seçer. Çaprazlama işlemi, seçim işleminden hemen sonra başlar. Seçim işlemi aracılığıyla, popülasyon içerisinden seçilmiş bireyleri çaprazlayarak yeni birey ya da bireylerin elde edilmesini sağlar. Mutasyon işleminin görevi ise lokal optimuma erken yakınsamayı önlemek ve genetik çeşitliliği sürdürmektir. Bu amaçla, seçilen bireylerin genetik yapısını değiştirerek popülasyona yeni birey ya da bireyler kazandırır. Bu adımlar tamamlandıktan sonra, eski popülasyon ile oluşan yeni bireyler belirli oranda karıştırılarak yeni bir popülasyon oluşturulur. Yukarıda anlatılan bu aşamalar, seçim işleminden tekrar başlatılarak istenilen sonuç elde edilene ya da aranan kriter bulunana kadar devam ettirilir. Diğer yandan, seçim işlemlerinde her jenerasyonda genellikle en iyi bireyler seçildiğinden, seçim işlemlerinin popülasyonu adım adım iyiye (global optimum değere) götürmesi beklenmektedir. Bundan dolayı, seçim işleminin bu beklentiyi karşılaması için iyi bir seçim yöntemine sahip olması gerekir.

İyi bir seçim yönteminin sadece tek biçimli (unimodal) fonksiyonlarda değil, aynı zamanda çok biçimli (multimodal), düzensiz (noisy) veya aldatıcı (deceptive) fonksiyonlarda da iyi sonuçlar vermesi gerekir. Farklı fonksiyon türlerinde de iyi sonuçlar verebilen seçim yöntemlerinde yapılacak iyileştirmeler, Genetik Algoritmaların yoğunlukla kullanıldığı alanlarda (parametre ve sistem tanılama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik

tasarımları, planlama, yapay zekâ uygulamaları, uzman sistemler, fonksiyon ve kombinasyonlu eniyileme problemleri, ağ tasarım problemleri, yol bulma problemleri, çizelgeleme problemleri, sosyal ve ekonomik planlama) performansı büyük ölçüde artıracağından önem arz etmektedir.

Yaptığımız çalışmanın amacı, yeni seçim yöntemleri sunarak, Genetik Algoritmaların performansında iyileştirme yapmaktır. Bu amaçla, tezin ilk bölümünde Saldırgan (ASM), Saldırgan Olmayan (NASM), Bütünleşik Saldırgan (IASM), Bütünleşik Saldırgan Olmayan (INASM), Yabancı (OA), Saldırgan Olmayan Yabancı (N-OA), Bipolar Eşleşme Eğilimi (BMT), seçim yöntemleri önerilmiştir:

Önerilen yöntemlerin çoğu Standart Turnuva (ST) seçim yöntemine dayandığından, performansları ST ve ST'ye dayanan yaygın seçim yöntemleri olan Sınırlı Turnuva (Restricted Tournament Selection, RTS), Tarafsız Turnuva (Unbiased Tournament Selection, UTS), İnce Taneli Turnuva (Fine-Grained Tournament Selection, FGS) ve Kooperatif Turnuva (Cooperative Selection, CS) ile karşılaştırılmıştır. Karşılaştırma için Genetik Algoritmalar alanında iyi bilinen yirmi bir test fonksiyonu kullanılmıştır. Yapılan testler için parametreler; çaprazlama olasılığı = 0.7, turnuva boyutu = 4, mutasyon olasılığı = 0.05 ve popülasyon boyutları = {50, 100, 200} olacak şekilde uygulanmıştır. Testler, rassallığın etkisini en aza indirmek için 25 farklı rastgele başlangıç noktası içeren 25 farklı çalıştırma ile tekrarlanmıştır. Tezde sunulan sonuçlar, bu 25 çalıştırmadan elde edilen en iyi sonuçların medyan ve çeyrekler açıklığı değerleridir.

Tezin ikinci bölümünde ise standart bir Genetik Algoritma için en uygun parametre ayarlarını bulmak amacıyla meta optimizasyonu yöntemlerinden yararlanılmıştır. Önerdiğimiz iyileştirici yöntem, BMT, çalışmamızın ilk bölümünde gösterdiği performanstan dolayı, tezin ikinci bölümünde meta optimizasyonu olarak uygulanmış ve alınan sonuçlar Standart Genetik Algoritması, Parçacık Sürü Optimizasyonu ve Kabadan İnceye Arama (Coarse to Fine Search) yöntemleriyle eşit koşullar altında kıyaslanmıştır:

Optimizasyon problemlerinde kullanılan meta optimizasyonu, bir algoritmanın performansını arttırmak için; başka bir algoritma tarafından parametrelerinin ayarlanması anlamına gelir. Algoritmanın verimliliğini arttırmanın yanı sıra, algoritmanın nasıl davrandığını da anlamamıza yardımcı olur. Meta optimizasyon yöntemleri iki ana grupta sınıflandırılabilir: çevrimdışı ve çevrimiçi.

Çevrim dışı yöntemlerde, algoritma çalışmadan önce parametre ayarları tanımlanır. Başlangıçta çevrimdışı bir yöntem, bir dizi eğitim seti üzerinde bazı parametre yapılandırmalarını analiz eder. Ardından, seçilen yapılandırma, üzerinde çalışılan tüm örneklere uygulanır. Eğer seçilen yapılandırma eğitim setindeki diğer örneklerle eşit yapıya sahipse, çevrim dışı yöntem iyi çalışır. Ancak, örneklerin yapıları birbirinden farklıysa işe yaramayabilir. Bu durumda, en iyi performansı elde etmek için uygun parametre ayarlarının seçilmesi, hesaplama zamanının artmasına sebep olacaktır.

Çevrim içi yöntemler, algoritma problem örneklerinden birini çözmek için çalışırken, algoritmanın farklı parametre değerlerini test ederler. Verilen örneğin en iyi parametre yapılandırmasına karar vermek için süreç boyunca geri bildirimler toplarlar. Bu prensip, eğitim setindeki her örneğin farklı parametre yapılandırmalarını bulmak için tüm örneklere uygulanır. Bu nedenle, çevrim içi yöntemlerin çevrim dışı yöntemlere göre daha başarılı olduğu düşünülebilir. Standart bir Genetik algoritmasını iyileştirmek için çevrim içi bir yöntem kullanma nedenimiz de budur.

Standart bir Genetik algoritmanın efektif bir şekilde çalışması için, parametrelerinin iyi seçilmesi gerektir. Bu amaçla, ilk olarak Kaba Kuvvet Arama (Brute Force Search) yöntemi, verilen parametre ayarları kullanılarak uygulanmıştır: nüfus büyüklüğü = {25, 50, 100, 200}, turnuva boyutu = {3, 4, 5, 6}, çaprazlama olasılığı = 0,25'den 1'e 0.05 adımlarla, mutasyon olasılığı = 0.025'den 0.1'e, 0.005 adımlarla, 300 yineleme boyunca. Rassallık etkisini azaltmak, testlerin güvenilirliğini ve doğruluğunu arttırmak için her test 25 farklı rasgele başlangıç noktası ile 25 kez tekrarlanmıştır.

Kaba Kuvvet Arama ulaşılmak istenen sonuçları sunmasına rağmen, yaklaşık 4000 parametre analiz ettiği için, büyük hesaplama bütçesine neden olmaktadır. Bu sebeple, hesaplama bütçesini azaltmak amacıyla, yalnız 400 parametre yapılandırmasıyla Kabadan İnceye Arama algoritması uygulanmıştır. Ayrıca, meta optimizasyon yöntemi olarak kullanılan Standart Genetik Algoritması, Parçacık Sürü Optimizasyonu ve BMT de aynı sayıda parametre yapılandırması ile uygulanmış ve alınan sonuçlar Kabadan İnceye Arama yöntemi ile kıyaslanmıştır.

Yapılan çalışmalardan elde edilen sonuçlar incelendiğinde, önerilen seçim yöntemlerinden özellikle INASM, N-OA ve BMT, ST ve ST'ye dayalı yaygın olarak kullanılan seçim yöntemlerine (RTS, UTS, FGS ve CS) kıyasla daha başarılı sonuçlar verdiği gözlemlenmiştir. Ayrıca, BMT çalışmamızın ilk bölümünde gösterdiği performansı, tezin ikinci bölümünde de göstermiş ve kıyaslanan yöntemler arasında en iyi sonucu sunmuştur.

ACKNOWLEDGEMENTS

I would like to appreciate Assoc.Prof. Dr. Mustafa ORAL and Assist. Prof. Dr. B.Melis ÖZYILDIRIM for their encouragements, motivations and useful suggestions.

I would like to thank members of PhD thesis jury; Prof. Dr. S. Ayşe ÖZEL, Assist. Prof. Dr. Ahmet AYDIN, Assist. Prof. Dr. Ersin ÖZDEMİR and Assist. Prof. Dr. Ahmet GÖKÇEN, for their suggestions.

Special thanks to my wife Gülizar Burcu GENÇAL for her endless love, support, encouragement and motivation.

Finally, I would like to thank my family that raised me and brought me to this day.

CONTENTS **PAGE**

ABSTRACT.....I
ÖZ II
EXTENDED ABSTRACT III
GENİŞLETİLMİŞ ÖZET VII
ACKNOWLEDGEMENTS.....XI
LIST OF TABLES XVI
LIST OF FIGURES XVIII
ABBREVATIONS..... XX
1. INTRODUCTION 1
1.1. The Problem and Problem Approach..... 2
1.2. Aims and Objectives..... 3
1.3. Contribution to Existing Knowledge 3
1.4. The Structure of the Thesis..... 4
2. LITERATURE REVIEW 5
2.1. Introduction..... 5
2.2. Literature Review on Genetic Algorithms 5
2.3. Literature Review on Meta Optimization 8
3. MATERIALS AND METHODS..... 13
3.1. Genetic Algorithms..... 13
3.2. Selection Methods..... 15
3.2.1. Proportionate Selection Methods 15
3.2.1.1. Roulette Wheel..... 15
3.2.1.2. Stochastic Sampling..... 16
3.2.1.3. Stochastic Remainder Sampling 17
3.2.1.4. Deterministic Selection..... 17
3.2.1.5. Stochastic Universal Sampling 18
3.2.2. Ranking 18

3.2.2.1. Linear Ranking 18
3.2.2.2. Exponential Ranking 19
3.2.2.3. Truncation Selection 19
3.2.3. Tournament 19
3.2.3.1. Boltzmann Tournament Selection 20
3.2.3.2. Restricted Tournament Selection 21
3.2.3.3. Correlative Tournament Selection 21
3.2.3.4. Fine-Grained Tournament Selection 21
3.2.3.5. Unbiased Tournament Selection 22
3.2.3.6. Cooperative Tournament Selection 22
3.2.4. Steady State Genetic Algorithm 22
3.2.5. Elitism 23
3.2.6. Boltzmann Selection 23
3.2.7. Fitness Sharing Algorithm 23
3.2.8. Local Selection 24
3.2.9. Age Based Selection 24
3.2.10. High Low Fit Selection 24
3.2.11. Neural Network Based Selection 25
3.2.12. Sexual Selection 25
3.3. Meta Optimization 25
3.4. The Methods of Meta Optimizations 27
3.4.1. Coarse to Fine Search 28
3.4.2. Genetic Algorithms 29
3.4.3. Particle Swarm Optimization (PSO) 29
3.4.4. Bipolar Mating Tendency 30
3.5. Proposed Methods 30
3.5.1. Introduction 30
3.5.2. Aggressive Selection Method 31
3.5.3. Integrated Aggressive Selection Method 32

3.5.4. Non-Aggressive Selection Method 33
3.5.5. Integrated Non-Aggressive Selection Method 35
3.5.6. Outlander Algorithm 35
3.5.7. Non-Aggressive Outlander Algorithm 36
3.5.8. Bipolar Mating Tendency 37
3.6. Tests 38
3.6.1. Introduction 38
3.6.2. Test Functions (Benchmarks) 39
3.6.2.1. Ackley's Function 40
3.6.2.2. Axis Parallel Hyper-Ellipsoid Function 40
3.6.2.3. Booth's Function 41
3.6.2.4. Branins's Function 41
3.6.2.5. Sixth Function of Bukin 41
3.6.2.6. De Jong's Function 41
3.6.2.7. Fifth Function of De Jong 42
3.6.2.8. Drop Wave Function 42
3.6.2.9. Easom's Function 42
3.6.2.10. Goldstein-Price's Function 43
3.6.2.11. Griewangk's Function 43
3.6.2.12. Holder Table Function 43
3.6.2.13. Langermann's Function 43
3.6.2.14. Michalewicz's Function 44
3.6.2.15. Rastrigin's Function 44
3.6.2.16. Rosenbrock's Valley 44
3.6.2.17. Rotated Hyper-Ellipsoid Function 44
3.6.2.18. Shubert's Function 45
3.6.2.19. Schwefel's Function 45
3.6.2.20. Six Hump Camel Back Function 45
3.6.2.21. Sum of Different Powers Function 45

3.6.3. Statistical Tests 46
3.6.3.1. Friedman Test 46
3.6.3.2. Wilcoxon Signed Rank Test 47
4. RESULTS AND DISCUSSIONS 49
4.1. Introduction 49
4.2. Preliminary Tests 49
4.3. Results by Benchmarks 50
4.4. Discussion for the Benchmark Tests 61
4.4.1. Discussion on ASM 62
4.4.2. Discussion on IASM 62
4.4.3. Discussion on NASM 63
4.4.4. Discussion on INASM 63
4.4.5. Discussion on OA 64
4.4.6. Discussion on N-OA 64
4.4.7. Discussion on BMT 64
4.4.7.1. Why BMT Offers Better Results than ST? 65
4.4.7.2. Offering an Instance 65
4.4.7.3. Compare Genetic Diversities of BMT and ST 66
4.4.7.4. Harmony between Exploration and Exploitation 69
4.5. Results and Discussion for the Statistical Tests 73
4.5.1. Friedman Test 73
4.5.2. Wilcoxon Signed Rank Test 74
4.6. Results by Meta Optimization Techniques 78
4.7. Discussion for the Meta Tests 83
5. CONCLUSIONS 85
REFERENCES 87
CURRICULUM VITAE 95
APPENDIX 97

LIST OF TABLES **PAGE**

Table 3.1 Mating permissions of individuals in a population.....33
Table 3.2 Benchmarks with their classes and global values.....40
Table 4.1.1 Test results for Ackley function…….....51
Table 4.1.2 Test results for Axis Parallel Hyper-Ellipsoid function….....51
Table 4.1.3 Test results for Booth function…….....52
Table 4.1.4 Test results for Branins's function…….....52
Table 4.1.5 Test results for Sixth Function of Bukin..……..….....53
Table 4.1.6 Test results for De Jong's function…….....53
Table 4.1.7 Test results for Fifth Function of De Jong……..….....54
Table 4.1.8 Test results for Drop Wave function.....54
Table 4.1.9 Test results for Easom function…….....55
Table 4.1.10 Test results for Goldstein-Price's function….....55
Table 4.1.11 Test results for Griewangk's function…….....56
Table 4.1.12 Test results for Holder Table function.....56
Table 4.1.13 Test results for Langermann's function…….....57
Table 4.1.14 Test results for Michalewicz's function…….....57
Table 4.1.15 Test results for Rastrigin's function…….....58
Table 4.1.16 Test results for Rosenbrock's Valley function…….....58
Table 4.1.17 Test results for Rotated Hyper-Ellipsoid function…….....59
Table 4.1.18 Test results for Shubert's function…….....59
Table 4.1.19 Test results for Schwefel's function…….....60
Table 4.1.20 Test results for Six Hump Camel Back function…….....60
Table 4.1.21 Test results for Sum of Different Powers function…….....61
Table 4.2 Snapshot of a population for a hypothetical problem …….....65
Table 4.3 Mean ranks of the methods for the Friedman tests.....73
Table 4.4 Statistical values for the Friedman tests.....74
Table 4.5 The pairwise results of Wilcoxon signed rank test (pop size = 50)..75

Table 4.6 The pairwise results of Wilcoxon signed rank test (pop size = 100)76
Table 4.7 The pairwise results of Wilcoxon signed rank test (pop size = 200)77
Table 4.8 Parameter settings recommended by Brute Force Search.....80
Table 4.9 Parameter settings recommended by of Coarse to Fine Search.....81
Table 4.10 Parameter settings recommended by of GAs…….....81
Table 4.11 Parameter settings recommended by of PSO…….....82
Table 4.12 Parameter settings recommended by of BMT...…….....82
Table 4.13 Test results of all meta techniques……..83

LIST OF FIGURES **PAGE**

Figure 3.1 The process of Genetic Algorithms.....14
Figure 3.2 Approaches for parameter tuning.....26
Figure 3.3 The process of meta optimization.....27
Figure 3.4 Specifying better configurations in randomly selected parameters.28
Figure 3.5 Check the neighborhood of the best configurations….....29
Figure 4.1 Proposed methods in Axis Parallel Hyper Ellipsoid……..62
Figure 4.2 The phenotype measurements of BMT with ST.....66
Figure 4.3 The genotype measurements of BMT with ST 67
Figure 4.4 The performances of BMT and ST in De Jong's function ……..68
Figure 4.5 The discovered points from BMT and ST in different pop sizes….72

ABBREVATIONS

ANN	: Artificial Neural Network
ASM	: Aggressive Selection Method
BMT	: Bipolar Mating Tendency
CS	: Cooperative Selection
DE	: Differential Evolution
EAs	: Evolutionary Algorithms
FIFO	: First in First Out
FGTS	: Fine-Grained Tournament Selection
GAs	: Genetic Algorithms
IASM	: Integrated Aggressive Selection Method
INASM	: Integrated Non-Aggressive Selection Method
NASM	: Non- Aggressive Selection Method
LR	: Linear Ranking
N-OA	: Non-Aggressive Outlander Algorithm
OA	: Outlander Algorithm
PSO	: Particle Swarm Optimization
ROAR	: Randomized Online Aggressive Racing
RTS	: Restricted Tournament Selection
RW	: Roulette Wheel
SSR	: Stochastic Sampling with Replacement
SSWR	: Stochastic Sampling without Replacement
SRSR	: Stochastic Remainder Sampling with Replacement
SRSWR	: Stochastic Remainder Sampling without Replacement
ST	: Standard Tournament Selection
SA	: Simulated Annealing
SPO	: Sequential Parameter Optimization
SMAC	: Sequential Model Based Algorithm Configuration

UTS : Unbiased Tournament Selection

1. INTRODUCTION

Evolutionary Algorithms (EAs) are successfully employed to solve complex optimization problems such as; robot applications, image and voice recognition, artificial intelligence applications, path finding problems, scheduling problems network design, social and economic planning etc. Particularly, Genetic Algorithms (GAs) are one of the well-known EAs. GAs iteratively improve the initial solution by using intelligent strategies, which include selection, recombination, mutation, and produce new solutions.

In GAs, the selection process is crucial due to its effect on the diversity of a population and convergence ability to optimal solution. Therefore, it should have an adequate selection method in order to take advantages of the selection process. However, most selection methods, especially common ones (Roulette Wheel, Linear Ranking, Tournament), are insufficient to overcome the main problems of GAs. One main problem of GAs is the lack of adjusting the balance between exploration and exploitation.

As the name implies, exploration is an activity that discovers information or resources. For instance, animals are always in search of nature to satisfy their natural needs such as food and accommodation. In addition to them, exploration also occur in humans. As in discovery of America, they tend to find promising regions with valuable resources. Since the majority of the selection methods have been influenced by natural life, they should also tend to explore better regions or resources. In general, researchers have utilized large population sizes to discover most part of the search space. Nevertheless, this traditional technique causes increase in computational budget.

On the other hand, exploitation refers researching a limited (but promising) area of the search space owing to hope developing the promising solution. In selection methods, giving priority to good solutions in reproduction process is the common way for exploitation. However, when they favor the good solutions,

average or poor solutions are overlooked, mostly dismissed from reproduction process. Hence, they lose certain amount of genetic material found in less favorable individuals. This causes to lose the genetic diversity of population, to get stuck in local optima and premature convergence.

Another problem that GAs generally encounter is choosing optimum parameter settings. Parameter selection directly affects how an algorithm behaves; therefore, it is significantly important to select efficient parameters such as crossover rate, mutation rate, tournament size, population size etc. Many researchers have generally tuned these parameters by utilizing "default parameter settings" (Lobo and Goldberg 2004). However, this technique might not offer desired solutions for all problem types. For example, selecting a decent parameter configuration can find the optimum value in one of the problem instances, yet, that parameter configuration may not satisfy other problem instances.

1.1. The Problem and Problem Approach

Many Evolutionary Algorithms (EAs) such as Particle Swarm Optimization (PSO) and GAs abandon to explore search space when they find a solution near to the optimal value. The reason why they give up from exploration is a common belief that exploiting the area improves the optimal solution. However, this approach may cause the algorithms to get stuck to a local optimum point or to miss the global value.

The selection methods of EAs play an important role on the quality of next generation. However, particularly in GAs, they are not accomplished to provide a sustainable exploration as they especially assist to exploitation. Because of their expectation of having a better generation than before, they ignore less favorable individuals and replace these individuals with better copies of individuals. On the other hand, disappearance of the certain amount of genetic material found in less favorable individuals induces to lose the genetic diversity of the population. In order to solve the problem with GAs, it is required to have a sufficient selection method which has a process that keeps genetic diversity alive.

Moreover, the lack of selecting appropriate parameter settings is another main problem of GAs. When an algorithm presents better solution in one of the problem types, it does not mean that the algorithm works well in other problem types. Hence, it is required to tune the algorithm to observe its best performance in different conditions.

In order to overcome these problems, two common techniques have been utilized: presenting a new selection method or adjusting parameters of an algorithm by applying meta optimization techniques.

1.2. Aims and Objectives

A decent selection method should give adequate results in irregular, noisy or deceptive functions. Hence, it is aimed to achieve the qualified selection methods that overcome not only basic functions, but also irregular, noisy or deceptive functions. This includes the following objectives;

- To identify most common selection methods in the literature of GAs
- To test effectiveness of exploitation by proposing different type of algorithms
- To analyze influence of exploration
- To benefit from advantages of existing knowledge to create new methods
- To find the best parameter settings of GAs to tune their parameters by operating meta optimization techniques

1.3. Contribution to Existing Knowledge

In the first part of study, **seven new selection methods**, Aggressive (ASM), Non-Aggressive (NASM), Integrated Aggressive (IASM), Integrated Non-Aggressive (INASM), Outlander (OA), Non-Aggressive Outlander (N-OA), Bipolar Mating Tendency (BMT), were proposed. These selection methods can be classified

into three groups: **exploitation based**; ASM and NASM, **hybrids**; IASM and INASM, and **exploration based**; OA, N-OA and BMT.

On the other hand, in the second part of study, **there does not exist such a study that tests two search algorithms (Brute Force Search, Coarse to Fine Search) with three meta optimization methods (GA, PSO, BMT) and presents a short survey in the literature of meta optimization.**

1.4. The Structure of the Thesis

The thesis is organized as follows: In Chapter two, studies in the literature of GAs and meta optimizations are briefly explained. In Chapter three, in addition to explaining GAs and meta optimizations, common selection methods, proposed methods and performed tests are introduced. While results and discussion take place in Chapter four, the thesis is concluded in Chapter five.

2. LITERATURE REVIEW

2.1. Introduction

As long as we can create and evaluate qualified individuals, Genetic Algorithms (GAs) can solve an optimization problem efficiently. Otherwise, it may get stuck in local optimums or cause premature convergence. Generally, two approaches have been utilized to solve these problems: introducing a novel selection method or tuning parameters of an algorithm that already proposed.

In the first part of this section, common selection methods that have been used in the literature of GAs were presented. After that, the section provides information about which methods are commonly used to tune parameters of an algorithm.

2.2. Literature Review on Genetic Algorithms

Roulette Wheel (RW) is the first selection method for GAs (Holland 1975a). Even if implementation of RW is straight-forward, this method has some disadvantages such as; individuals having high fitness value dominates the population too early, no selection pressure for some cases, etc.

Linear ranking (LR) was introduced in 1989 (Grefenstette and Baker 1989). The idea was to eliminate the main problems of RW, which are premature convergence and selection pressure. Even if LR tries to solve these problems, the method still encounters with some other issues: reducing selection pressure may cause to converge slowly, population must be sorted in each iteration, and unfortunately, it is not suitable for parallel processing since it requires a global ranking for all individuals.

Another well-known selection method is Tournament. Tournament Selection was analyzed, and published by Blickle in 1995 (Blickle and Thiele 1995). The benefits of Tournament Selection can be listed as; having less time complexity, O(n), being suitable for parallel programming, and being capable of adjusting

selection pressure. One disadvantage of Tournament Selection is the possibility of being trapped up in local optima if the fitness function has many local optimums.

Goldberg proposed Boltzmann Tournament Selection which is inspired by Simulated Annealing (SA) that utilizes Boltzmann distribution and the temperature value to converge global optima in optimization problems (Goldberg 1990). Instead of paying attention to the cooling process of SA, he actually used Boltzmann distribution in order to have a population which gradually improves towards the best solution.

For multimodal optimization problems in GAs, Restricted Tournament Selection was presented by Harik (Harik 1995). In this method, he added a window around randomly chosen mates in tournament selection. That window whose size is predefined shows the neighborhood of a selected individual. As in tournament selection, two individuals are randomly chosen, however, they did not mate as usual. Instead, a local search has been carried out within the 'windows' of the chosen individuals. Hence, it is expected that each mate would be improvised if there is a better solution found within the neighborhood.

Matsui introduced Correlative Tournament Selection to protect population diversity (Matsui 1999). While the first parent is chosen by Standard Tournament Selection (ST), the second parent is decided by Correlative Tournament Selection. The algorithm randomly takes some number of individuals from the population. Then, it compares the values of these individuals that obtained by correlation function. An individual is awarded as second parent if the individual has better value than comparing other selected individuals.

De La Maza and Tidor offered Boltzmann Selection (de la Maza and Tidor 1992). The idea of the algorithm is based on the temperature value which controls selection pressure. The algorithm starts with high temperatures to make selection pressure low which gives selection chance not only to the best individuals, but also all the other individuals. Thereafter, it progressively decreases the temperature to increase selection pressure. This procedure facilitates the algorithm to investigate all

search space at the beginning, then it is primarily focused to the better areas which were already found.

Local Selection method was presented by Pohlheim (Pohlheim 1995). The method is based on local neighborhood. In this method, each individual in a population can only mate with an individual that is located in its pre-defined neighborhood. However, it was observed that the method is unfeasible for small populations.

Goh et al. proposed Sexual Selection that influenced by Darwinian approach on sexual selection (Goh et al. 2003). Initially, the algorithm specifies females and males in a population. Females are decided either randomly or according to the specific knowledge of the problem. Moreover, male selection is performed as in ST. The algorithm carries on until all females are mated.

The algorithm of High-Low Fit Selection is initiated by sorting individuals in the population (Ali and Elamin 2006). After that, it divides the population into two sub groups: individuals having high fitness and individuals having low fitness. The method randomly selects two parents from each group, and mates them.

Yalkın and Korkmaz proposed a Neural Network based selection method (Yalkin and Korkmaz 2012). First of all, the algorithm works as a standard GA. While working on, the algorithm collects data for selection mechanism. The process of collecting data is proceeded until appropriate data is obtained. Then, selection method is utilized. While the first mate is chosen by using ST, the second mate is selected by using the Neural Network.

Another method, which is used in standard GAs, is Age Based Replacement (Eiben and Smith 2015). Instead of using better individuals having high fitness value, this method selects individuals according to their ages. The age of an individual means that the total time of the individual maintaining in the population. When a new individual participates to the population, the new individual is replaced with the oldest individual. However, this method may cause to lose the best individuals in the population (Eiben and Smith 2015).

2.3. Literature Review on Meta Optimization

The first paper in meta optimization dates back to 1978 (Mercer and Sampson 1978). Mercer and Sampson employed a GA to find the best parameter settings of crossover and mutation rates for another GA. However, the study was limited since there did not exist a modern computer in those years; therefore, only one single run could be tested in the experiment. A few years later, parameters of GAs, which are crossover rate, mutation rate, population size, generation gap, scaling window and selection strategy, were investigated by using two approaches of parameter tuning (Grefenstette 1986). It was indicated that the online performance is much better against to the offline performance. Freisleben and Hartfelder presented a GA meta optimizer to use for tuning population size of another GA (Freisleben and Härtfelder 1993). As in (Grefenstette 1986), Freisleben and Hartfelder tested many genetic operators. Their results show that regulating parameters during the search process is significantly important in meta optimization.

On the other hand, Back offered an alternative way to tune parameters of GAs (Bäck 1994) where the meta algorithm benefits from “master-slave approaches”. In that approach, the meta algorithm, “master”, dominates some number of workers that parallelly run on different processors, “slaves”. Each slave carries different parameter description of a GA, individuals. At each generation of the master, slaves evaluate individuals for the master. Based on the fitness values of individuals, the master decides its population member.

Unlike the other methods that were mentioned, Hinterdings only adjusted mutation parameter by using Gaussian noise which was obtained from a normally distributed random variable (Hinterding 1995). He claims that permitting an algorithm to set its parameters during run can provide more efficient operation for the algorithm. In another paper, GAs and Simulated Annealing (SA) were applied to discover the best parameter settings of GAs in multi peak problems (Keane 1995). Both methods present better values than the value where default parameters were employed. It was concluded that tuned parameters much more effective than the

default parameter settings. Cortez et al. proposed a meta-GA for parameter estimation purposes in time series forecasting problems (Cortez et al. 2001). In recent years, a basic GA was utilized as an optimizer in order to optimize parameters of a decision tree learner (Camilleri and Neri). The performance of the meta-GA was compared with the performance of Brute Force Search. According to the results in the decision tree learner, it is asserted that SGA offered predominant results comparing to Brute Force Search.

GAs were also employed to choose adequate parameter settings of PSO in scheduling problems (Matrenin and Sekaev 2015). It was achieved that a tuned PSO show higher performance than the PSO that has a standard setting.

In addition to use GAs as meta optimization techniques, PSO has also been applied as a meta optimizer. The first usage of PSO in the literature of meta optimization, called OPSO, was applied to PSO to improve its effectiveness (Meissner et al. 2006). This technique was performed on neural network training to design an appropriate model for predicting blood-brain barrier permeation of small organic molecules. After that, Pedersen and Chipperfield tried to simplify PSO by using PSO as an meta optimizer (Pedersen and Chipperfield 2010). The simplified method was implemented on artificial neural network cancer and card problems, and it produced better performance on several problems. Moreover, they presented a new approach to tune the variants of Differential Evolution (DE) by using Artificial Neural Network (Pedersen and Chipperfield 2008). However, this method did not offer any advantage comparing to standard DE whose parameters are predefined. Thus, they suggested to utilize an online meta optimization technique in order to have better performance on DE. To apply on the watershed management problem, Mason et al. proposed a PSO meta optimizer that optimizes parameters of common velocity update equations (The Attractive Repulsive PSO, The Dissipative PSO, the Adaptive Velocity PSO etc.) in order to understand which features of each benefit the algorithm (Mason et al. 2018). The meta optimizer was used to ensure that the

best parameter configurations were utilized in each velocity update equations; therefore, they can be compared fairly.

Furthermore, several offline methods have been presented: Birattari et al. introduced F-Race algorithm that based on a statistical method (Birattari et al. 2002). Instead of testing all possible parameter settings (candidates), F-Race focuses on the candidates that offer promising results, and disregards others. In that study, an instance of Traveling Salesman Problem was operated to configure the parameter of an Ant Colony Optimization algorithm. A few years later, Balaprakash proposed iterated F-Race algorithm (Balaprakash et al. 2007). The only difference between F-Race and iterated F-Race is that iterated F-Race has a number of iterations which consists of three main steps: construction of candidate solutions, evaluating the candidates, and updating the best candidate solution. The detail explanation of F-Race and iterated F-Race was mentioned in (Birattari et al. 2010).

François and Lavergne proposed a statistical offline method to find the best parameter settings of EAs (François and Lavergne 2001). The method includes two main parts: problem level model and global model. In problem level model part, EAs were tested on a specific problem to determine their behaviors on that problem. On the other hand, in global model, a set of problems were utilized to identify the behaviors of EAs on these problem sets. According to the obtained behaviors, the best parameter settings of EAs were chosen. These parameter settings were applied other set of problems to get best results on those problems. Another new method, CRE (the parameter calibration and relevance estimation), was also operated to EAs to acquire high performance on the agent-based application (Nannen and Eiben 2006). The study emphasized that the evolutionary behaviors of agents cannot make progress by themselves. Instead, they require tuned parameters for their welfare.

Diaz and Laguna presented CALIBRA technique that consists of two mechanisms: experimental designs and local search (Adenso-Díaz and Laguna 2006). The experimental designs assist to explore the promising region of the search space. Thus, it provides better starting point for the initial stage. The aim of local

search stage is to find local optimum value in the area that are found. In order to do this, the method decreases the range of parameter settings to focus on the promising region. However, this technique cannot be employed if an algorithm has more than five parameters. A year later, ParamILS, a local search method, was introduced that offers no limitation on number of parameters (Hutter et al. 2007). Moreover, the performance of ParamILS is better than the performance of CALIBRA. Hence, ParamILS was performed for different purposes (Ansótegui et al. 2009, Hutter et al. 2009b).

Sequential Parameter Optimization (SPO) one of the known statistical approaches is proposed by Bartz et al. (Bartz-Beielstein et al. 2005). In order to analyze efficiency of SPO, they tested the algorithm on three cases: discovering the algorithmic chemistry of GAs, parameter optimization on PSO and EAs. Even if SPO succeed on these three cases, the algorithm needs to be improved due to its disadvantages such as operating only decimal or integer values. Therefore, the improved version of SPO, SPO+, was proposed which presents better performance than SPO (Hutter et al. 2009a). As an alternative of those methods that were introduced, Sequential Model Based Algorithm Configuration, named “SMAC”, and Randomized Online Aggressive Racing, called “ROAR”, were presented (Hutter et al. 2011). The goal of the paper is to solve problems that other sequential model-based optimization techniques get stuck in: limitation on numeric parameters and optimizing performance of the target algorithm for single instance. The new methods, ROAR and SMAC, were compared with other known techniques, e.g. ParamILS, SPO. Their optimization skills on the local search and the tree search solver were tested in the propositional satisfiability problems (SAT). Based on the results, SMAC presents better performance than ROAR.

Branke and Elomari proposed a new method, Flexible Budget offline, to adjust the parameters of EAs: population size and offspring population size (Branke and Elomari 2012). For the meta level, each possible parameter settings of EAs is initially applied itself. According to their results, for each iteration, they are ranked

from 1 to n, the number of parameter settings that are going to be tested. Thus, the algorithm selects parameter configurations with lower ranks for parent selection. Although the new method was operated in a single run, it produced better results comparing to the traditional technique, Fixed Budget Method where the parameters are chosen by comparing their results at the end of the iterations.

Yuan and Gallagher produced a hybrid method of Racing, a statistical method that races algorithms, and meta-EA owing to their advantageous aspects (Yuan and Gallagher 2007). Nevertheless, the hybrid method only works when parameter spaces are large. Moreover, the hybrid algorithm does not offer any achievements comparing to the algorithms that create it, meta-EA and Racing.

Besides of above studies that explained, some researchers have been compared offline and online methods (Smit and Eiben 2009, Pellegrini et al. 2010, Francesca et al. 2011, Aburomman and Ibne Reaz 2016). To summarize these researches, online techniques offer better approximation to the global points than offline techniques. In addition to them, some surveys (Hinterding et al. 1997, Eiben et al. 1999, Meyer-Nieberg and Beyer 2007), and studies (Bartz and Bartz-Beielstein 2003, Erik 2010) have been introduced for those who are interested in meta optimization.

3. MATERIALS AND METHODS

3.1. Genetic Algorithms

The principle of Genetic Algorithms (GAs) was developed by John Holland (Holland 1975). His main idea is to construct a new model for optimization problems. GAs, inspired from Darwin's Evolution Theory, ensure robust search in a search space (Mitchell 1996). The search process of the algorithm, which is stochastic, utilizes a group (population) of solutions (individuals).

Basically, a GA initiates with randomly creating the potential solutions, called initial population. After the initial population is generated, the algorithm proceeds by evaluating fitness values of individuals in the population. Afterwards, its bio-inspired searching steps starts: most importantly selection, recombination, also called crossover, and mutation.

One of the most important operators of GAs is selection. As a common technique, selection operators offer priority to fitter individuals. They tend to choose the individuals whose fitness values are better in the population. Thereby, this operation permits the better individuals to survive for the next generation, which means that selection pressure is high. Most of the selection methods employ this idea such as proportionate methods, ranking, tournament, etc.

The aim of crossover operator is to produce offspring by using two or more individuals which are selected by the selection operator. The crossover operator combines these individuals to create better offspring than their parents (Mitchell 1996). The crossover probability (p_c) decides whether an individual is going to attend the crossover phase, which refers that attending crossover step is not required. If an individual does not want to mate with another individual, then the individual can be directly copied for the next generation.

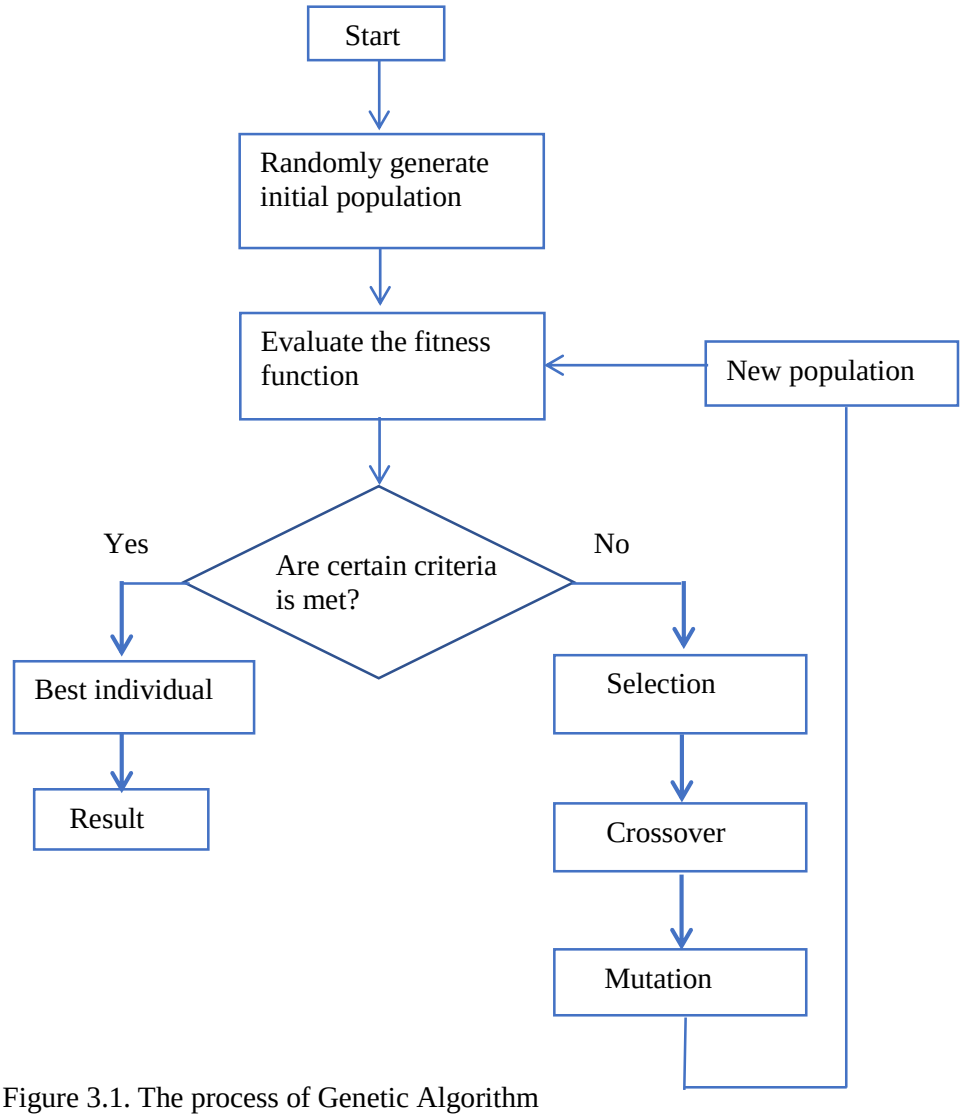


Figure 3.1. The process of Genetic Algorithm

Mutation operator intends to maintain genetic diversity and to prevent being squeezed to the local optima (Mitchell 1996). As in crossover, mutation probability (p_m) decides whether an individual is going to attend the mutation phase. The mutation operator modifies the genetic material of individuals that are chosen according to p_m. In the interval of [0,1], variety of this probability can be utilized

(Shopova and Vaklieva-Bancheva 2006). However, it is recommended to use low values such as between 0.01 and 0.1 (Shopova and Vaklieva-Bancheva 2006).

The next generation of the population is created by combining a proportion of offspring and members of the current population. Then, the algorithm returns to the selection process and repeats the steps until the termination criteria are met; either the optimum solution is found or maximum number of generations are reached. The algorithmic steps of GAs are illustrated in Figure 3.1.

3.2. Selection Methods

The purpose of this section is to give some knowledge about common selection methods in the field of GAs. Standard Tournament, Restricted Tournament, Unbiased Tournament, Fine-Grained Tournament and Cooperative selection methods, which have been the most commonly used, were operated for the comparisons. The remaining selection methods are presented here for the sake of completeness.

3.2.1. Proportionate Selection Methods

Proportionate methods select individuals according to their fitness values. Well-known proportionate methods are Roulette Wheel, Stochastic Sampling, Stochastic Remainder Sampling, Deterministic, and Stochastic Universal Selection:

3.2.1.1. Roulette Wheel

Roulette Wheel (RW) Selection is the simplest proportionate selection method (Goldberg 1989). RW consists of segments whose size equals to the population size. The size of each segment demonstrates the proportion of an individual based on its fitness value. If the fitness value of the individual is better, then it has larger segment on the wheel. The most common way to select individuals on the wheel is generating random number from interval (0,1). The generated number indicates which individual will be chosen. The process is repeated until

essential number of individuals are selected. The advantage of RW is to be implemented easily. On the other hand, one of the most important disadvantages of RW is the individual having high fitness value dominates the population too early, premature convergence, which means that the algorithm might not find the global value.

3.2.1.2. Stochastic Sampling

RW is also called Stochastic Sampling with Replacement (SSR) (Coley 1999) since it permits individuals for multiple selection.

Stochastic Sampling without Replacement (SSWR), also called Stochastic Sampling with Partial Replacement, is introduced by De Jong to solve the problem in SSR. On the contrary of SSR, individuals cannot be chosen multiple times in SSWR.

The process of SSWR starts by evaluating the expected occurrence of each individual:

$$pselected_i = \frac{F_i}{F_{total}} \tag{1}$$

$$e_i = pselected_i * n \tag{2}$$

where $\mathrm{pselected_i}$ is probability of selected individual (*i*), $\mathrm{F_i}$ is the fitness of *i*, $\mathrm{F_{total}}$ is total of all fitness values, $\mathrm{e_i}$ is the expected occurrence of *i*, and n is the population size.

The algorithm works as in SSR. The only difference is that SSWR is employed the integer part of the expected occurrence as the offspring count. If individual *i* prefers to be in the next generation, willing to not attend crossover phase, then its offspring count is decreased by 1. If the individual is selected for crossover, the offspring count is diminished by 0.5. When the offspring count becomes negative

or zero, then the value is set to 0. Thus, the individual cannot be selected anymore. However, this procedure may cause to lose the best individual in the population (Coley 1999).

3.2.1.3. Stochastic Remainder Sampling

Stochastic Remainder Sampling with Replacement (SRSR) divides the fitness value of individuals to average fitness value to obtain probability of selected individual.

$$pselected_i = \frac{F_i}{F_{average}} \tag{3}$$

The integer part of $\mathrm{pselected_i}$ demonstrates how many times individual *i* needs to be occurred in the mating pool. For instance, if $\mathrm{pselected_i}$ is 4.2, individual *i* occurs 4 times as a parent in the mating pool. Afterwards, each free space in the mating pool is filled by using RW until the mating pool is full.

On the other hand, Stochastic Remainder Sampling without Replacement (SRSWR) initially behaves as SRSR. However, SRSWR fills the mating pool by using a “weighted coin toss” (Goldberg 1989). Based on the fractional components of the expected occurrences, random numbers are generated. If the generated number is less than the fractional component of expected occurrence, then the individual can once again take place in the mating pool.

3.2.1.4. Deterministic Selection

As in Stochastic Sampling, expected occurrences are computed by using Equation (1) and (2). The integer parts of expected occurrences represent how many times individuals occur in the mating pool as in Stochastic Remainder Samplings. The difference occurs in the process of completing the mating pool. According to the fitness values, individuals in the population are sorted in a list. Starting from the

top of the list, each free space in the mating pool is filled until the mating pool is full.

3.2.1.5. Stochastic Universal Sampling

Stochastic Universal Sampling (SUS) is an improved version of fitness proportionate (Baker 1987). The method initiates as RW. Then, pointers with equal intervals are placed along the outside of the wheel. These pointers indicate which individuals need to be selected. This process is repeated until the mating pool is full.

3.2.2. Ranking

Ranking overcomes the problems where other methods, mentioned before, get stuck such as premature convergence and selection pressure. It is also claimed that Ranking Selection is more robust comparing the other methods (Grefenstette and Baker 1989). However, the method still has some problems such that reducing selection pressure may cause to converge slowly and sorting the population in each iteration is time consuming. Linear Ranking, Exponential Ranking and Truncation selection are commonly used methods in ranking.

3.2.2.1. Linear Ranking

Linear Ranking (LR) initiates by sorting individuals according to their fitness values. Thereafter, starting from the best individual to the worst individual, they are ranked from n, the population size, to 1. The probability of the individual is obtained by dividing the rank value of the individual to the total rank value. (Grefenstette and Baker 1989). According to this probability, the individual can be selected as a parent.

3.2.2.2. Exponential Ranking

Exponential Ranking, also called Non-Linear Ranking, initially sorts individuals from worst to best according to their fitness value. Then, the selection probabilities are calculated exponentially (Blickle and Thiele 1995):

$$pselected_i = \frac{c-1}{c^n-1} c^{n-i} \tag{4}$$

where c is in the interval (0,1) and n is the population size. When c is increased, the exponentiality of the method would be diminished.

3.2.2.3. Truncation Selection

As in other ranking selection methods, Truncation selection also initiates by sorting individuals in the population according to their fitness values. Only T, the truncation threshold value, best individuals are selected to produce offspring. Equal probability is assigned to T selected individuals for the selection phase. Truncation selection is commonly used by breeders. Moreover, it is also utilized in Muhlenbein's breeder genetic algorithm (Blickle and Thiele 1995).

3.2.3. Tournament

Tournament selection was introduced by Goldberg (Goldberg 1989). The algorithm starts by randomly choosing a number of individuals from the population. The probability *p* is assigned to the best individual, having high fitness value (Blickle and Thiele 1995). Other selected individuals gradually take less probability than the best individual based on the probability *p*, e.g. the probability of the second individual can be $p.(1-p)$, and the probability of the third individual can be $p.(1-p)^2$ etc. Then, the competition occurs among these selected individuals in order to choose a parent.

In the deterministic way that are called Standard Tournament Selection (ST), which has been widely employed, the algorithm selects some number of individuals from the population; however, only the best individual has chance to attend crossover phase, i.e. $p = 1$.

In 1995, Blickle analyzed ST (Blickle and Thiele 1995). Its benefits, which were mentioned in that paper, are having less time complexity, O(n), being suitable for parallel programming, and being capable of adjusting selection pressure. Thus, ST has been applied due to its advantages. Multi-selection and not sampling, an individual that is not chosen for the tournament, are the main problem of ST. Another disadvantage of the tournament can be that it is possible to squeeze in local optima when the fitness function has many local optimums.

Some known selection methods that are based on ST are given below:

3.2.3.1. Boltzmann Tournament Selection

Boltzmann Tournament Selection was presented by Goldberg in 1990 (Goldberg 1990). He originated Simulated Annealing (SA) which uses Boltzmann distribution and the temperature value. In optimization problems, SA operates the cooling procedure to converge global optima. He designed the process of SA for GAs to have a population which gradually improves towards the best solution. In this algorithm, first individual is taken randomly, then, second individual whose fitness value is different than the first one's by the threshold value, which is 0.5, is chosen. The third individual is selected in two manner: In the *strict* way, the third individual whose fitness value is different than the first one's and the second one's by the threshold value is taken. On the other way, in the *relax* choice, the third individual whose fitness value is different than the first one's by the threshold value is selected.

3.2.3.2. Restricted Tournament Selection

Harik designed a new method called Restricted Tournament Selection for multimodal optimization problems in GAs (Harik 1995). Even if ST has many advantages, it is not efficient for multimodal problems. Hence, he modified ST by employing a window around the randomly chosen mates. That window specifies the neighborhood of a selected individuals, and its size is pre-defined. Initially, two individuals are randomly chosen for mating. Nevertheless, rather than mating straight forward as in ST, a local search has been carried out within the 'windows' of the chosen individuals. By this way, each mate would be improved if there is a better solution found within the neighborhood. In order to increase the effectiveness of the algorithm, some other techniques are also introduced (Roy and Parmee 1996, Lobo and Bazargani 2015, Farias and De Magalhães 2018).

3.2.3.3. Correlative Tournament Selection

Matsui presented Correlative Tournament Selection (Matsui 1999) to protect population diversity. The first mate is chosen by ST; however, the second mate is selected by Correlative Tournament Selection. For the second mate, the algorithm randomly takes a number of individuals from the population as in ST. The difference is that the values of these individuals are obtained by correlation function, Hamming distance, instead of a fitness function. If an individual has better value than other selected individuals, the individual is accepted as a second parent.

3.2.3.4. Fine-Grained Tournament Selection

Fine-Grained Tournament Selection (FGTS) was introduced by Filipovic (Filipović 2003). The only difference between FGTS with ST is that FGTS has a flexible tournament size value instead of a constant value. It means variety of the tournament size can be operated during the selection procedure. The reason for employing the flexible tournament size is to create a balance between exploration

and the exploitation. Although the tournament size is mostly taken as 2, larger values can also be utilized to increase the selection pressure.

3.2.3.5. Unbiased Tournament Selection

Sokolov and Whitley offered Unbiased Tournament Selections (Sokolov et al. 2005). These methods work as ST. However, for each individual, they also save information about participating to the recombination step, called objective function value. Thus, an individual with higher fitness and lower objective function value is selected for the recombination.

3.2.3.6. Cooperative Tournament Selection

Laredo et.al, on the other hand, presented a new selection method called Cooperative Selection (Jiménez Laredo et al. 2014). This new method solves the problems of ST that directly affect the loss of the genetic diversity: not sampling and multi-selection. The method operates large population sizes to avoid the not sampling problem. On the other hand, its altruistic behavior assists to solve the multi-selection problem. As in ST, a number of individuals, called association size, are selected from the population. According to their cooperative fitness values, the rank values are given to the chosen individuals. These values aid to decide who the winner of the computation is. When the winner of the computation is determined, the method utilizes altruistic behavior, which makes the cooperative fitness value of the winner as a half of the totals of fitness values of the winner's neighbors.

3.2.4. Steady State Genetic Algorithm

Steady State Genetic Algorithm randomly chooses a number of individuals from the population (Davis and Mitchell 1991). After these individuals produce offspring, the algorithm compares the fitness values of the offspring with their parents to decide who is going to survive, offspring or the parents. Then, it saves

other individuals for the next generation. The advantage of the method is creating a new generation on the same population where the parents are also selected. Hence, it offers a substantial reduction in the memory requirements of a system.

3.2.5. Elitism

After the main operations of GAs end, a GA may be encountered with the problem that the best individuals are missed from the population. Thus, elitism method protects these precious individuals in order to retain them for the next generation.

3.2.6. Boltzmann Selection

Boltzmann Selection was described by De la Maza (de la Maza and Tidor 1992). The idea of the algorithm is based on the temperature value which controls selection pressure. The algorithm starts with high temperatures to make selection pressure low which gives selection chance not only to the best individuals, but also all the other individuals. Thereafter, it progressively decreases the temperature to increase selection pressure. This procedure facilitates the algorithm to investigate all search space at the beginning, then it is primarily focused to the better areas which were already found.

3.2.7. Fitness Sharing Algorithm

Deb and Goldberg introduced Fitness Sharing Algorithm (Deb and Goldberg 1989). As usual, individuals are randomly chosen from the population. Their phenotype and genotype distances are measured by respectively utilizing “Euclidean Distance” and “Hamming Distance”. If the phenotype or genotype distances are smaller than the predefined threshold value, then the chosen individuals are mated. Otherwise, the process is repeated until the parents that satisfy the desired condition are found. Eshelman and Schaffer was also used the same idea to present Incest Prevention Algorithm (Eshelman and Schaffer 1991). Conversely, in this method,

the phenotype or genotype distances of individuals have to be greater than threshold value for the mating procedure. Besides, threshold value can be reduced if better offspring was not found.

3.2.8. Local Selection

In this method, individuals in the population belong to a constrained environment called the local neighborhood. The neighborhood is defined by the structure where the population is distributed (Pohlheim 1995). Local Selection allows individuals to mate only with the neighbors. Initially, by using any selection method, the algorithm chooses a number of individuals from the population. For each selected individual, its local neighborhood is identified. Then, every selected individual chooses a partner from this neighborhood.

3.2.9. Age Based Selection

Age Based Replacement is based on the age value instead of depending on the fitness value. The age of an individual in the population represents the number of existences of the individual during its life cycle. After offspring are created, they are replaced with the oldest individuals in the population. However, this strategy more likely causes to lose the best individuals in the population because of FIFO strategy (Eiben and Smith 2015). Therefore, it is not a common in GAs.

3.2.10. High Low Fit Selection

High Low Fit Selection starts the process by sorting individuals in the population (Ali and Elamin 2006). Then, the algorithm divides the population to two parts: High Fit (HF) where individuals having high fitness are in, and Low Fit (LF) which includes individuals having low fitness value. The algorithm selects first mate from HF, and second mate from LF.

3.2.11. Neural Network Based Selection

Neural Network-based selection was introduced by Yalkın and Korkmaz (Yalkin and Korkmaz 2012). Initially, while standard GA works, the algorithm collects data for Artificial Neural Network (ANN). Data collection is maintained until the appropriate data is taken. After the data collection, the algorithm chooses the first mate via ST while the second mate is selected according to the collected data by operating ANN.

3.2.12. Sexual Selection

Sexual Selection is a selection method that influenced by Darwinian approach (Goh et al. 2003). The algorithm initiates its process by identifying females and males in the population. Females are specified, either randomly or according to the specific knowledge of the problem, and the rests are accepted as males. The male selection is performed as in ST. The algorithm proceeds until all females are mated.

3.3. Meta Optimization

Parameter selection affects how an evolutionary algorithm behaves; therefore, it is significantly important to choose efficient parameters such as crossover rate, mutation rate, tournament size, population size etc. Many researchers have generally tuned these parameters by utilizing “default parameter settings”, crossover rate = 0.7, mutation rate = 0.05, tournament size = 3 (Lobo and Goldberg 2004). Nevertheless, this technique might not offer desired solutions for all problem types. For example, selecting a decent parameter configuration can find the optimum value in one of the problem instances; however, that parameter configuration may not satisfy other problem instances. Hence, it can cause premature convergence or it gets stuck in one local optima. Two common approaches have been a remedy for that trouble: hierarchical search techniques and meta optimization, see Figure 3.2.

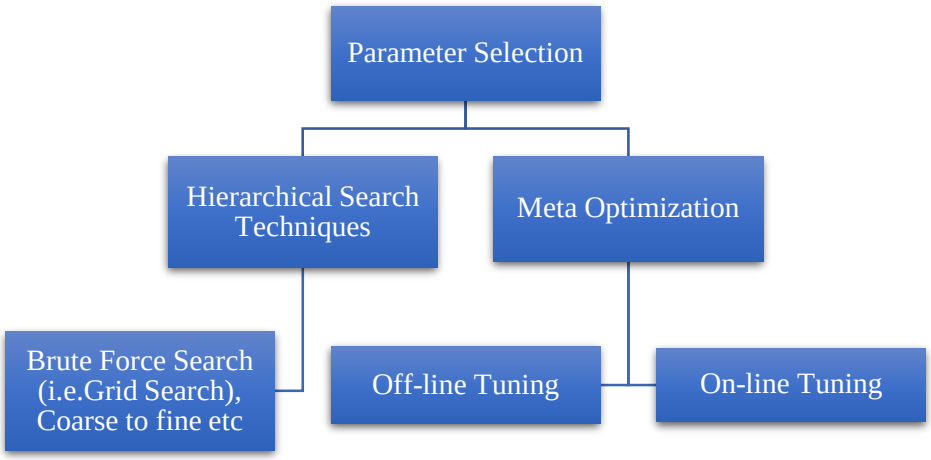


Figure 3.2. Approaches for parameter tuning

Brute force search is a traditional technique that tests all available parameter configurations of an algorithm to explore optimal parameter settings of the algorithm. It can be adequate when only few parameters are necessary. Otherwise, it causes large computational budget. Due to small computational time requirement, researchers have recently operated meta optimization techniques to find the optimal parameter settings of an optimization algorithm.

Meta optimization used in optimization problems refers that improving performance of an algorithm by tuning parameters of the algorithm, Figure 3.3 clarifies the process of meta optimization. Besides improving efficiency of the algorithm, it also assists us to understand how the algorithm behaves in different types of problems. Meta optimization techniques can be classified in two main groups: off-line and on-line.

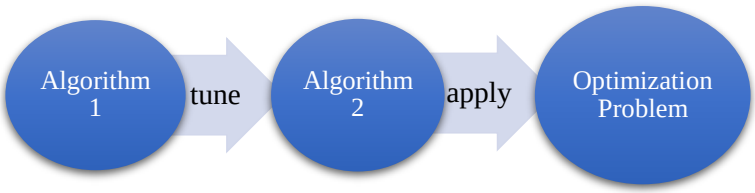


Figure 3.3. The process of meta optimization

In off-line tuning methods, parameter settings are identified before the algorithm starts. Initially, an off-line method analyzes some number of parameter configurations on a training set of instances. Then, the chosen configuration is applied to all instances that are struggled. This technique works well if selected instances have equal structure to other instances in the training set. However, it might not work when the class of instances is more heterogeneous. Since the instances have diversity structures, choosing appropriate parameter settings takes large computational time.

On-line tuning methods adjust parameter configuration of an algorithm while the algorithm is working to solve one of the problem instances. They collect feedbacks during the process of the algorithm to decide the best parameter configuration for that instance. This principle is applied to all problem instances to find suitable parameter configuration of each instances. Hence, it is obvious to notice that on-line methods provide saving the computational time while off-line tuning methods takes more time.

3.4. The Methods of Meta Optimizations

The methods that are used for optimization can be employed for meta optimization as well. In this section, Coarse to Fine Search, GA and Particle Swarm Optimization algorithms will be introduced.

3.4.1. Coarse to Fine Search

Brute force search is accomplished if only few parameters are needed to be tuned. However, when number of parameters is increased, it starts to take large computational budget. In that case, Coarse to Fine Search can be employed to overcome the problem of computational budget. Instead of operating all possible parameter configurations, this technique randomly selects some number of parameter configurations, and identifies best configurations in the selected parameter settings (Staelin 2003) (see Figure 3.4. for the example of two parameters). Afterwards, it focuses on the area in the search space where the best configurations are found, shown in Figure 3.5. Once again, it randomly chooses some number of configurations from the area to explore whether a better configuration occurs in there. If the better configuration is discovered, then the best configurations are updated.

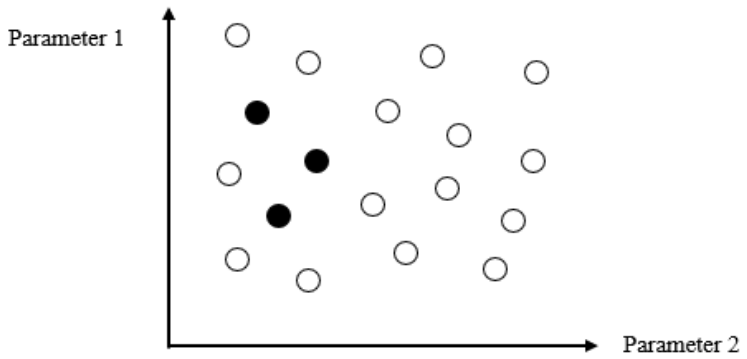


Figure 3.4. Specifying better configurations in randomly selected parameters

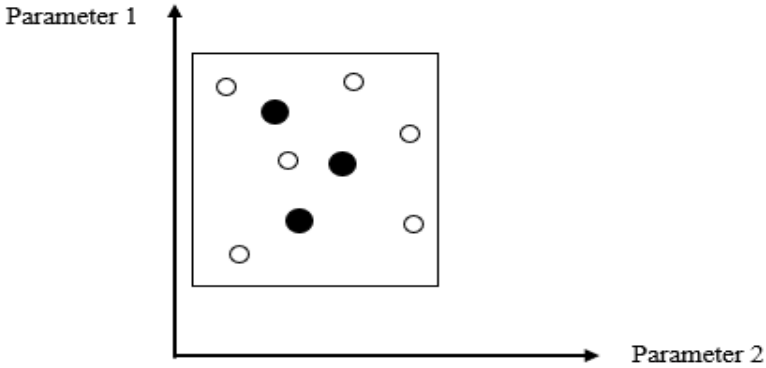


Figure 3.5. Check the neighborhood of the best configurations

3.4.2. Genetic Algorithms

For this section, it is required to inform that a standard GA, where Standard Tournament Selection (ST) is utilized, is employed as a meta optimizer.

3.4.3. Particle Swarm Optimization (PSO)

The origin of Particle Swarm Optimization (PSO) was introduced by Eberhart and Kennedy (Eberhart and Kennedy 1995). Essentially, the algorithm was designed to simulate social behavior of bird flock and fish swarm. However, it was comprehended that PSO actually performs optimization.

As in GAs, PSO initiates its process by creating initial population, “swarm”, with random solutions. Unlike to GAs, each potential solution, “particle”, also includes a velocity to move around in the search space. The movement of the particles are based on their best positions, which have already stored as a *pbest*, and the best position that the swarm found, *gbest*. If a particle explores a better position, then, according to the discovered position, *pbest*, *gbest* or both can be improved. The process is repeated until the desired criteria are achieved. The algorithm works as follows:

Create swarm with random particles
Calculate the fitness values of the particles
Create initial velocity for each particle, pbest
While iteration time
    Find gbest in the swarm
    Update pbest and gbest, if applicable
    Obtain next generation
End

3.4.4. Bipolar Mating Tendency

Bipolar Mating Tendency, which explained in the next section, is utilized as a meta optimizer to examine its performance on the area of meta optimization.

3.5. Proposed Methods

3.5.1. Introduction

Instead of creating a complicated selection method, can a simple method solve the problems that GAs generally get stuck in? This question sheds light the way of reaching Bipolar Mating Tendency (BMT) selection method. However, before achieving the method, the number of trial algorithms were examined:

The first selection method is Aggressive Selection Method (ASM) (Oral and Gençal 2016). The method can only permit mating procedure if an individual finds another individual having better or equal fitness value comparing to individual itself. Otherwise, it rejects the mating and starts from the beginning. However, besides losing time and its rigid mood, this method does not offer the results that desired to have; therefore, it was required to be improved.

Due to advantageous of ST, it is operated with ASM in order to have a hybrid version, Integrated Aggressive Selection Method (Oral and Gençal 2016). Even if it gives better results comparing to ASM, it still has some problems, especially in rejection procedure.

What if the structure of ASM is changed, would it overcome the problems that were encountered before? Thanks to this question, the third selection method, Non-Aggressive Selection Method (NASM), was presented in (Oral and Gençal 2017a). Furthermore, in the same paper, ST was also performed with NASM to obtain the integrated version, Integrated Non-Aggressive Selection Method. Even though Non-Aggressive versions precisely show better performance comparing to Aggressive versions, they still spend time for searching appropriate mates due to the rejection step. Thus, Outlander Algorithms were proposed.

Based on the previous question, ASM was altered again. Nevertheless, in this time, the rejection procedure did not put in the process to create an algorithm, Outlander Algorithm (OA) (Gençal and Oral 2017). Instead, the obligated mechanism, which is the first mate has to couple with an individual not in the population when it rejects the second mate, was performed. Even though OA provides more exploration, the obligated mechanism should have been smoothed for the exploitation. That is why, Non-Aggressive Outlander Algorithm was introduced (Oral and Gençal 2018). It offers precious results comparing to other proposed methods; however, it is not accomplished for large population.

In Outlander Algorithms, the idea of mating two individuals from different populations proves that even if unfavorable individuals are in the mating process, appropriate results can also be achieved. This idea leads the thought that choosing the less favorable individuals instead of the favorable ones. Hence, the origin of BMT was arose.

3.5.2. Aggressive Selection Method

In Aggressive Selection Method (ASM), an individual can only mate with another individual having better or equal fitness value comparing to individual itself (Oral and Gençal 2016). The algorithm works as follows for a minimization problem;

count = 1

```
while count <= population size
        Randomly choose first and second mate
        Determine costs of first and second mate
        If cost (second mate) <= cost (first mate)
                Allow mating
                count = count+1;
        Else
                Reject mating
        end
end
```

3.5.3. Integrated Aggressive Selection Method

Integrated Aggressive Selection method (IASM) is a hybrid version of ASM and Tournament (Oral and Gençal 2016). In IASM, the first mate is selected by ST, and the second mate is chosen by ASM. For a minimization problem, the pseudo code of IASM is given below:

```
count = 1
while count <= population size
        Choose first mate via tournament
        Randomly choose the second mate
        Determine costs of first and second mate
        If cost (second mate) <= cost (first mate)
                Allow mating
                count = count+1;
        Else
                Reject mating
        end
end
```

The main idea of ASM and IASM is that permitting only the better individuals to attend crossover step in order to improve population progressively. For instance, it is assumed that the fitness values of four individuals in a population are respectively A, B, C and D where D < C < B < A. In a minimization problem, it is clear that D is the best individual in the population. Table 3.1 demonstrates that the worst individuals can only mate with better individuals while D can mate with all members of the population. Hence, it is apparently figured out that D has more chance than others for mating.

Table 3.1. Mating permissions of individuals in a population

	Second Mate			
First Mate	*1*	*2*	*3*	*4*
1	***Allowed***	***Allowed***	***Allowed***	***Allowed***
2	***Rejected***	***Allowed***	***Allowed***	***Allowed***
3	***Rejected***	***Rejected***	***Allowed***	***Allowed***
4	***Rejected***	***Rejected***	***Rejected***	***Allowed***

Because of the rejection mechanism, ASM and IASM can cause increasing the computational cost. Thus, NASM and INASM are respectively improved versions of ASM and IASM were proposed to moderate the rigid selection scheme of ASM and IASM (Oral and Gençal 2016).

3.5.4. Non-Aggressive Selection Method

Non-Aggressive Selection Method (NASM) was presented by developing ASM (Oral and Gençal 2017a). The only difference between ASM and NASM is that the second mate can be selected even if it does not have better fitness value than the first mate has. The thought of adjusting selection pressure which also inspired the next proposed methods is first time applied in this method. Algorithmic steps of the proposed method can be described as follows for a minimization problem:

```
count = 1
while count <= population size
        Randomly choose first and second mate
        Determine costs of first and second mate
        Calculate the relaxation probability (p_r)
        If cost (second mate) <= cost (first mate)
                Allow mating
                count = count+1;
        Else
                If random value < p_r
                        Allow mating
                        count = count+1;
                Else
                        Reject mating
                end
        end
end
```

The relaxation probability (p_r), which is inspired from Boltzmann probability, can be computed by employing the Equation (5) and (6):

$$p_r = \frac{1}{1+exp^{\frac{\Delta}{T}}} \tag{5}$$

$$\Delta = cost(second) - cost(first) \tag{6}$$

where T is the temperature value.

3.5.5. Integrated Non-Aggressive Selection Method

Integrated Non-Aggressive Selection method (INASM), is a hybrid version of Tournament and NASM selection methods (Oral and Gençal 2017a). In INASM, the first mate is chosen by using ST, and the second mate is selected by IASM. For a minimization problem, the algorithm works as follows:

```
count = 1
while count <= population size
    Choose first mate via tournament
    Randomly choose second mate
    Determine costs of first and second mate
    Calculate the relaxation probability (p_r)
    If cost (second mate) <= cost (first mate)
        Allow mating
        count = count+1;
    Else
        If random value < p_r
            Allow mating
            count = count+1;
        Else
            Reject mating
        end
    end
end
```

3.5.6. Outlander Algorithm

Even if IASM offers better results, it still can cause increasing the processing time due to the rejecting to mate. To solve the rejection problem, Outlander Algorithm (OA) was proposed (Gençal and Oral 2017). OA selects the first mate by

using ST as in IASM. However, when the first mate rejects the second mate, a matchmaker obligates the first mate to couple with an “outlander” that is not in the population and randomly created just for the current mating process. The cost of outlander is trivial for mating and the decision of the matchmaker is indisputable. The pseudo-code of OA for a minimization problem is;

```
count = 1
while count <= population size
        Choose first mate via tournament
        Randomly choose second mate
        Determine costs of first and second mate
        If cost (second mate) <= cost (first mate)
                Second mate is chosen
        Else
                Create a new individual as the second mate
        end
        Allow mating
        count = count+1;
end
```

3.5.7. Non-Aggressive Outlander Algorithm

The diversity of the population has increased through the Outlander Algorithm (OA) and better results have been obtained from studies presented above. By integrating the temperature regime into OA, inappropriate mates have also contributed to gene transfer, and genetic diversity has been enriched by adding in and out populations. However, in order to achieve an even better selection method, the second choice of individuals in the OA was changed by inspiring from INASM, and a new method of Non-Aggressive Outlander Algorithm (N-OA) was introduced

(Oral and Gençal 2018). The N-OA is mainly based on INASM while OA is IASM-based. For a minimization problem, the principle of NO-A is as follows:

```
count = 1
while count <= population size
        Choose first mate via tournament
        Randomly choose second mate
        Determine costs of first and second mate
        Calculate the relaxation probability (p_r)
        If cost (second mate) <= cost (first mate)
                Allow mating
        Else
                If random value < p_r
                        Allow mating
                Else
                        Create a new individual as the second mate
                        Allow mating
                end
        end
        count = count+1;
end
```

3.5.8. Bipolar Mating Tendency

Bipolar Mating Tendency (BMT) is based on ST (Oral and Gençal 2017b). As in ST, BMT randomly selects two groups of individuals from the population. The first mate is chosen as usual in standard tournament selection. However, the individual who is going to be chosen as a second mate is ambiguous. Depending on the first mate's psychological state that is bipolar, the second mate can be either the

best individual or the worst individual from the group members. Pseudocode of the method for a minimization problem is given below:

```
count = 1
while count <= population size
        Choose the first mate via standard Tournament
        Randomly choose the candidates for the second mate
        Calculate the best and the worst from the candidates
        If random value <= bipolarity value
                The second mate is the best one
        Else
                The second mate is the worst one
        end
        Allow mating
        count = count+1;
end
```

The bipolarity value is an empirical value, which is found by testing it from 0.05 to 1 with 0.05 steps, whose results are presented in Appendix 1. Based on the tests, 0.25 is identified as providing the best performance comparing to other values. Thus, the bipolarity value is taken as **0.25**.

3.6. Tests

3.6.1. Introduction

This section consists of two parts: In the first part, proposed methods were compared with Standard Tournament Selection (ST) and prevalent selection methods that are also based on ST: Restricted Tournament Selection (RTS), Unbiased Tournament Selection (UTS), Fine-Grained Tournament Selection (FGTS) and Cooperative Selection (CS).

Twenty-one well known test functions in the field of GAs were employed for the comparison. Furthermore, non-parametric statistical tests, Friedman and Wilcoxon Signed Rank, were applied to demonstrate the significance of the results.

In the second part, meta optimization methods are used to find the most suitable parameter settings for a standard GA. Due to its performance in the first part of tests, BMT was applied as a meta optimizer and the results were compared with the standard GA, Particle Swarm Optimization (PSO) and Coarse to Fine Search.

3.6.2. Test Functions (Benchmarks)

Test functions which are continuous have various classes (Molga and Smutnicki 2005, Surjanovic and Bingham 2013). Unimodal class consists of sensitive functions that accomplish to converge slowly to single global extremum. Multimodal class includes functions having more than one local extremum (Molga and Smutnicki 2005).

Table 3.2. Benchmarks with their classes and global values

Function	Definition	Class	Global Value
f_1	Ackley	Multimodal	0
f_2	Axis Parallel Hyper-Ellipsoid	Unimodal	0
f_3	Booth	Unimodal	0.397887
f_4	Branins	Multimodal	0
f_5	Sixth Function of Bukin	Multimodal	0
f_6	De Jong	Unimodal	0
f_7	Fifth De Jong	Multimodal	Many (One of them is 0.998)
f_8	Drop wave	Multimodal	0
f_9	Easom	Unimodal	-1
f_{10}	Goldstein-Price	Unimodal	0
f_{11}	Griewangk	Multimodal	3
f_{12}	Holder Table	Multimodal	-19.2085
f_{13}	Langermann	Multimodal	Many

f_{14}	Michalewicz	Multimodal	- 1.8013034
f_{15}	Rastrigin	Multimodal	0
f_{16}	Rosenbrock's Valley	Unimodal	0
f_{17}	Rotated Hyper-Ellipsoid	Unimodal	0
f_{18}	Shubert	Multimodal	Many (One of them is -210.48)
f_{19}	Schwefel	Multimodal	-837.9658
f_{20}	Six Hump Camel Back	Multimodal	-1
f_{21}	Sum of Different Powers	Unimodal	0

In order to investigate the quality of the proposed method, benchmarks that commonly used by the researchers in the field of evolutionary algorithms were utilized. Table 3.2 demonstrates the corresponding benchmarks and their classes, and figures of the test functions are displayed in Appendix 2.

3.6.2.1. Ackley's Function

$$f_1 = -a.exp\left(-b.\sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)\right) + a + \exp(1) \quad (7)$$

a = 20, b = 0.2, c = 2π is recommended to be set (Molga and Smutnicki 2005). Test area is usually restricted to -32.768 $\leq x_i \leq$ 32.768, $i = 1, \dots, n$. Its global minimum f_1= 0 is obtainable for $x_i = 0, i = 1, \dots, n$.

3.6.2.2. Axis Parallel Hyper-Ellipsoid Function

$$f_2 = \sum_{i=1}^{n} i.{x_i}^2 \quad (8)$$

Test area is usually restricted to -5.12 $\leq x_i \leq$ 5.12, $i = 1, \ldots, n$. Its global minimum f_2 = 0 is obtainable for $x_i = 0, i = 1, \ldots, n$.

3.6.2.3. Booth's Function

$$f_3 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \tag{9}$$

Test area is usually restricted to -10 $\leq x_i \leq$ 10, $i = 1, 2$. Its global minimum f_3 = 0 is obtainable for $x = (1,3)$.

3.6.2.4. Branins's Function

$$f_4 = a(x_2 - b{x_1}^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + e \tag{10}$$

a = 20, b = 0.2, c = 2π is recommended to be set (Molga and Smutnicki 2005). Global minimum f_4 = 0,397887 is obtainable for three points: (-π, 12.275), (π, 2.275), (9.42478, 2.475).

3.6.2.5. Sixth Function of Bukin

$$f_5 = 100. - b.\sqrt{|x_2 - 0.001{x_1}^2|} + 0.01|x_1 + 10| \tag{11}$$

The function has many local minimum points. Test area is usually restricted to -15 $\leq x_1 \leq$ -5, -3 $\leq x_2 \leq$ 3.

3.6.2.6. De Jong's Function

$$f_6 = \sum_{i=1}^{n} {x_i}^2 \tag{12}$$

Test area is usually restricted to -5.12 $\leq x_i \leq$ 5.12, $i = 1, \dots, n$. Its global minimum f_6 = 0 is obtainable for $x_i = 0, i = 1, \dots, n$.

3.6.2.7. Fifth Function of De Jong

$$f_7 = \{0.002 + \sum_{i=-2}^{2} \sum_{j=-2}^{2} [5(i+2) + j + 3 + (x_1 - 16j)^6 + (x_2 - 16i)^6]^{-1}\}^{-1} \quad (13)$$

Test area is usually restricted to -65.536 $\leq x_1 \leq$ 65.536, -65.536 $\leq x_2 \leq$ 65.536.

3.6.2.8. Drop Wave Function

$$f_8 = -\frac{1+\cos(12\sqrt{{x_1}^2+{x_2}^2})}{\frac{1}{2}({x_1}^2+{x_2}^2)+2} \quad (14)$$

Test area is usually restricted to -5.12 $\leq x_1 \leq$ 5.12, -5.12 $\leq x_2 \leq$ 5.12. $f_8 = -1$ is obtainable at the point (0,0).

3.6.2.9. Easom's Function

$$f_9 = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) \quad (15)$$

Test area is usually restricted to -100 $\leq x_i \leq$ 100, $i = 1, 2$ f_9 = -1 is obtainable at the point (π,π).

3.6.2.10. Goldstein-Price's Function

$$f_{10} = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3{x_1}^2 - 14x_2 + 6x_1x_2 + 3{x_2}^2)].[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12{x_1}^2 + 48x_2 - 36x_1x_2 + 27{x_2}^2)] \quad (16)$$

Test area is usually restricted to -2 $\leq x_1 \leq$ 2, -2 $\leq x_2 \leq$ 2. f_{10} = 3 is obtainable at the point (0,-1).

3.6.2.11. Griewangk's Function

$$f_{11} = \frac{1}{4000}\sum_{i=1}^{n} {x_i}^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (17)$$

Test area is usually restricted to -600 $\leq x_i \leq$ 600, $i = 1, \dots, n$. Its global minimum f_{11} = 0 is obtainable for $x_i = 0, i = 1, \dots, n$.

3.6.2.12. Holder Table Function

$$f_{12} = -\left|\sin(x_1)\cos(x_2)\exp\left(\left|1 - \frac{\sqrt{{x_1}^2 + {x_2}^2}}{\pi}\right|\right)\right| \quad (18)$$

Test area is usually restricted to -100 $\leq x_i \leq$ 100, $i = 1, 2$ f_{12} = -19.2085 is obtainable at the points (8.05502, 9.66459), (8.05502, -9.66459), (-8.05502, 9.66459), (-8.05502, -9.66459).

3.6.2.13. Langermann's Function

$$f_{13} = \sum_{i=1}^{m} c_i \exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}\left(x_j - a_{ij}\right)^2\right)\cos\left(\pi \sum_{j=1}^{n}(x_j - a_{ij})^2\right) \quad (19)$$

where c_i $(i = 1, \dots, m)$ and a_{ij} $(i = 1, \dots, m; j = 1, \dots, n)$ are constant numbers. Local minimums are unevenly distributed. Also, it is recommended to set m = 5 (Molga and Smutnicki 2005).

3.6.2.14. Michalewicz's Function

$$f_{14} = -\sum_{i=1}^{n} \sin(x_i)\, [\sin\left(\frac{i.{x_i}^2}{\pi}\right)]^{2m} \tag{20}$$

Test area is usually restricted to $0 \leq x_i \leq \pi$, $i = 1, \dots, n$. Global minimums are f_{14} = - 4.687 for n = 5, f_{14} = - 9.66 is for n = 10.

3.6.2.15. Rastrigin's Function

$$f_{15} = 10n + \sum_{i=1}^{n}({x_i}^2 - 10\cos(2\pi x_i) \tag{21}$$

Test area is usually restricted to -5.12 $\leq x_i \leq$ 5.12, $i = 1, \dots, n$. Its global minimum f_{14} = 0 is obtainable for $x_i = 0, i = 1, \dots, n$.

3.6.2.16. Rosenbrock's Valley

$$f_{16} = \sum_{i=1}^{n-1}[100(x_{i+1} - {x_i}^2)^2 + (1 - x_i)^2] \tag{22}$$

Test area is usually restricted to -2.048 $\leq x_i \leq$ 2.048, $i = 1, \dots, n$. Its global minimum f_{16} =0 is obtainable for $x_i = 0, i = 1, \dots, n$.

3.6.2.17. Rotated Hyper-Ellipsoid Function

$$f_{17} = \sum_{i=1}^{n} \sum_{j=1}^{i} {x_j}^2 \tag{23}$$

Test area is usually restricted to -65.536 $\leq x_i \leq$ 65.536, $i = 1, \dots, n$. Its global minimum $f_{13} = 0$ obtainable for $x_i = 0, i = 1, \dots, n$.

3.6.2.18. Shubert's Function

$$f_{18} = -\sum_{i=1}^{5} icos((i+1)x_1 + 1) \sum_{i=1}^{5} icos((i+1)x_2 + 1) \tag{24}$$

Test area is usually restricted to -5.12 $\leq x_1 \leq$ 5.12, -5.12 $\leq x_2 \leq$ 5.12.

3.6.2.19. Schwefel's Function

$$f_{19} = \sum_{i=1}^{n} [-x_i \sin(\sqrt{|x_i|})] \tag{25}$$

Test area is usually restricted to -500 $\leq x_i \leq$ 500, $i = 1, \dots, n$. Its global minimum $f_{19} = -418.9829n$ is obtainable for $x_i = 420.9687, i = 1, \dots, n$.

3.6.2.20. Six Hump Camel Back Function

$$f_{20} = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2) x_2^2 \tag{26}$$

Test area is usually restricted to -3 $\leq x_1 \leq$ 3, -2 $\leq x_2 \leq$ 2 f_{20}= - 1.0316 is obtainable at the point (- 0.0898, 0.7126) and (0.0898, - 0.7126).

3.6.2.21. Sum of Different Powers Function

$$f_{21} = \sum_{i=1}^{n} |x_i|^{i+1} \tag{27}$$

Test area is usually restricted to -1$\leq x_i \leq$1, $i = 1, \dots, n$. Its global minimum $f_{21} = 0$ is obtainable for $x_i = 0, i = 1, \dots, n$.

3.6.3. Statistical Tests

While comparing algorithms, hypothesis testing (Haigh and Conover 2006) has been commonly used in order to infer. However, the null hypothesis, H_0, and the alternative hypothesis, H_1, are supposed to be defined to make inferences. The null hypothesis is a proposition that generally refers no differences between compared algorithms. Conversely, the alternative hypothesis is an expression of the differences. In our case;

H_0 *: There is no difference between compared selection methods*

H_1 *: There is difference between compared selection methods*

Moreover, the probability value of a statistical test, α, decides which hypothesis should be rejected, which is commonly chosen 0.05 or 0.1. For our test, α is accepted as 0.05.

3.6.3.1. Friedman Test

The Friedman test, introduced by Friedman (Friedman 1937, 2007), is a non-parametric statistical test. It has been utilized to specify differences in behavior of more than one method. In the Friedman test, test cases are represented in rows and results of compared methods takes places in columns.

The method initiates its procedure by giving rank to each row, then, it computes the total value of ranks for each column. The significance of the test is determined by X^2, Chi-square, distribution with k-1 degrees of freedom (df) where k is the number of compared methods. Appropriate chi-square values corresponding to df value can be found in (Kafadar and Sheskin 2006). If found X^2 value is greater than the expected one, it is required to reject the null hypothesis. Found X^2 value can be obtained by utilizing Equation (28):

$$X^2{}_{found} = \frac{12}{n.k.(k+1)} * \sum R^2 - 3n.(k+1) \quad (28)$$

3.6.3.2. Wilcoxon Signed Rank Test

The Wilcoxon Signed Rank test is another non-parametric statistical test that has been used to state the differences between two samples or two algorithms. As a common way, the algorithm initially takes differences between the results of the two methods whose sizes are Nx1 in order to obtain the difference vector, where N means the number of tests. Then, the test ranks to each row of the vector from 1 to N, i.e. min. value is ranked 1. Afterwards, it computes R_+ and R_- values which represent total rank of positive or negative values of the difference vector. Based on the values, called T value of the test, $\min(R_+, R_-)$ is determined. Thus, *p value* of this test is computed by utilizing the T value.

4. RESULTS AND DISCUSSIONS

4.1. Introduction

The proposed methods were implemented by using Matlab 2017a. For all calculations, Arithmetic Crossover and Random Mutation have been employed as crossover and mutation operations, respectively. Arithmetic Crossover is accomplished by applying Equation (29) to determine the offspring:

$$ {}^{o}_{i}G = \frac{{}^{1}_{i}G + {}^{2}_{i}G}{2} + \lambda . \left| {}^{2}_{i}G - {}^{1}_{i}G \right| . (2.rand - 1) \tag{29} $$

where λ is a predefined value and was taken as 0.6 throughout the tests. In the formula, ${}^{0}_{i}G$ represents i^{th} gene of offspring, ${}^{1}_{i}G$ and ${}^{2}_{i}G$ stand for i^{th} genes of first and second mate, respectively. This formula prohibits population collapse. Random Mutation, on the other hand, is employed to stochastically select gene(s) by replacing any value within the range of corresponding dimension(s).

In this section, the results of test functions, statistical tests and meta-optimization methods are explained and discussed.

4.2. Preliminary Tests

The performances of proposed methods are comparatively analyzed with well-known selection methods: ST, Linear Ranking (LR) and Roulette Wheel (RW). Identical parameter settings were applied to all selection methods such that crossover probability = 0.7, tournament size = 4, mutation probability = 0.05, and population sizes = {50, 100, 200}. The tests are repeated with 25 runs that have 25 different random seeds, in order to minimize the effects of randomization.

From the Appendix 3, it can be seen that the performances of the proposed algorithms overpower LR and RW selection methods. Therefore, LR and RW selection methods will be omitted from further analysis. Since the most of the

proposed methods are based on ST, tests and analysis will be carried out on ST selection method. Furthermore, in order to demonstrate the abilities of the proposed methods, some other selection methods that are based on ST will be included for the rest of the studies.

4.3. Results by Benchmarks

The performances of proposed methods are comparatively analyzed with ST and prevalent selection methods that are based on ST; CS, FGTS, RTS and UTS. The parameter settings that are described in Section 4.2 were applied to all selection methods.

Tables 4.1.1 – 4.1.21 demonstrate results of the test functions. In these tables, median values were used to determine which method was more successful while examining the test results. The reason why the mean value was not used for interpreting the results is that if we had a high value in one of the replicates, the mean would rise and directly affects other 24 replicates even if we had low results in these replicates. Hence, this would not allow us to interpret other good results. Besides, Interquartile Range (IQR), which is a robust estimator, is employed to show how 50% of data behave.

Table 4.1.1. Test results for Ackley function

f_1	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	1,03E+00	2,05E+00	3,39E-01	3,27E-01	6,02E-02	5,74E-02
NASM	3,14E-01	1,01E+00	6,75E-02	1,83E-01	7,07E-03	1,93E-02
IASM	1,15E-01	9,28E-01	4,62E-02	2,79E-01	4,73E-09	2,55E-06
INASM	1,95E-01	4,64E-01	5,58E-06	2,82E-03	**8,88E-16**	9,55E-10
OA	2,44E-01	5,15E-01	2,51E-02	5,65E-02	1,35E-03	2,27E-02
N-OA	3,54E-02	1,20E-01	3,29E-03	1,74E-02	7,74E-05	2,49E-03
BMT	**3,65E-04**	2,40E-03	**3,91E-06**	1,42E-04	8,64E-09	1,82E-07
CS	3,57E+00	1,37E+00	4,02E-01	9,87E-01	7,78E-02	1,75E-01
FGTS	5,97E-01	1,59E+00	4,51E-03	2,77E-01	4,03E-06	6,26E-03
RTS	5,63E-01	1,36E+00	3,49E-02	2,70E-01	2,34E-06	2,20E-04
ST	2,31E-01	1,21E+00	4,51E-03	1,24E-01	4,71E-06	6,26E-03
UTS	6,94E-01	1,99E+00	1,05E-01	3,83E-01	6,06E-02	2,49E-01

Table 4.1.2**.** Test results for Axis Parallel Hyper-Ellipsoid function

f_2	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	2,84E-02	1,89E-01	1,17E-02	4,37E-02	1,77E-03	5,73E-03
NASM	4,64E-03	1,80E-02	1,92E-03	1,33E-02	1,24E-05	4,13E-04
IASM	6,02E-03	5,84E-02	2,68E-08	3,30E-04	1,74E-18	1,23E-10
INASM	1,46E-03	8,78E-03	1,92E-06	9,47E-05	**6,89E-27**	2,57E-18
OA	1,26E-03	5,41E-03	6,67E-05	6,25E-04	1,24E-06	1,93E-05
N-OA	6,45E-05	9,59E-04	3,48E-06	1,40E-05	1,08E-10	4,07E-08
BMT	**3,69E-09**	7,82E-07	**1,17E-12**	4,16E-10	1,66E-17	4,00E-15
CS	1,03E-02	1,64E-02	1,31E-04	9,62E-04	1,22E-03	5,57E-03
FGTS	1,92E-04	2,32E-03	2,71E-07	6,39E-05	1,20E-16	8,54E-06
RTS	6,75E-04	2,16E-03	5,58E-06	9,45E-05	3,86E-08	2,62E-05
ST	2,97E-04	1,77E-03	1,36E-06	4,83E-05	7,29E-17	1,96E-07
UTS	3,87E-04	1,00E-03	2,94E-05	4,90E-04	2,24E-05	5,47E-04

Table 4.1.3. Test results for Booth function

f_3	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	1,01E-02	4,83E-02	5,54E-03	2,54E-02	5,85E-04	3,54E-03
NASM	3,41E-03	1,69E-02	4,07E-05	8,01E-04	6,66E-15	5,45E-09
IASM	2,57E-03	1,19E-02	1,73E-04	1,05E-03	1,48E-05	4,75E-04
INASM	7,82E-04	1,03E-02	8,95E-06	3,76E-04	**1,38E-18**	3,10E-08
OA	7,65E-04	2,19E-03	6,89E-05	8,07E-04	1,19E-05	6,94E-05
N-OA	**1,18E-04**	7,35E-04	7,32E-07	1,13E-05	3,83E-10	3,80E-07
BMT	3,29E-04	1,79E-02	**1,35E-07**	2,20E-05	1,65E-13	6,56E-10
CS	3,05E-01	7,39E-01	2,64E-02	1,40E-01	1,20E-02	4,16E-02
FGTS	2,26E-02	2,44E-01	2,15E-04	2,93E-03	1,30E-10	1,73E-05
RTS	2,68E-02	1,15E-01	1,37E-05	9,87E-04	4,71E-08	1,20E-05
ST	3,16E-02	1,83E-01	2,15E-04	3,16E-03	1,30E-10	1,73E-05
UTS	1,31E-01	2,78E-01	8,92E-03	5,39E-02	9,02E-05	1,10E-03

Table 4.1.4. Test results for Branins function

f_4	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	4,05E-01	3,71E-02	3,99E-01	3,96E-03	**3,98E-01**	4,34E-04
NASM	3,99E-01	6,36E-03	**3,98E-01**	4,93E-04	**3,98E-01**	1,22E-04
IASM	**3,98E-01**	9,61E-03	**3,98E-01**	7,77E-04	**3,98E-01**	3,64E-10
INASM	**3,98E-01**	3,15E-03	**3,98E-01**	9,08E-05	**3,98E-01**	2,15E-08
OA	**3,98E-01**	2,33E-03	**3,98E-01**	8,93E-04	**3,98E-01**	1,12E-05
N-OA	**3,98E-01**	5,63E-04	**3,98E-01**	4,42E-05	**3,98E-01**	3,90E-07
BMT	**3,98E-01**	2,32E-04	**3,98E-01**	8,08E-08	**3,98E-01**	5,72E-13
CS	4,65E-01	2,05E-01	4,01E-01	6,31E-03	**3,98E-01**	3,87E-03
FGTS	4,02E-01	2,28E-02	**3,98E-01**	4,70E-04	**3,98E-01**	4,97E-05
RTS	4,00E-01	1,35E-02	**3,98E-01**	5,09E-04	**3,98E-01**	8,22E-05
ST	4,02E-01	1,98E-02	**3,98E-01**	1,59E-03	**3,98E-01**	4,97E-05
UTS	4,08E-01	2,43E-02	4,00E-01	7,63E-03	**3,98E-01**	8,53E-04

Table 4.1.5. Test results for Sixth Bukin function

f_5	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	1,60E+00	1,14E+00	1,15E+00	9,05E-01	9,60E-01	4,53E-01
NASM	6,66E-01	2,34E+00	6,30E-02	3,24E-01	2,58E-02	1,81E-01
IASM	4,04E-01	6,84E-01	4,14E-01	5,38E-01	1,99E-01	1,26E-01
INASM	1,41E-01	8,47E-01	9,10E-02	5,89E-01	4,29E-02	2,19E-01
OA	1,43E+00	2,36E+00	1,42E+00	2,27E+00	7,73E-01	1,64E+00
N-OA	1,03E-01	4,06E-01	3,50E-02	4,30E-01	4,52E-02	6,08E-01
BMT	**2,19E-02**	2,25E-02	**1,42E-02**	1,89E-02	**1,79E-02**	1,78E-02
CS	2,04E+00	6,13E+00	2,78E-01	1,16E+00	3,17E-01	1,62E+00
FGTS	1,49E+00	3,42E+00	3,76E-01	1,81E+00	2,37E-01	9,98E-01
RTS	1,04E+00	2,62E+00	2,72E-02	1,43E-01	2,38E-02	3,37E-02
ST	1,95E+00	3,52E+00	9,21E-01	2,36E+00	2,37E-01	1,09E+00
UTS	2,05E+00	3,03E+00	2,06E+00	2,13E+00	1,30E+00	1,33E+00

Table 4.1.6 Test results for De Jong function

f_6	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	1,39E-02	2,63E-02	9,06E-04	7,94E-03	5,52E-05	2,73E-04
NASM	2,59E-04	3,24E-03	1,51E-05	8,07E-05	3,00E-07	4,27E-06
IASM	4,91E-04	3,39E-03	1,62E-08	4,19E-05	2,06E-20	5,78E-11
INASM	1,06E-04	1,14E-03	7,75E-11	3,47E-06	**3,69E-23**	4,92E-13
OA	1,72E-04	4,07E-04	2,61E-05	1,16E-04	8,25E-08	4,39E-07
N-OA	7,79E-06	5,39E-05	2,86E-08	2,72E-07	5,56E-11	3,13E-08
BMT	**1,20E-08**	4,10E-07	**7,24E-14**	6,00E-12	5,06E-18	9,66E-17
CS	5,60E-03	8,41E-03	2,58E-05	9,68E-05	2,14E-05	6,35E-05
FGTS	1,52E-04	5,87E-04	5,25E-07	2,34E-05	9,68E-16	1,33E-09
RTS	3,12E-04	1,34E-03	3,02E-06	1,06E-04	6,39E-14	6,55E-07
ST	1,64E-04	5,60E-04	5,60E-07	1,94E-05	9,68E-16	1,33E-09
UTS	4,62E-04	1,83E-03	3,58E-05	1,61E-04	1,56E-06	8,56E-05

Table 4.1.7. Test results for Fifth De Jong function

f_7	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	3,26E+00	4,91E+00	1,03E+00	3,25E+00	1,01E+00	5,15E-01
NASM	2,27E+00	4,42E+00	1,99E+00	1,33E+00	1,20E+00	1,65E+00
IASM	1,99E+00	1,91E+00	9,99E-01	6,65E-02	**9,98E-01**	6,90E-06
INASM	1,02E+00	1,06E+00	**9,98E-01**	1,70E-05	**9,98E-01**	2,09E-09
OA	9,99E-01	5,08E-02	**9,98E-01**	4,52E-06	**9,98E-01**	3,42E-09
N-OA	**9,98E-01**	1,85E-06	**9,98E-01**	1,07E-09	**9,98E-01**	4,76E-11
BMT	**9,98E-01**	9,94E-01	**9,98E-01**	9,36E-12	**9,98E-01**	8,88E-16
CS	9,80E+00	8,03E+00	**9,98E-01**	2,36E-03	1,09E+00	2,23E+00
FGTS	9,99E-01	8,20E-01	**9,98E-01**	3,79E-03	**9,98E-01**	6,89E-05
RTS	1,01E+00	1,10E+00	**9,98E-01**	8,55E-06	**9,98E-01**	6,98E-08
ST	9,99E-01	7,89E-02	**9,98E-01**	3,34E-03	**9,98E-01**	1,76E-05
UTS	1,71E+00	9,96E-01	**9,98E-01**	2,44E-03	**9,98E-01**	7,34E-05

Table 4.1.8**.** Test results for Drop Wave function

f_8	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	-9,28E-01	1,58E-01	-9,32E-01	2,39E-02	-9,36E-01	5,36E-04
NASM	-9,36E-01	3,15E-03	-9,36E-01	2,20E-02	-9,77E-01	4,74E-02
IASM	-9,33E-01	2,78E-01	-9,36E-01	2,91E-02	-9,84E-01	6,18E-02
INASM	**-9,36E-01**	1,50E-01	-9,36E-01	6,23E-02	-9,96E-01	6,38E-02
OA	-9,32E-01	9,01E-02	-9,36E-01	1,56E-02	-9,36E-01	3,05E-02
N-OA	**-9,36E-01**	9,73E-04	-9,36E-01	6,23E-02	-9,36E-01	6,36E-02
BMT	**-9,36E-01**	5,92E-13	**-9,98E-01**	9,36E-12	**-1,00E+00**	6,38E-02
CS	**-9,36E-01**	8,91E-02	-9,36E-01	9,82E-03	-9,36E-01	5,96E-02
FGTS	**-9,36E-01**	6,45E-05	-9,36E-01	5,34E-02	-9,95E-01	2,41E-02
RTS	**-9,36E-01**	5,25E-02	-9,36E-01	6,34E-02	-1,00E+00	2,70E-03
ST	**-9,36E-01**	9,88E-05	-9,36E-01	5,35E-02	-9,98E-01	2,68E-02
UTS	**-9,36E-01**	6,89E-04	-9,36E-01	5,53E-03	-9,36E-01	4,92E-02

Table 4.1.9. Test results for Easom function

f_9	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	-5,59E-02	4,64E-01	-6,55E-01	7,3E-01	-8,67E-01	2,4E-01
NASM	-2,50E-05	1,28E-02	-1,10E-02	6,8E-01	-7,87E-01	6,8E-01
IASM	-3,62E-01	8,57E-01	-9,51E-01	1,7E-01	-9,73E-01	6,2E-02
INASM	-1,57E-01	9,21E-01	-9,85E-01	5,04E-01	**-1,0E+00**	3,9E-02
OA	-9,45E-01	3,30E-01	-9,98E-01	3,88E-02	-9,98E-01	6,30E-03
N-OA	**-9,9E-01**	1,33E-02	**-1,0E+00**	1,87E-03	**-1,0E+00**	1,53E-05
BMT	-9,73E-01	1,00E+00	**-1**	2,80E-07	**-1**	1,64E-09
CS	-2,52E-23	1,53E-11	-4,73E-01	9,43E-01	-8,97E-01	9,85E-01
FGTS	-5,53E-05	9,69E-01	-7,60E-01	9,57E-01	-9,94E-01	1,42E-01
RTS	-7,68E-05	7,15E-01	-8,97E-01	9,93E-01	-1,00E+00	1,08E-02
ST	-3,71E-02	8,68E-01	-1,51E-01	8,91E-01	-9,93E-01	4,78E-01
UTS	-1,68E-06	3,67E-01	-5,93E-02	8,22E-01	-9,20E-01	9,98E-01

Table 4.1.10. Test results for Goldstein-Price function

f_{10}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	3,49E+00	1,48E+00	3,04E+00	1,02E-01	**3,00E+00**	9,72E-03
NASM	3,11E+00	7,08E-01	**3,00E+00**	2,57E-02	**3,00E+00**	2,58E-03
IASM	3,11E+00	5,88E-01	**3,00E+00**	8,23E-03	**3,00E+00**	2,58E-04
INASM	3,01E+00	4,61E-02	**3,00E+00**	1,95E-03	**3,00E+00**	1,17E-04
OA	**3,00E+00**	2,88E-03	**3,00E+00**	1,07E-03	**3,00E+00**	1,60E-05
N-OA	**3,00E+00**	7,44E-04	**3,00E+00**	6,24E-06	**3,00E+00**	1,17E-08
BMT	**3,00E+00**	1,40E+00	**3,00E+00**	2,08E-09	**3,00E+00**	3,81E-13
CS	4,47E+00	6,84E+00	3,01E+00	3,03E-02	**3,00E+00**	8,33E-03
FGTS	3,11E+00	5,30E-01	**3,00E+00**	9,92E-03	**3,00E+00**	1,96E-04
RTS	3,09E+00	8,30E-01	**3,00E+00**	4,10E-02	**3,00E+00**	1,84E-04
ST	3,19E+00	4,32E-01	3,01E+00	1,76E-02	**3,00E+00**	4,95E-04
UTS	3,14E+00	9,63E-01	3,01E+00	5,67E-02	**3,00E+00**	1,50E-03

Table 4.1.11. Test results for Griewangk function

f_{11}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	2,12E-01	2,21E-01	7,52E-02	5,47E-02	3,54E-02	3,39E-02
NASM	3,12E-02	2,92E-02	3,39E-02	4,33E-02	9,88E-03	1,20E-02
IASM	7,28E-02	9,93E-02	2,32E-02	2,30E-02	7,40E-03	7,56E-03
INASM	4,79E-02	5,90E-02	1,23E-02	2,05E-02	**2,06E-03**	9,78E-03
OA	1,40E-01	1,54E-01	3,42E-02	5,15E-02	2,07E-02	1,58E-02
N-OA	3,38E-02	3,18E-02	1,58E-02	3,12E-02	7,54E-03	3,48E-03
BMT	**1,97E-02**	3,21E-02	**7,40E-03**	2,47E-03	7,40E-03	7,40E-03
CS	5,96E-01	5,21E-01	6,49E-02	1,05E-01	3,15E-02	4,86E-02
FGTS	8,63E-02	1,18E-01	3,06E-02	3,79E-02	1,14E-02	1,65E-02
RTS	5,55E-02	1,54E-01	2,01E-02	2,59E-02	7,82E-03	1,25E-02
ST	1,06E-01	1,51E-01	3,12E-02	4,48E-02	1,15E-02	1,79E-02
UTS	1,99E-01	1,89E-01	7,14E-02	7,19E-02	3,08E-02	3,19E-02

Table 4.1.12. Test results for Holder Table function

f_{12}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	-18,6178	1,01E+00	-19,1110	4,54E-01	-19,2016	2,15E-02
NASM	-19,1925	3,47E-02	-19,2079	7,10E-03	**-19,2085**	2,01E-04
IASM	-19,0511	3,03E-01	-19,1743	1,32E-01	-19,2080	3,81E-03
INASM	-19,2063	8,15E-03	-19,2083	2,16E-03	**-19,2085**	6,59E-05
OA	-19,2051	8,02E-03	-19,2084	5,07E-04	-19,2084	2,02E-04
N-OA	-19,2084	2,83E-04	-19,2084	8,74E-05	**-19,2085**	9,76E-07
BMT	**-19,2085**	2,78E-05	**-19,2085**	8,36E-09	**-19,2085**	1,04E-11
CS	-17,8498	7,8684762	-19,2073	0,00848	-19,2080	0,0024476
FGTS	-19,2017	0,0271871	-19,2050	0,00716	**-19,2085**	0,0001681
RTS	-19,2051	0,0324796	-19,2083	0,00171	**-19,2085**	7,84E-06
ST	-19,1925	0,1114633	-19,2061	0,00970	-19,2084	0,0002494
UTS	-19,1902	0,0979455	-19,2067	0,01165	-19,2080	0,0012285

Table 4.1.13. Test results for Langermann function

f_{13}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	-4,04E+00	3,4E-01	-4,09E+00	7,6E-02	-4,13E+00	5,96E-02
NASM	-4,02E+00	4,2E-01	-4,12E+00	5,1E-02	-4,13E+00	4,46E-02
IASM	-4,02E+00	2,6E-01	-4,10E+00	4,86E-02	-4,13E+00	4,14E-02
INASM	-4,08E+00	7,5E-02	-4,10E+00	6,11E-02	-4,15E+00	2,86E-02
OA	-4,10E+00	4,7E-02	**-4,13E+00**	2,77E-02	**-4,16E+00**	2,82E-02
N-OA	**-4,13E+00**	5,98E-02	**-4,13E+00**	5,39E-02	**-4,16E+00**	2,82E-02
BMT	**-4,13E+00**	7,19E-02	-4,10E+00	4,37E-02	**-4,16E+00**	3,49E-02
CS	-3,84E+00	1,58E+00	-4,08E+00	8,99E-02	-4,10E+00	7,37E-02
FGTS	-4,03E+00	1,08E-01	-4,09E+00	5,53E-02	-4,10E+00	5,32E-02
RTS	-4,08E+00	1,75E-01	-4,12E+00	3,00E-02	-4,13E+00	3,27E-02
ST	-4,03E+00	9,30E-02	-4,08E+00	6,11E-02	-4,11E+00	5,63E-02
UTS	-4,02E+00	2,32E-01	-4,07E+00	9,40E-02	-4,08E+00	5,95E-02

Table 4.1.14. Test results for Michalewicz function

f_{14}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	**-1,80E+00**	1,04E-02	**-1,80E+00**	1,68E-03	**-1,80E+00**	1,62E-04
NASM	**-1,80E+00**	2,09E-03	**-1,80E+00**	7,70E-04	**-1,80E+00**	4,12E-06
IASM	**-1,80E+00**	1,87E-03	**-1,80E+00**	4,20E-05	**-1,80E+00**	8,58E-10
INASM	**-1,80E+00**	1,76E-03	**-1,80E+00**	6,60E-06	**-1,80E+00**	3,53E-13
OA	**-1,80E+00**	1,38E-04	**-1,80E+00**	5,77E-05	**-1,80E+00**	2,38E-06
N-OA	**-1,80E+00**	1,04E-05	**-1,80E+00**	4,31E-07	**-1,80E+00**	9,69E-10
BMT	**-1,80E+00**	7,56E-05	**-1,80E+00**	7,23E-11	**-1,80E+00**	2,20E-14
CS	-1,79E+00	4,72E-02	**-1,80E+00**	2,09E-03	**-1,80E+00**	4,04E-04
FGTS	**-1,80E+00**	2,83E-03	**-1,80E+00**	4,62E-05	**-1,80E+00**	1,47E-07
RTS	**-1,80E+00**	2,75E-03	**-1,80E+00**	4,32E-05	**-1,80E+00**	3,06E-07
ST	**-1,80E+00**	4,06E-03	**-1,80E+00**	4,62E-05	**-1,80E+00**	1,47E-07
UTS	**-1,80E+00**	5,85E-03	**-1,80E+00**	1,20E-03	**-1,80E+00**	7,64E-06

Table 4.1.15. Test results for Rastrigin function

f_{15}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	1,47E+00	8,05E-01	1,01E+00	1,10E+00	6,63E-01	9,98E-01
NASM	9,96E-01	1,60E-01	9,95E-01	1,00E+00	1,64E-01	9,95E-01
IASM	2,08E-01	7,25E-01	**0,00E+00**	8,44E-02	4,78E-04	1,10E-02
INASM	1,22E-01	5,41E-01	8,42E-07	1,25E-02	**0,00E+00**	1,08E-06
OA	**0,00E+00**	2,32E-03	**0,00E+00**	0,00E+00	**0,00E+00**	0,00E+00
N-OA	**0,00E+00**	0,00E+00	**0,00E+00**	0,00E+00	**0,00E+00**	0,00E+00
BMT	3,13E-03	3,56E-01	**0,00E+00**	0,00E+00	**0,00E+00**	0,00E+00
CS	1,17E+00	1,32E+00	1,75E-02	1,08E-01	2,79E-03	1,50E-02
FGTS	3,13E-02	2,83E-01	9,88E-03	1,08E-01	4,11E-05	3,77E-03
RTS	1,00E-01	5,44E-01	1,73E-03	1,13E-02	3,49E-07	3,56E-05
ST	2,19E-01	5,36E-01	2,51E-02	8,53E-02	4,11E-05	4,94E-03
UTS	6,26E-01	7,48E-01	2,15E-02	7,74E-02	2,36E-03	2,34E-02

Table 4.1.16. Test results for Rosenbrock's Valley function

f_{16}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	3,25E-02	8,32E-02	7,07E-03	2,74E-02	1,84E-03	8,59E-03
NASM	1,31E-02	4,39E-02	7,49E-03	3,13E-02	3,34E-04	5,00E-03
IASM	1,21E-02	7,54E-02	3,26E-04	1,23E-02	4,21E-05	1,34E-02
INASM	4,06E-02	1,56E-01	1,39E-02	7,83E-02	8,73E-05	1,00E-02
OA	1,73E-03	6,04E-03	6,91E-05	5,48E-04	6,83E-07	3,61E-06
N-OA	**9,04E-04**	1,07E-02	**1,58E-05**	1,64E-04	**2,15E-09**	8,68E-07
BMT	4,65E-02	6,90E-02	2,07E-04	7,44E-03	1,80E-05	1,49E-02
CS	2,75E-01	5,35E-01	4,33E-02	8,72E-02	2,51E-02	4,86E-02
FGTS	4,13E-02	1,56E-01	5,68E-03	3,91E-02	6,02E-06	1,79E-03
RTS	4,77E-02	1,26E-01	1,30E-03	2,70E-02	3,07E-03	1,54E-02
ST	5,02E-02	1,60E-01	6,96E-03	3,85E-02	6,02E-06	2,17E-03
UTS	5,42E-02	1,19E-01	2,60E-02	6,43E-02	4,06E-03	1,52E-02

Table 4.1.17. Test results for Rotated Hyper-Ellipsoid function

f_{17}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	2,59E-01	1,20E+00	1,08E-01	2,84E-01	1,11E-03	6,12E-03
NASM	2,06E-02	1,17E-01	1,61E-03	7,87E-03	4,76E-05	2,24E-04
IASM	2,39E-02	2,26E-01	1,28E-05	3,07E-03	2,05E-09	1,37E-03
INASM	7,26E-03	4,41E-02	3,44E-06	3,51E-04	1,42E-15	1,45E-07
OA	5,47E-03	2,67E-02	3,73E-04	3,00E-03	1,45E-05	4,75E-04
N-OA	1,24E-03	4,53E-03	7,50E-06	5,32E-05	3,13E-10	5,77E-07
BMT	**3,44E-08**	9,14E-06	**9,82E-12**	8,66E-10	**8,03E-16**	7,16E-13
CS	2,15E+00	3,10E+00	7,13E-03	5,79E-02	2,92E-03	1,81E-02
FGTS	6,18E-02	1,97E-01	1,72E-04	4,22E-03	2,71E-10	4,61E-05
RTS	5,60E-02	4,62E-01	4,22E-05	1,21E-03	3,21E-09	1,82E-06
ST	6,40E-02	2,28E-01	4,84E-04	8,48E-03	2,72E-10	2,34E-05
UTS	1,01E-01	3,63E-01	1,43E-02	2,85E-02	1,03E-04	1,64E-03

Table 4.1.18. Test results for Shubert function

f_{18}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	-2,05E+02	7,27E+00	-2,08E+02	2,87E+00	-2,09E+02	1,29E+00
NASM	-2,05E+02	1,19E+01	-2,07E+02	3,78E+00	**-2,10E+02**	1,38E+00
IASM	-2,02E+02	4,45E+01	-2,06E+02	1,65E+01	**-2,10E+02**	1,62E+00
INASM	-2,03E+02	4,33E+01	-2,09E+02	4,86E+00	**-2,10E+02**	4,05E-01
OA	-2,08E+02	7,90E+00	**-2,10E+02**	1,63E+00	**-2,10E+02**	3,48E-01
N-OA	**-2,10E+02**	5,18E+00	**-2,10E+02**	5,02E-01	**-2,10E+02**	2,24E-01
BMT	**-2,10E+02**	4,48E+01	**-2,10E+02**	1,77E-04	**-2,10E+02**	5,25E-08
CS	-1,56E+02	7,21E+01	**-2,10E+02**	4,49E+01	**-2,10E+02**	1,40E+00
FGTS	-2,06E+02	4,55E+01	**-2,10E+02**	7,16E-01	**-2,10E+02**	1,92E-01
RTS	-2,08E+02	4,48E+01	**-2,10E+02**	1,14E+00	**-2,10E+02**	2,52E-03
ST	-2,07E+02	4,51E+01	**-2,10E+02**	1,32E+01	**-2,10E+02**	2,95E-01
UTS	-1,93E+02	4,26E+01	-2,08E+02	1,47E+01	**-2,10E+02**	4,15E-01

Table 4.1.19. Test results for Schwefel function

f_{19}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	-7,16E+02	1,76E+02	-8,20E+02	1,10E+02	**-8,38E+02**	9,77E+00
NASM	-8,25E+02	1,19E+02	-8,36E+02	2,74E+01	**-8,38E+02**	7,11E-01
IASM	-8,37E+02	1,80E+00	**-8,38E+02**	5,14E-02	**-8,38E+02**	2,62E-03
INASM	**-8,38E+02**	8,84E-01	**-8,38E+02**	6,42E-02	**-8,38E+02**	9,71E-07
OA	**-8,38E+02**	7,36E-01	**-8,38E+02**	7,72E-02	**-8,38E+02**	5,68E-03
N-OA	**-8,38E+02**	2,11E-01	**-8,38E+02**	1,14E-03	**-8,38E+02**	8,94E-06
BMT	**-8,38E+02**	1,72E-03	**-8,38E+02**	7,04E-09	**-8,38E+02**	2,61E-12
CS	-7,10E+02	1,99E+02	-8,38E+02	3,85E-01	-8,38E+02	1,41E-01
FGTS	-8,38E+02	1,01E+00	-8,38E+02	3,67E-01	-8,38E+02	1,32E-02
RTS	-8,37E+02	6,14E+00	-8,38E+02	9,08E-02	-8,38E+02	1,20E-03
ST	-8,38E+02	1,40E+00	-8,38E+02	1,05E-01	-8,38E+02	1,50E-01
UTS	-8,36E+02	3,04E+00	-8,38E+02	1,52E+00	-8,38E+02	8,58E-02

Table 4.1.20. Test results for Six Hump Camel Back function

f_{20}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	**-1,03E+00**	1,63E-03	**-1,03E+00**	1,16E-03	**-1,03E+00**	1,51E-04
NASM	**-1,03E+00**	1,47E-03	**-1,03E+00**	5,11E-04	**-1,03E+00**	1,03E-04
IASM	**-1,03E+00**	2,46E-03	**-1,03E+00**	5,49E-05	**-1,03E+00**	1,01E-08
INASM	**-1,03E+00**	1,44E-03	**-1,03E+00**	1,57E-05	**-1,03E+00**	1,26E-11
OA	**-1,03E+00**	2,91E-04	**-1,03E+00**	2,39E-05	**-1,03E+00**	3,55E-06
N-OA	**-1,03E+00**	1,84E-05	**-1,03E+00**	1,47E-06	**-1,03E+00**	9,48E-10
BMT	**-1,03E+00**	4,09E-07	**-1,03E+00**	1,36E-10	**-1,03E+00**	1,15E-12
CS	-1,02E+00	2,33E-02	**-1,03E+00**	3,40E-04	**-1,03E+00**	1,27E-04
FGTS	**-1,03E+00**	9,26E-04	**-1,03E+00**	2,85E-04	**-1,03E+00**	7,62E-08
RTS	**-1,03E+00**	1,90E-03	**-1,03E+00**	9,26E-05	**-1,03E+00**	2,01E-07
ST	**-1,03E+00**	1,43E-03	**-1,03E+00**	7,31E-05	**-1,03E+00**	7,62E-08
UTS	**-1,03E+00**	2,18E-03	**-1,03E+00**	3,83E-04	**-1,03E+00**	4,32E-05

Table 4.1.21. Test results for Sum of Different Powers function

f_{21}	*Pop size = 50*		*Pop size = 100*		*Pop size = 200*	
Method	*Median*	*IQR*	*Median*	*IQR*	*Median*	*IQR*
ASM	3,34E-06	8,02E-06	8,39E-07	2,02E-06	1,03E-07	3,90E-07
NASM	7,36E-08	7,07E-07	7,15E-09	1,48E-07	1,74E-09	8,95E-09
IASM	1,08E-07	5,72E-06	1,86E-09	6,31E-08	2,28E-23	2,42E-15
INASM	1,04E-07	4,52E-07	5,65E-12	2,73E-10	**1,41E-26**	1,34E-16
OA	3,67E-08	1,39E-06	2,54E-09	3,33E-08	6,35E-11	1,20E-09
N-OA	1,21E-09	5,91E-09	5,07E-12	1,24E-10	2,01E-14	9,87E-12
BMT	**4,92E-14**	5,96E-12	**2,90E-17**	1,19E-15	7,82E-24	1,23E-20
CS	9,87E-05	3,07E-04	3,45E-07	4,17E-06	1,32E-07	7,99E-07
FGTS	8,26E-07	6,27E-06	9,10E-10	7,08E-08	1,74E-14	1,21E-10
RTS	1,00E-06	1,82E-05	1,32E-09	2,33E-07	7,08E-15	1,41E-10
ST	2,15E-06	1,05E-05	9,91E-10	3,48E-08	1,74E-14	8,21E-11
UTS	2,56E-06	1,76E-05	4,81E-07	2,42E-06	1,09E-08	8,70E-08

4.4. Discussion for the Benchmark Tests

Each proposed method is discussed in their own sections. Furthermore, the visual presentations of the proposed methods are displayed to understand how the methods behave. As an instance, Axis Parallel Hyper Ellipsoid function is chosen to show their behaviors when the number of iterations and population size are 100. This function was selected for the demonstration since it clearly displays their behaviors. However, the demonstrations of other functions can be found in Appendix 4.

In addition to the proposed methods, all tested methods (ST, CS, FGTS, RTS and UTS) were also was shown in the Figure 4.1 to illustrate the difference in behaviors.

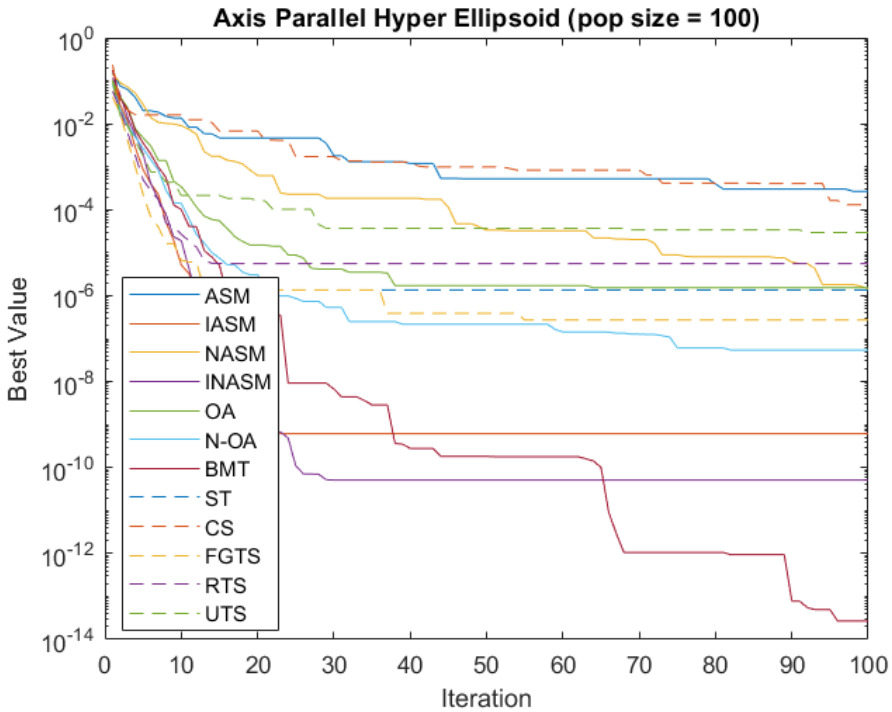


Figure 4.1. Proposed methods in Axis Parallel Hyper Ellipsoid

4.4.1. Discussion on ASM

Since ASM is a straightforward method, it is not expected that ASM presents a decent performance. Unlike the expectation, it offers the best results in Branin, Goldstein Price, Schwefel functions when population size is 200, and in Michalewicz and Six Hump Camel Back functions for all cases.

According to the figure, ASM cannot investigate the search space in an efficient manner because of its mating procedure.

4.4.2. Discussion on IASM

Adding ST to ASM, IASM, causes to have a more effective selection method than before. In addition to the functions where ASM introduces its best results, IASM

also gives better results in Rastrigin function when population size is 100, and in Fifth De Jong and Schubert functions when population size is 200.

As Figure 4.1 shows, IASM abandons for searching after nearly 20th iterations. Thus, it does not present the behavior that is desired to have. On the other hand, it offers better performance than ST and the prevalent methods that are also based on ST (CS, FGTS, RTS and UTS).

4.4.3. Discussion on NASM

It is apparent to say that improvement on ASM, NASM, was accomplished since NASM achieves better value than ASM reached. Besides to the functions where ASM performs in a decent manner, IASM also offers better results in Holder Table and Schubert functions when population size is 200.

Based on the figure, in 10th iteration, NASM achieves the value that ASM barely reaches in about 80th iteration. Even though NASM presents better performance than ASM, the rejection mechanism affects its behavior, as in ASM. Thus, it does not give the performance that willing to attain.

4.4.4. Discussion on INASM

It is obvious that INASM carries out better performances than NASM. It shows better results in the condition where NASM is succeeded. Moreover, it is only the winner in six conditions out of sixty-three conditions. On the other hand, it cannot achieve any accomplishment when population size is lower, except Branins, Michalewicz and Six Hump Camel Back. For this reason, INASM is not acknowledged as the method that desired to reach.

According to Figure 4.1, INASM gives up to search after nearly 25th iterations. Nevertheless, INASM presents better performance than all tested methods, except BMT.

4.4.5. Discussion on OA

Based on the obtained results, OA has proven itself in Fifth De Jong, Langermann and Schubert functions when population sizes are 100 and 200, in Branins, Goldstein Price, Michalewicz, Rastrigin, Schwefel, and Six Hump Camel Back functions for all population sizes.

The figure shows that OA presents better results than ASM thanks to the idea of mating with individuals not in the population. However, it abandons to explore search space about 60th iteration.

4.4.6. Discussion on N-OA

N-OA introduces its best performances where OA also has accomplishments. Furthermore, it is the only winner in Holder Table function when population size is 200, and in Booth function where population size is 50.

As Figure 4.1 demonstrates, improvement on OA, N-OA, has succeed. It does not give up to investigate the search space until 80th iteration. However, its performance is predominant to ST, CS, FGTS, RTS and UTS.

4.4.7. Discussion on BMT

Unlike the other proposed methods, BMT only abandon to search the space in a short time owing to its bipolar mood, see Figure 4.1. Thus, it presents the best performance in Axis Parallel Hyper Ellipsoid function. However, based on the obtained results, BMT also gives predominant results in other test functions:

In twenty-one functions with three population sizes (i.e. sixty-three conditions), BMT is successful in forty-nine conditions, and it is only the winner in twenty-one conditions among those sixty-three conditions. It shows its best performance either in unimodal functions or in multimodal functions. In addition, it has achievements in all population sizes. Therefore, **it is apparent to say that the first aim of this study, having a qualified selection method, was carried out.**

Since BMT presents the best performance comparing to other methods, it is selected as a meta optimizer for the second part of study. Furthermore, another reason that why BMT is chosen is the curiosity of the performance of BMT in the area of meta optimization.

On the other hand, BMT produces predominant results comparing to ST. In order to quieten the suspicion of how BMT that is based on ST presents more successful results than ST, next section provides the clarifications.

4.4.7.1. Why BMT Offers Better Results than ST?

Even though BMT presents better performance than ST, the results of BMT can be questioned, as it is very similar to ST. This section has been created to appease the suspicion of readers. Three techniques can be operated in order to answer the question: giving an example, population diversity measurements and the visualization of the discovered points.

4.4.7.2. Offering an Instance

It was assumed that it is required to solve a minimization problem with BMT and ST, where population size is 4 and tournament size is 2. After a number of iterations, it is supposed that the results in Table 4.2 was obtained.

Table 4.2. Snapshot of a population for a hypothetical problem

Individuals	*Points*	*Fitness Values*
X	(-1, 2)	2
Y	(-1, 2)	2
Z	(-1, 2)	2
T	(2, 5)	3

If ST is utilized, it only allows mating to X, Y, Z. As T cannot be chosen, it cannot find any other points except the individuals in the table. Hence, besides losing

the genetic diversity, it gets stuck in the point of X (Y or Z) even if the algorithm is at the beginning of its procedure.

However, as BMT allows the worst individuals for mating, the individual T can also attend the mating procedure. Thus, new points in the search space can be discovered; therefore, the genetic diversity maintains which causes that BMT may find better result comparing to ST.

4.4.7.3. Compare Genetic Diversities of BMT and ST

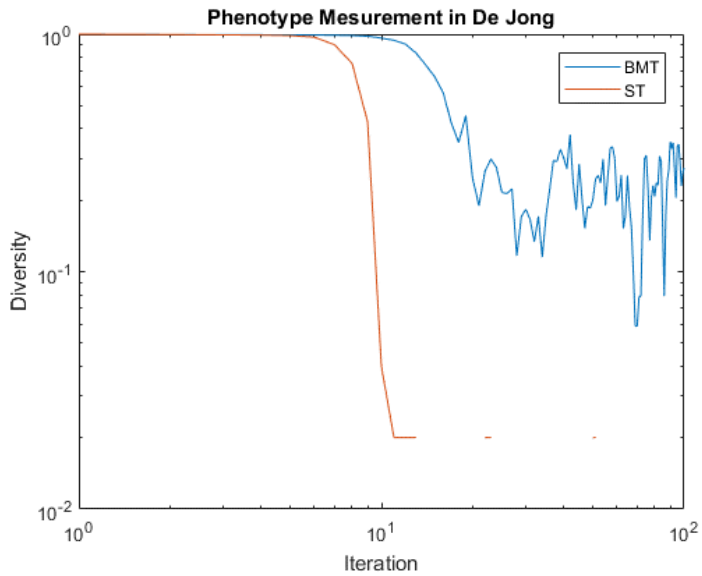


Figure 4.2. The phenotype measurements of BMT with ST

Many techniques such as Phenotype, Standard deviation, Genotype etc. can be utilized in order to evaluate genetic diversity of population (Mauldin 1984). However, it is advised to employ phenotype and genotype measurements since they provide consistent results (Mauldin 1984). Thus, in this thesis, phenotype and genotype techniques are chosen to measure the diversity of population after sequentially applying BMT and ST.

In phenotype measurement technique, for each iteration, the number of unique fitness values are divided to the population size in order to obtain the rate of genetic diversity of the population. For this procedure, it is required to have a fitness function to compute fitness values. De Jong's function was operated to demonstrate the performance of BMT and ST on the subject of the genetic diversity, see Figure 4.2. According to the figure, ST loses the genetic diversity of population nearly in 10th iteration while BMT is protecting it throughout all iterations.

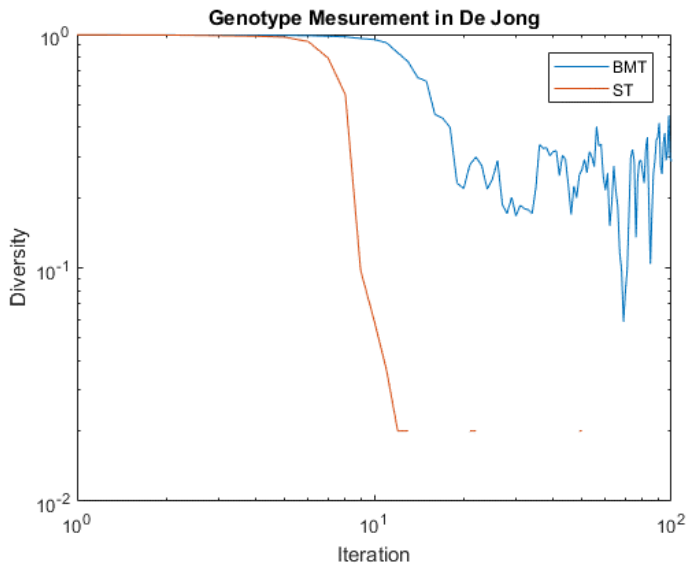


Figure 4.3. The genotype measurements of BMT with ST

Genotype diversity of the population is measured by comparing genetic structure of each individual. For this comparison, hamming distance has been generally used to show whether the structures of compared individuals are similar (Mauldin 1984). However, hamming distance can only be applied if the genetic structure of individuals is binary. Since a real coded GA is operated in the thesis, another distance should be implemented. When the genes are real numbers, Euclidean distance is the most common distance to evaluate genotype diversity of the population (Zhu 2003, Thomsen 2005, Mouret and Doncieux 2012). Hence, it

was also operated in this thesis to compare genetic structures of individuals. As in phenotype measurement, De Jong's function was also employed to display the effectiveness of BMT and ST on the genotype diversity of the population, in Figure 4.3. Figure 4.3 demonstrates that the behavior of BMT and ST are almost identical with Figure 4.2.

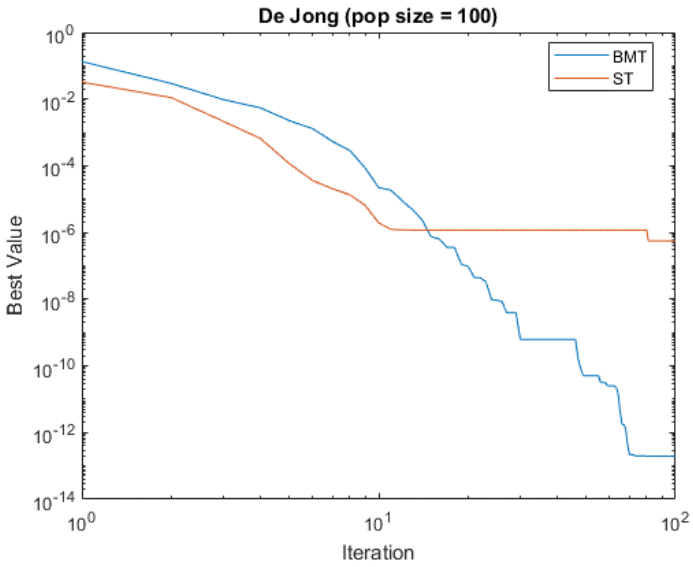


Figure 4.4. The performances of BMT and ST in De Jong's function

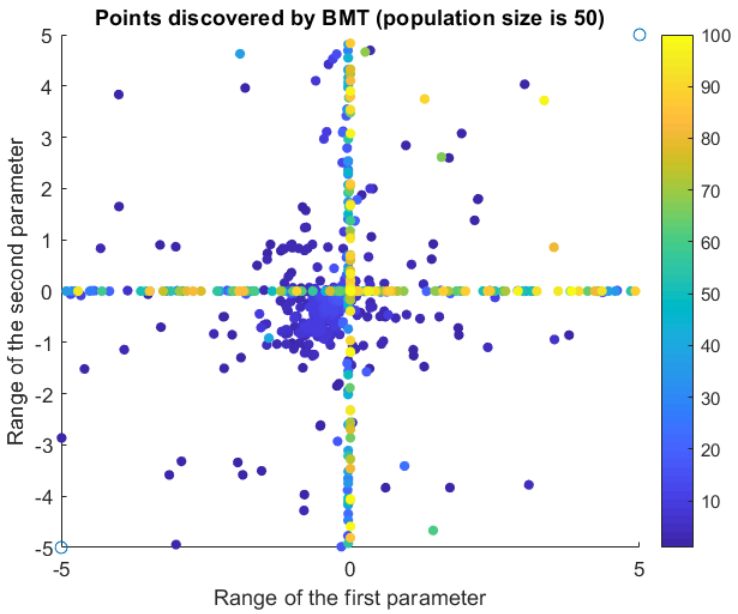
BMT offers better performance comparing to the performance of ST on the matter of remaining genetic diversity; therefore, it is expected that in De Jong's function, BMT should present an adequate result comparing to ST. Figure 4.4 was obtained to declare whether this expectation is accurate when population size is 100. Based on the figure, the performance of BMT is better than the performance of ST. Hence, in addition to performing successful performance to keep genetic diversity alive, BMT is also accomplished in the subject of reproductivity.

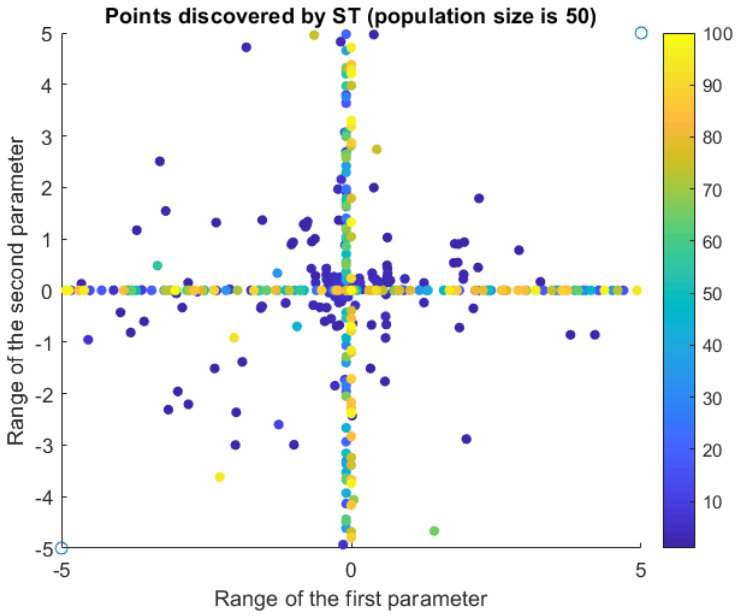
4.4.7.4. Harmony between Exploration and Exploitation

At the beginning of the first section, the importance of the harmony between exploration and exploitation is emphasized. In addition to having efficient exploitation manner, the appropriate selection method should not give up exploration throughout its process.

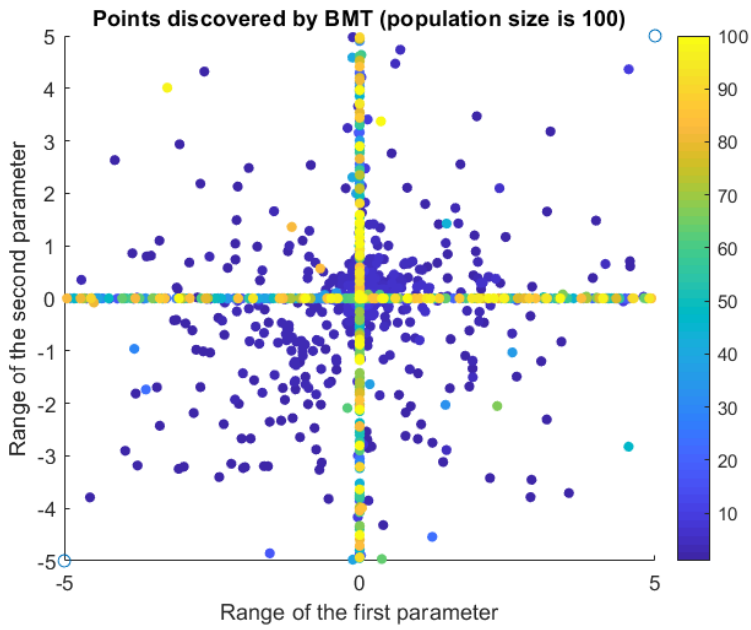
This section displays the behavior of BMT and ST on the subject of exploration and exploitation. For this aim, Axis Parallel Hyper Ellipsoid function is utilized to demonstrate their behaviors in different population sizes.

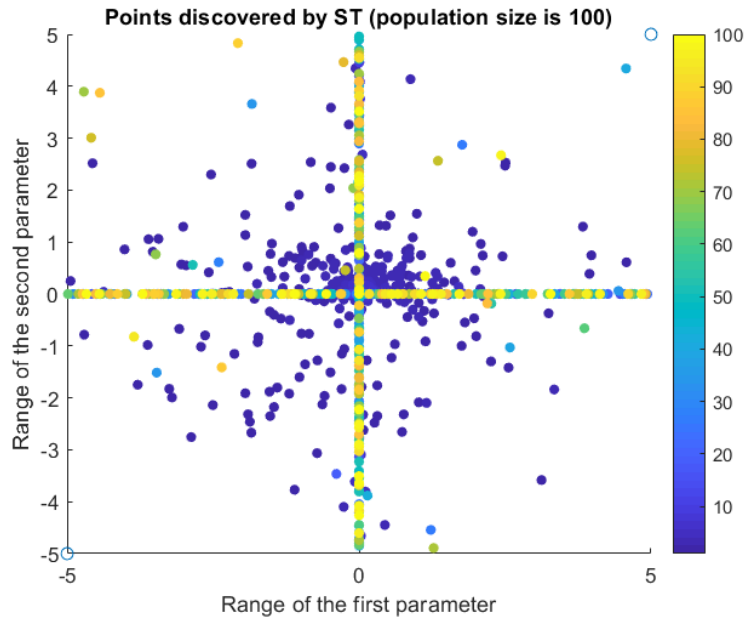
As in Figure 4.5, BMT presents more exploration and exploitation than ST in all population sizes. Since it adjusts exploration and exploitation in a decent manner, no doubt should be occurred about BMT results which are superior to ST.


Points discovered by BMT (population size is 50)
Range of the second parameter
Range of the first parameter
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5
100
90
80
70
60
50
40
30
20
10

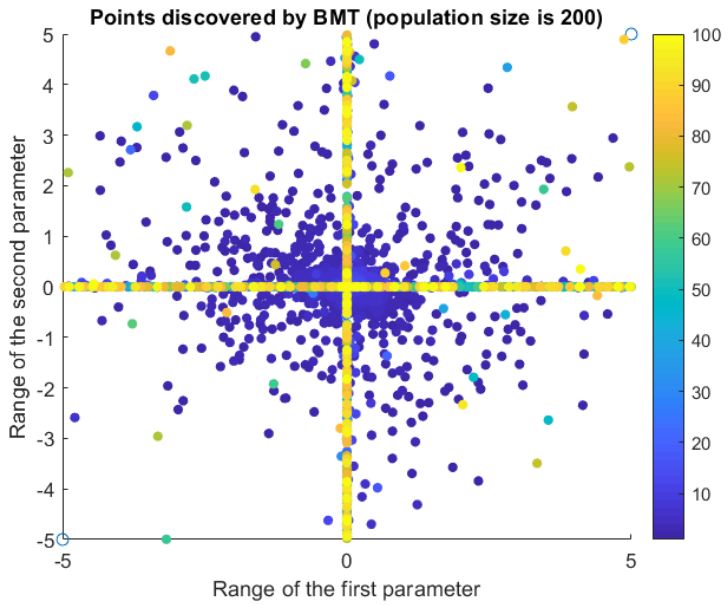

Points discovered by ST (population size is 50)
Range of the second parameter
Range of the first parameter
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5
100
90
80
70
60
50
40
30
20
10

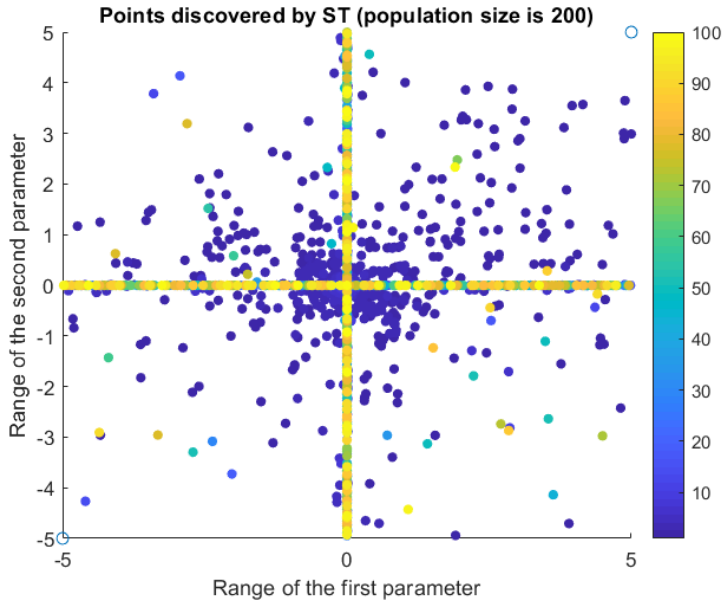
(a)


Points discovered by BMT (population size is 100)
Range of the second parameter
Range of the first parameter
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5
100
90
80
70
60
50
40
30
20
10


Points discovered by ST (population size is 100)
Range of the second parameter
Range of the first parameter
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5
100
90
80
70
60
50
40
30
20
10

(b)





(c)

Figure 4.5. The discovered points from BMT and ST in different pop sizes

4.5. Results and Discussion for the Statistical Tests

The results of Friedman test were obtained by IBM SPSS Statistics 22. Furthermore, *signrank* function of Matlab 2017a was utilized to attain the results of Wilcoxon Signed Rank Test.

4.5.1. Friedman Test

It is obvious to clarify which selection method is more successful by looking at Table 4.3. A selection method having minimum rank value means that the selection method offers better performance comparing to others. According to Table 4.3, BMT presents better performance for all population sizes in the twenty-one test functions.

Table 4.3. Mean ranks of the methods for the Friedman tests

Methods	Pop Size =50	Pop Size =100	Pop Size =200
ASM	9.41	10.47	10.35
IASM	6.47	5.88	3.94
NASM	5.65	7.82	8.82
INASM	4.53	3.59	2.18
OA	4.06	5.88	6.88
N-OA	2.12	2.24	4.29
BMT	**1.71**	**1.76**	**2.12**
ST	7.82	6.88	6.03
CS	11.94	11.41	11.47
FGTS	7.24	5.94	5.74
RTS	7.29	6.47	5.82
UTS	9.76	10.65	10.35

From the table in (Kafadar and Sheskin 2006), expected $X^2 = 19.68$ since df = 11 and $\alpha = 0.05$. In addition to having probability values that are less than α for all population sizes, found X^2 values, which are in Table 4.4, are also greater

than the expected one. Thus, the null hypothesis must be rejected, which refers that "*There are differences between compared selection methods"* in all cases. However, this test does not clarify especially which selection methods are different than other; therefore, Wilcoxon Signed Rank Test is applied.

Table 4.4. Statistical values for the Friedman tests

Statistical value	Pop Size =50	Pop Size =100	Pop Size =200
Chi-Square	137.1206	131.7954	145.6912
prob_value	6.2681E-24	7.53803E-23	1.1292E-16

4.5.2. Wilcoxon Signed Rank Test

Wilcoxon Sign Rank Test is operated in doubles: One method is taken from the first column of the table and respectively compared with all subsequent methods.

P values describe the significant differences between the compared algorithms. These values must be less than α to reject null hypothesis, otherwise, it is required to accept it, which means that compared algorithms has similarity in terms of performance.

According to Table 4.5, it is required to accept the null hypothesis, which is "*There is no difference between compared selection methods"*, for the following cases:

As p values are respectively 0.193, 0.193, 0.381, 0.356 and 0.435, ST is similar with ASM, NASM, IASM, FGTS and RTS in the manner of performance. Also, OA and INASM has similarity since p value is 0.163. In addition to this results, the performance of NASM and IASM are analogous (p value is 0.407). Furthermore, FGTS, RTS, NASM and IASM present similar performances when population is 50.

Table 4.5. The pairwise results of Wilcoxon signed rank test (pop size = 50)

	P values														
Mthds	ASM	IASM	NASM	INASM	OA	N-OA	BMT	ST	CS	FGTS	RTS	UTS			
ASM	1	2.65E-03	1,18E-03	4,18E-03	1,18E-03	1,18E-03	3,60E-03	**1,93E-01**	4,18E-03	8,40E-02	4,42E-02	**3,56E-01**			
IASM		1	**4,07E-01**	2,17E-02	3,12E-02	2,93E-04	1,40E-03	**1,93E-01**	2,93E-04	**6,19E-01**	**2,87E-01**	8,61E-03			
NASM			1	4,42E-02	3,12E-02	5,62E-03	8,61E-03	**3,81E-01**	2,93E-04	**1,49E-01**	**2,46E-01**	2,77E-02			
INASM				1	**1,63E-01**	2,93E-04	1,18E-03	2,65E-03	2,93E-04	3,09E-03	3,60E-03	2,93E-04			
OA					1	2,93E-04	3,09E-03	6,49E-03	2,93E-04	1,93E-03	3,09E-03	2,93E-04			
N-OA						1	2,29E-02	2,93E-04	2,93E-04	2,93E-04	4,38E-04	2,93E-04			
BMT							1	2,93E-04	2,93E-04	1,18E-03	4,38E-04	2,93E-04			
ST								1	2,93E-04	**3,56E-01**	**4,35E-01**	4,85E-03			
CS									1	2,93E-04	2,93E-04	2,93E-04			
FGTS										1	**7,23E-01**	1,00E-03			
RTS											1	7,13E-04			
UTS												1			

Table 4.6. The pairwise results of Wilcoxon signed rank test (pop size = 100)

	P values														
Mthds	ASM	IASM	NASM	INASM	OA	N-OA	BMT	ST	CS	FGTS	RTS	UTS			
ASM	1	2,65E-03	9,88E-03	6,49E-03	8,46E-04	2,93E-04	2,26E-03	5,52E-02	**9,81E-01**	4,95E-02	2,65E-03	**3,32E-01**			
IASM		1	**1,77E-01**	1,13E-02	**4,92E-01**	4,21E-04	4,38E-04	**2,15E-01**	1,00E-03	**2,55E-01**	**9,59E-01**	1,18E-03			
NASM			1	4,95E-02	1,91E-02	2,93E-04	1,48E-02	**4,92E-01**	3,95E-02	**2,46E-01**	**1,02E-01**	3,52E-02			
INASM				1	**2,27E-01**	**1,77E-01**	4,18E-03	1,13E-02	2,93E-04	9,88E-03	2,45E-02	2,93E-04			
OA					1	2,93E-04	7,59E-02	**3,32E-01**	2,93E-04	**6,87E-01**	**1,49E-01**	2,93E-04			
N-OA						1	**1,49E-01**	2,93E-04	2,93E-04	2,93E-04	2,93E-04	2,93E-04			
BMT							1	4,38E-04	2,93E-04	4,38E-04	5,23E-03	2,93E-04			
ST								1	2,65E-03	1,07E-02	**2,55E-01**	5,99E-04			
CS									1	5,03E-04	2,93E-04	**9,06E-01**			
FGTS										1	**5,69E-01**	2,93E-04			
RTS											1	2,93E-04			
UTS												1			

Table 4.7. The pairwise results of Wilcoxon signed rank test (pop size = 200)

	P values													
Mthds	ASM	IASM	NASM	INASM	OA	N-OA	BMT	ST	CS	FGTS	RTS	UTS		
ASM	1	1,40E-03	1,48E-02	2,93E-04	2,93E-04	2,93E-04	2,93E-04	9,88E-03	3,52E-02	1,48E-02	1,64E-03	**9,81E-01**		
IASM		1	7,48E-03	6,71E-03	**1,13E-01**	**9,06E-01**	5,57E-02	2,17E-02	2,93E-04	1,91E-02	**1,36E-01**	2,93E-04		
NASM			1	2,93E-04	1,13E-02	2,93E-04	2,93E-04	6,84E-02	3,95E-02	6,84E-02	2,65E-03	6,84E-02		
INASM				1	1,91E-02	2,10E-02	**7,20E-01**	2,65E-03	2,93E-04	3,09E-03	2,93E-04	2,93E-04		
OA					1	2,93E-04	1,64E-03	**3,81E-01**	2,93E-04	**6,53E-01**	3,95E-02	2,93E-04		
N-OA						1	3,09E-03	6,84E-02	2,93E-04	**1,02E-01**	4,85E-03	2,93E-04		
BMT							1	1,93E-03	2,93E-04	1,93E-03	2,93E-04	2,93E-04		
ST								1	2,93E-04	**8,08E-01**	**2,46E-01**	2,93E-04		
CS									1	2,93E-04	2,93E-04	2,65E-03		
FGTS										1	**2,10E-01**	2,93E-04		
RTS											1	2,93E-04		
UTS												1		

Based on the p values in Table 4.6, ASM, CS and UTS show analogous performances when population size is 100. Moreover, IASM has similarity with NASM, OA, ST, FGTS and RTS in the manner of performance. While INASM presents similar performance with OA and N-OA, the performance of BMT is only similar to the performance of N-OA.

The results of Table 4.7 can be interpreted as; the performance of ASM is only similar with the performance of UTS. IASM has similarity with OA, N-OA and RTS. Besides, while the performance of OA is similar to the performance of ST and FGTS, the performance of INASM is analogous to the performance of BMT. Also, ST demonstrates similar performance with FGTS, RTS and OA.

As a conclusion, Wilcoxon Signed Rank Test shows that the performance of BMT is almost unique comparing to other methods (excepts cases that were mentioned above).

4.6. Results by Meta Optimization Techniques

Population size, tournament size, crossover probability and mutation rate are the most important parameters of GAs as they directly affect the efficiency of the algorithm; therefore, it is essential to tune them. For this purpose, each parameter configuration of GAs is initially performed by Brute Force Search based on their following limits: population sizes = {25, 50, 100, 200}, tournament size = {3, 4, 5, 6}, crossover probability = 0.25 to 1 with 0.05 step, mutation probability = 0.025 to 0.1 with 0.005 step in 300 iterations. The tests are repeated with 25 runs that have 25 different random seeds to minimize the effects of randomization. Thus, in total, nearly “30 million” operations were made. (4 operations for population sizes, 4 operations for tournament sizes, 16 operations for crossover probability, 16 operations for mutation rate, 25 random seed and 300 iterations) Indeed, 4096 parameter configurations were efficiently tested since median values of 25 runs were represented in the results (it was accepted that the number of iterations is fixed).

Even if Brute Force Search offers the desired results, it obviously causes large computational budget. On the other hand, Coarse to Fine Search, GAs, PSO and BMT algorithms were applied with only 400 parameter configurations to reduce computational budget, which means only 10% of parameter configurations of Brute Force Search was utilized.

Each parameter configurations that were employed in Brute Force Search is defined as an individual that having four genes where each gene represents one parameter value: population size, tournament size, crossover probability and mutation rate. For the initial populations, 20 individuals were randomly created in each meta techniques. Afterwards, their processes are carried out with 20 population sizes and 20 iterations (i.e. 400 configurations).

For the GA optimizer, the best parameter setting that was obtained by Brute Force Search algorithm was used to tune itself. The parameter configuration of BMT was also found by Brute Force Search and applied in (Oral and Gençal 2017b). However, standard PSO setting (Eberhart and Kennedy 1995) was utilized to adjust GAs.

Table 4.8 demonstrates that the recommended parameter settings by Brute Force Search for each benchmark functions. It is apparent to say that if population size is high, GAs whose selection method is ST work well. Furthermore, it is suggested that the tournament size value should be taken as 3 and the crossover probability should be higher than 0.6. Nevertheless, the mutation probability is unstable for each benchmark functions.

Table 4.9 presents the suggested parameter configurations by Coarse to Fine Search algorithm. Not surprisingly, Coarse to Fine Search achieves the best results where population size is high, tournament size is generally 3, and variety of crossover and mutation probabilities are identified.

The advised parameter settings by standard GA are shown in Table 4.10. As in Table 4.8, the GA optimizer recommends to take higher population size, tournament size 3 with variety of crossover and mutation probabilities.

According to Table 4.11, PSO does not give reasonable parameter configurations in the function of Langermann. However, it presents better performance when population size is high and other parameters are varied.

On the other hand, BMT optimizer, in Table 4.12, reaches its best results when crossover probability is higher or equal to 0.6, population size 200 and tournament size is frequently 3.

Table 4.8. Parameter settings recommended by Brute Force Search (BFS)

Function	***Crossover prob.***	***Population size***	***Tournament size***	***Mutation rate***
f_1	0.6	200	3	0.025
f_2	0.8	200	3	0.025
f_4	0.65	200	3	0.025
f_6	0.7	200	3	0.045
f_7	0.65	200	3	0.08
f_8	0.8	200	3	0.06
f_{10}	0.6	200	3	0.025
f_{11}	0.95	200	3	0.055
f_{13}	1	200	5	0.08
f_{14}	0.95	200	3	0.025
f_{15}	0.8	200	3	0.025
f_{16}	0.7	200	3	0.025
f_{17}	0.85	200	3	0.045
f_{18}	0.85	200	3	0.045
f_{19}	0.7	200	3	0.055
f_{20}	0.75	200	3	0.025
f_{21}	0.9	200	3	0.035

Table 4.9. Parameter settings recommended by Coarse to Fine Search (CFS)

Function	***Crossover prob.***	***Population size***	***Tournament size***	***Mutation rate***
f_1	0.3	200	3	0.1
f_2	0.8	200	3	0.08
f_4	0.65	200	3	0.025
f_6	0.8	200	3	0.03
f_7	0.65	200	3	0.085
f_8	0.75	200	3	0.09
f_{10}	0.6	200	3	0.025
f_{11}	0.95	200	3	0.055
f_{13}	1	200	5	0.08
f_{14}	0.95	200	3	0.025
f_{15}	0.75	200	3	0.025
f_{16}	0.7	200	4	0.05
f_{17}	0.85	200	3	0.025
f_{18}	0.85	200	3	0.045
f_{19}	0.7	200	4	0.055
f_{20}	0.75	200	3	0.025
f_{21}	0.85	200	3	0.075

Table 4.10. Parameter settings recommended by standard GA

Function	***Crossover prob.***	***Population size***	***Tournament size***	***Mutation rate***
f_1	0.8	200	3	0.035
f_2	0.9	200	3	0.04
f_4	0.65	200	3	0.025
f_6	0.8	200	3	0.03
f_7	0.65	200	3	0.08
f_8	0.8	200	3	0.06
f_{10}	0.85	200	3	0.095
f_{11}	0.95	200	3	0.055
f_{13}	0.95	200	4	0.05
f_{14}	0.95	200	3	0.025
f_{15}	0.8	200	3	0.025
f_{16}	0.9	200	4	0.075
f_{17}	0.9	200	3	0.04
f_{18}	0.85	200	3	0.045
f_{19}	0.6	200	3	0.085
f_{20}	0.6	200	3	0.05
f_{21}	0.9	200	3	0.085

Table 4.11. Parameter settings recommended by PSO

Function	***Crossover prob.***	***Population size***	***Tournament size***	***Mutation rate***
f_1	0.8	200	3	0.035
f_2	0.8	200	3	0.1
f_4	0.75	200	3	0.05
f_6	1	200	3	0.095
f_7	0.65	200	3	0.08
f_8	0.8	200	3	0.06
f_{10}	1	200	5	0.04
f_{11}	0.95	200	3	0.055
f_{13}	1	200	5	0.09
f_{14}	1	200	4	0.085
f_{15}	0.8	200	3	0.035
f_{16}	0.9	200	4	0.075
f_{17}	0.9	200	3	0.1
f_{18}	0.9	200	3	0.07
f_{19}	0.55	200	3	0.08
f_{20}	0.8	200	3	0.035
f_{21}	0.75	200	3	0.1

Table 4.12. Parameter settings recommended by BMT

Function	***Crossover prob.***	***Population size***	***Tournament size***	***Mutation rate***
f_1	0.85	200	3	0.08
f_2	0.9	200	3	0.04
f_4	1	200	3	0.1
f_6	0.9	200	3	0.06
f_7	0.65	200	3	0.08
f_8	0.9	200	3	0.06
f_{10}	0.9	200	4	0.075
f_{11}	0.8	200	3	0.08
f_{13}	1	200	5	0.08
f_{14}	0.85	200	3	0.03
f_{15}	0.8	200	3	0.025
f_{16}	0.85	200	4	0.095
f_{17}	0.9	200	3	0.04
f_{18}	0.9	200	3	0.095
f_{19}	0.65	200	3	0.075
f_{20}	0.8	200	3	0.05
f_{21}	0.8	200	3	0.025

Table 4.13. Test results of all meta techniques

Function	***BFS***	***CFS***	***GA***	***PSO***	***BMT***
f_1	8.881e-16	7.993e-15	1.154e-14	1.154e-14	**4.440e-15**
f_2	1.70e-153	7.555e-30	**1.674e-47**	5.346e-32	**1.674e-47**
f_4	0.3979	**0.3979**	**0.3979**	**0.3979**	**0.3979**
f_6	2.81e-164	1.561e-30	1.561e-32	1.051e-21	**1.552e-34**
f_7	0.998	**0.998**	**0.998**	**0.998**	**0.998**
f_8	-1	**-1**	**-1**	**-1**	**-1**
f_{10}	0	**0**	0.0015	0.002	0.0013
f_{11}	3	**3**	**3**	**3**	**3**
f_{13}	-4.1	**-4.1545**	**-4.1554**	**-4.1557**	**-4.1557**
f_{14}	-1.8013	**-1.8013**	**-1.8013**	**-1.8013**	**-1.8013**
f_{15}	0	**0**	**0**	**0**	**0**
f_{16}	0	2.550e-06	1.226e-06	1.226e-06	**7.654e-07**
f_{17}	3.28e-150	1.737e-24	**5.741e-39**	4.523e-29	**5.745e-39**
f_{18}	-210.4823	**-210.4823**	**-210.4823**	**-210.4823**	**-210.4823**
f_{19}	-837.9658	**-837.9658**	**-837.9658**	**-837.9658**	**-837.9658**
f_{20}	-1.0316	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**
f_{21}	2.56e-155	**1.099e-85**	4.032e-29	5.519e-29	1.643e-36

4.7. Discussion for the Meta Tests

The second aim of this study was to investigate the parameter settings of GAs by operating common search techniques (Brute Force Search and Coarse to Fine Search) and meta optimizers (GAs, PSO, BMT).

According to the all obtained results, it is suggested that the tournament size value should be 3, the crossover probability should be greater than or equal to 0.6, and population size should be higher, if applicable. Furthermore, variety of mutation size probabilities can be operated.

As Table 4.13 reveals, Brute Force Search still offers the best performance if computational budget is plausible. Otherwise, meta techniques should be operated to tune standard GAs, except PSO.

Coarse to Fine Search is the only winner Griewangk and Sum of Different Powers functions as it finds the global value or near to the global value. However, it does not achieve the global values in the six functions: Ackley, Axis Parallel Hyper-Ellipsoid, De Jong, Langermann, Rosenbrock's Valley and Rotated Hyper Ellipsoid.

On the other hand, there is no function that GA optimizer is the only winner. Nevertheless, it presents decent performance except Ackley, De Jong, Langermann, Griewangk, Rotated Hyper Ellipsoid and Sum of Different Powers functions.

Since PSO was not tuned itself, it does not present effective results comparing to others, shown in Table 4.13.

BMT optimizer is the only winner in three functions: Ackley, De Jong and Rosenbrock's Valley. Moreover, BMT presents powerful solutions in fifteen functions out of seventeen. Hence, **it proved itself as a robust meta optimizer for tuning standard GA.**

5. CONCLUSIONS

The main purpose of the study is to improve the performance of GAs. Generally, two approaches have been performed for this purpose: introducing a selection method, utilizing a meta optimization technique on the selection method that already proposed.

In the first part of the study, seven new selection methods, Aggressive, Non-Aggressive, Integrated Aggressive, Integrated Non-Aggressive (INASM), Outlander, Non-Aggressive Outlander (N-OA), Bipolar Mating Tendency (BMT), were presented. Besides, twenty-one test functions were utilized in order to test the performance of them. Their performances were compared with Standard Tournament Selection (ST) and prevalent selection methods that are based on ST: Restricted Tournament Selection (RTS), Unbiased Tournament Selection (UTS), Fine-Grained Tournament Selection (FGTS) and Cooperative Selection (CS).

Based on the results of the first part, N-OA, INASM and BMT provide better performance comparing to all selection method that were carried out. N-OA introduces its best performance in small population sizes while INASM especially has the achievements in the large population sizes. However, BMT offers the best performance in all population sizes. Furthermore, it is successful in forty-nine conditions out of sixty-three conditions in addition to being only the winner in twenty-one conditions. It is selected as a meta optimizer for the second part of study due to its accomplishment in the first part.

In the second part of the study, meta search techniques; Brute Force Search and Coarse to Fine Search, and meta optimization algorithms; GAs, Particle Swarm Optimization and BMT, were operated to tune standard GAs in order to obtain the best performance of the GAs. According to the results, BMT is a robust meta optimizer comparing to GA and PSO.

Consequently, it is obvious that the study has achieved its purpose since it produces significant algorithms that outperform a standard GA: BMT, N-OA and INASM.

As a future work, hybrid versions of BMT, N-OA and INASM should be implemented since they offer better performances in the first part of the study. Moreover, the performance of BMT will be tested in a real time problem.

REFERENCES

Aburomman, A. A., and M. Bin Ibne Reaz. 2016. A novel SVM-kNN-PSO ensemble method for intrusion detection system. Applied Soft Computing Journal, 38: 360-372.

Adenso-Díaz, B., and M. Laguna. 2006. Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search. Operations Research, 54.1: 99-114.

Ali, E. E. E., and E. Elamin. 2006. A proposed genetic algorithm selection method. King Saud University, College of Computer and Information Sciences.

Ansótegui, C., M. Sellmann, and K. Tierney. 2009. A gender-based genetic algorithm for the automatic configuration of algorithms. In: International Conference on Principles and Practice of Constraint Programming. Springer, Berlin, Heidelberg, 142-157.

Bäck, T. 1994. Parallel optimization of evolutionary algorithms. In: International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 418-427.

Baker, J. 1987. Reducing bias and inefficiency in the selection algorithm. 2nd International Conference on Genetic Algortihms and their Applications, Vol. 206, 14-21.

Balaprakash, P., M. Birattari, and T. Stuetzle. 2007. Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement. In: International workshop on hybrid metaheuristics. Springer, Berlin, Heidelberg, 108-122.

Bartz-Beielstein, T., C. W. G. Lasarczyk, and M. Preuß. 2005. Sequential parameter optimization. Evolutionary Computation. The 2005 IEEE Congress on, 773–780.

Bartz, T., and T. Bartz-Beielstein. 2003. Experimental analysis of evolution strategies: Overview and comprehensive introduction. Secretary of the SFB

531.

Birattari, M., T. Stützle, L. Paquete, and K. Varrentrapp. 2002. A Racing Algorithm for Configuring Metaheuristics. GECCO 2002 Proceedings of the Genetic and Evolutionary Computation Conference, 11-18.

Birattari, M., Z. Yuan, P. Balaprakash, and T. Stützle. 2010. F-race and iterated F-race: An overview. Experimental Methods for the Analysis of Optimization Algorithms, 311-336.

Blickle, T., and L. Thiele. 1995. A Comparison of Selection Schemes used in Genetic Algorithms. TIK-Report.

Branke, J., and J. Elomari. 2012. Meta-Optimization for Parameter Tuning with a Flexible Computing Budget. Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference, 1245-1252.

Camilleri, M., and F. Neri. Evolutionary Parameter Meta-Optimization in Decision Tree Learning.

Coley, D. A. 1999. An Introduction to Genetic Algorithms for Scientists and Engineers.

Cortez, P., M. Rocha, and J. Neves. 2001. A meta-genetic algorithm for time series forecasting.

Davis, L. D., and M. Mitchell. 1991. Handbook of Genetic Algorithms. Page Computer.

Deb, K., and D. E. Goldberg. 1989. An investigation of niche and species formation in genetic function optimization. *in* C. Morgan Kaufmann, San Francisco, editor. the 3rd International Conference on Genetic Algorithms (ICGA'89). George Mason University, Fairfax, VA, 42–50.

Eberhart, R., and J. Kennedy. 1995. A new optimizer using particle swarm theory. Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, 39–43.

Eiben, a. E., R. Hinterding, and Z. Michalewicz. 1999. Parameter control in

evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 3.2: 124-141 .

Eiben, A. E., and J. E. Smith. 2015. Introduction to Evolutionary Computing. New York, Vol. 53, p. 18.

Erik, M. 2010. Tuning & Simplifying Heuristical Optimization. PhD Thesis. University of Southampton.

Eshelman, L., and J. Schaffer. 1991. Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. Proceedings of the 3rd International Conference on Genetic Algorithm, 115–122.

Farias, R. G. G., and C. S. De Magalhães. 2018. Parent Selection Strategies in Niching Genetic Algorithms. 2018 IEEE Congress on Evolutionary Computation, CEC 2018 – Proceedings, 1-8.

Filipović, V. 2003. Fine-grained tournament selection operator in genetic algorithms. Computing and Informatics, 22.2: 143-161.

Francesca, G., P. Pellegrini, T. Stützle, and M. Birattari. 2011. Off-line and on-line tuning: A study on operator selection for a memetic algorithm applied to the QAP. In: European Conference on Evolutionary Computation in Combinatorial Optimization. Springer, Berlin, Heidelberg, 203-214.

François, O., and C. Lavergne. 2001. Design of evolutionary algorithms - A statistical perspective. IEEE Transactions on Evolutionary Computation, 5.2: 129-148.

Freisleben, B., and M. Härtfelder. 1993. Optimization of genetic algorithms by genetic algorithms. Artificial Neural Nets and Genetic Algorithms, 392–399.

Friedman, M. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. Journal of the American Statistical Association, 32:200, 675-701 .

Friedman, M. 2007. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. The Annals of Mathematical Statistics, 11.1:

86-92.

Gençal, M. C., and M. Oral. 2017. Outlander Algorithm Based on Integrated Aggressive Selection Method. International Journal of Science and Research (IJSR) 6:1673–1679.

Goh, K. S., A. Lim, and B. Rodrigues. 2003. Sexual selection for genetic algorithms. Artificial Intelligence Review, 19.2: 123-152.

Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley.

Goldberg, D. E. 1990. A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing. Complex Systems, 4.4: 445-460 .

Grefenstette, J. J. 1986. Optimization of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics, 16.1: 122-128 .

Grefenstette, J. J., and J. Baker. 1989. How genetic algorithms work: A critical look at implicit parallelism. Genetic Algorithm and Their Applications: Proceedings of Third International Conference of Genetic Algorithm.

Haigh, J., and W. J. Conover. 2006. Practical Nonparametric Statistics. Journal of the Royal Statistical Society. Series A (General).

Harik, G. 1995. Finding multimodal solutions using restricted tournament selection. Proceedings of the Sixth International Conference on Genetic Algorithms, 24-31.

Hinterding, R. 1995. Gaussian mutation and self-adaption for numeric genetic algorithms. Evolutionary Computation, 1995., IEEE International Conference on, 384.

Hinterding, R., Z. Michalewicz, and A. E. Eiben. 1997. Adaptation in evolutionary computation: A survey. Transations on Evolutionary Computation, 65-69.

Holland, J. H. 1975. Adaptation in Natural and Artificial Systems: An introductory Analysis with Applications to Biology, Control and Artificial Intelligence.

MIT Press.

Hutter, F., H. H. Hoos, and K. Leyton-Brown. 2011. Sequential model-based optimization for general algorithm configuration. In: International conference on learning and intelligent optimization. Springer, Berlin, Heidelberg, 507-523.

Hutter, F., H. H. Hoos, K. Leyton-Brown, and K. P. Murphy. 2009a. An Experimental Investigation of Model-Based Parameter Optimisation: SPO and Beyond. Proceedings of the 11th Annual conference on Genetic and evolutionary computation, 271-278.

Hutter, F., H. H. Hoos, K. Leyton-Brown, and T. Stützle. 2009b. ParamILS: An automatic algorithm configuration framework. Journal of Artificial Intelligence Research, 36: 267-306.

Hutter, F., F. Hutter, H. H. Hoos, and H. H. Hoos. 2007. Automatic Algorithm Configuration based on Local Search. BMC Bioinformatics. Vol. 7, pp. 1152-1157.

Jiménez Laredo, J. L., S. S. Nielsen, G. Danoy, P. Bouvry, and C. M. Fernandes. 2014. Cooperative Selection: Improving Tournament Selection via Altruism. In: European Conference on Evolutionary Computation in Combinatorial Optimization. Springer, Berlin, Heidelberg, 85-96.

Kafadar, K., and D. J. Sheskin. 2006. Handbook of Parametric and Nonparametric Statistical Procedures.

Keane, A. J. 1995. Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness. Artificial Intelligence in Engineering, 9.2: 75-83.

de la Maza, M., and B. Tidor. 1992. Increased flexibility in genetic algorithms: The use of variable Boltzmann selective pressure to control propagation. Computer Science and Operations Research. Elsevier, 425–440.

Lobo, F. G., and M. Bazargani. 2015. When Hillclimbers Beat Genetic Algorithms in Multimodal Optimization.

Lobo, F. G., and D. E. Goldberg. 2004. The parameter-less genetic algorithm in practice. Information Sciences, 167.1-4: 217-232.

Mason, K., J. Duggan, and E. Howley. 2018. A meta optimisation analysis of particle swarm optimisation velocity update equations for watershed management learning. Applied Soft Computing Journal, 62: 148-161.

Matrenin, P. V., and V. G. Sekaev. 2015. Particle Swarm optimization with velocity restriction and evolutionary parameters selection for scheduling problem. 2015 International Siberian Conference on Control and Communications, SIBCON 2015 – Proceedings, 1-5.

Matsui, K. 1999. New selection method to improve the population diversity in genetic algorithms. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028). Vol. 1, pp. 625-630.

Mauldin, M. L. 1984. Maintaining diversity in genetic search. Proceedings of the national conference on artificial intelligence, 247–250.

Meissner, M., M. Schmuker, and G. Schneider. 2006. Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics, 7.1: 125.

Mercer, R. E., and J. R. Sampson. 1978. Adaptive search using a reproductive meta-plan. Kybernetes, 7.3: 215-228.

Meyer-Nieberg, S., and H. G. Beyer. 2007. Self-adaptation in evolutionary algorithms. Parameter setting in evolutionary algorithms. Springer, Berlin, Heidelberg, 47-75

Mitchell, M. 1996. An introduction to genetic algorithms. Computers & Mathematics with Applications.

Molga, M., and C. Smutnicki. 2005. Test functions for optimization needs. Test functions for optimization needs.

Mouret, J.-B., and S. Doncieux. 2012. Encouraging behavioral diversity in evolutionary robotics: an empirical study. Evolutionary computation, 20.1:

91-133.

Nannen, V., and A. E. Eiben. 2006. A method for parameter calibration and relevance estimation in evolutionary algorithms. Proceedings of the 8th annual conference on Genetic and evolutionary computation, 183-190.

Oral, M., and M. Gençal. 2016. A Novel Selection Methods for Genetic Algorithms. IMSEC 1st International Mediterranean SCIENCE AND ENGINEERING CONGRESS. Adana, 254–258.

Oral, M., and M. C. Gençal. 2017a. Improved aggressive and integrated aggressive selection methods for genetik algorithms. 2017 International Conference on Computer Science and Engineering (UBMK), 304–309.

Oral, M., and M. C. Gençal. 2017b. Harmony Between the Best and the Worst Individuals in Tournament Selection. Majlesi Journal of Mechatronic Systems 6:25–28.

Oral, M., and M. C. Gençal. 2018. Application of Outlander Algortihm on Non-Aggressive Selection Methods. Çukurova University Journal of the Faculty of Engineering and Architecture 33:31–40.

Pedersen, M. E. H., and A. J. Chipperfield. 2008. Tuning differential evolution for artificial neural networks. HL0803. Hvass Laboratories.

Pedersen, M. E. H., and A. J. Chipperfield. 2010. Simplifying Particle Swarm Optimization. Applied Soft Computing, 10.2: 618-628.

Pellegrini, P., T. Stützle, and M. Birattari. 2010. Off-line vs. On-line tuning: A study on MAX - MIN ant system for the TSP. In: Swarm Intelligence: 7th International Conference Proceedings. Springer, p. 239

Pohlheim, H. 1995. The multipopulation genetic algorithm: Local selection and migration. Technical report, Technical University Ilmenau.

Roy, R., and I. C. Parmee. 1996. Adaptive restricted tournament selection for the identification of multiple sub-optima in a multi-modal function. In: AISB Workshop on Evolutionary Computing. Springer, Berlin, Heidelberg, p. 236-256.

Shopova, E. G., and N. G. Vaklieva-Bancheva. 2006. BASIC—A genetic algorithm for engineering problems solution. Computers & Chemical Engineering, 30.8: 1293-1309.

Smit, S. K., and A. E. Eiben. 2009. Comparing parameter tuning methods for evolutionary algorithms. 2009 IEEE Congress on Evolutionary Computation, CEC 2009, 399-406.

Sokolov, A., F. Collins, and D. Whitley. 2005. Unbiased tournament selection. GECCO 2005 Proceedings of the Genetic and Evolutionary Computation Conference, 1131–1138.

Staelin, C. 2003. Parameter selection for support vector machines. Hewlett-Packard Company, HPL-2002-354 (R.1).

Surjanovic, S., and D. Bingham. 2013. Virtual Library of Simulation Experiments: Test Functions and Datasets. http://www.sfu.ca/~ssurjano.

Thomsen, R. 2005. Multimodal optimization using crowding-based differential evolution, 1382–1389.

Yalkin, C., and E. E. Korkmaz. 2012. Neural network world: A neural network based selection method for genetic algorithms. Neural Network World, 22.6: 495.

Yuan, B., and M. Gallagher. 2007. Combining meta-EAs and racing for difficult EA parameter tuning tasks. Studies in Computational Intelligence, 121-142.

Zhu, K. Q. 2003. Population Diversity in Genetic Algorithm for Vehicle Routing Problem with Time Windows. Department of Computer Science, National University of Singapore, 1–10.

CURRICULUM VITAE

He was born in Adana, Seyhan. He finished his primary, secondary and high school in Adana. He graduated from Faculty of Arts and Science of Çukurova University, the Department of Mathematics on June 3rd, 2005. He also got Msc. Degree from Çukurova University Science Institute, the Department of Secondary School Science and Mathematics Education in 2006. He worked as a Math teacher until he earned a scholarship from the Republic of Turkey Ministry of National Education in 2011. The scholarship offered him to study in the USA. Thus, he got Msc. Degree from Indiana University at Bloomington the Department of Computer Science in 2014. He started his PhD. Degree from Çukurova University Science Institute, the Department of Computer Engineering. He is still a research assistant at the Department of Computer Engineering in Çukurova University. His research interests include artificial intelligence and image processing.

APPENDIX

Appendix 1: The results of bipolarity values

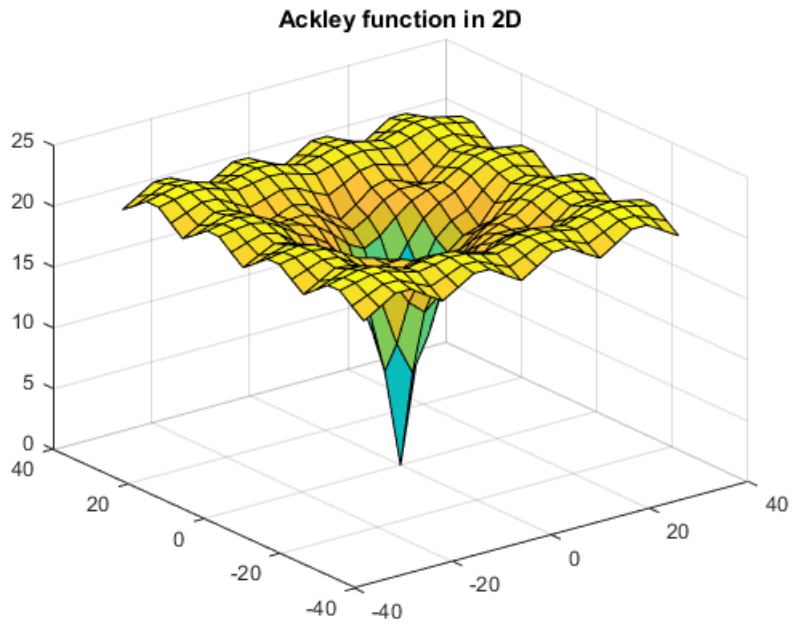
Function	Bipolarity Values (0.05 – 0.25)				
	0.05	**0.1**	**0.15**	**0.2**	**0.25**
f_1	0,000106	1,75E-05	1,38E-05	**3,81E-06**	**3,91E-06**
f_2	1,27E-09	9,17E-11	2,22E-10	1,01E-11	1,17E-12
f_3	**0,397887**	**0,397887**	**0,397887**	**0,397887**	**0,397887**
f_4	1,66E-10	2,51E-11	9,93E-12	6,50E-13	7,24E-14
f_5	**0,998004**	**0,998004**	**0,998004**	**0,998004**	**0,998004**
f_6	-0,94793	-0,93625	-0,93625	-0,93625	-0,93858
f_7	**3**	**3**	**3**	**3**	**3**
f_8	**0,007399**	**0,007396**	**0,007396**	**0,007397**	**0,007396**
f_9	-4,15581	-4,12758	-4,12758	-4,12758	**-4,10266**
f_{10}	**-1,8013**	**-1,8013**	**-1,8013**	**-1,8013**	**-1,8013**
f_{11}	**0**	**0**	**0**	**0**	**0**
f_{12}	0,002865	0,000753	0,001704	0,003191	**0,000207**
f_{13}	1,47E-08	1,13E-09	3,21E-10	1,39E-11	**9,82E-12**
f_{14}	**-210,482**	**-210,482**	**-210,482**	**-210,482**	**-210,482**
f_{15}	**-837,966**	**-837,966**	**-837,966**	**-837,966**	**-837,966**
f_{16}	**-1,03163**	**-1,03163**	**-1,03163**	**-1,03163**	**-1,03163**
f_{17}	1,23E-14	2,76E-15	4,93E-17	1,73E-18	2,90E-17

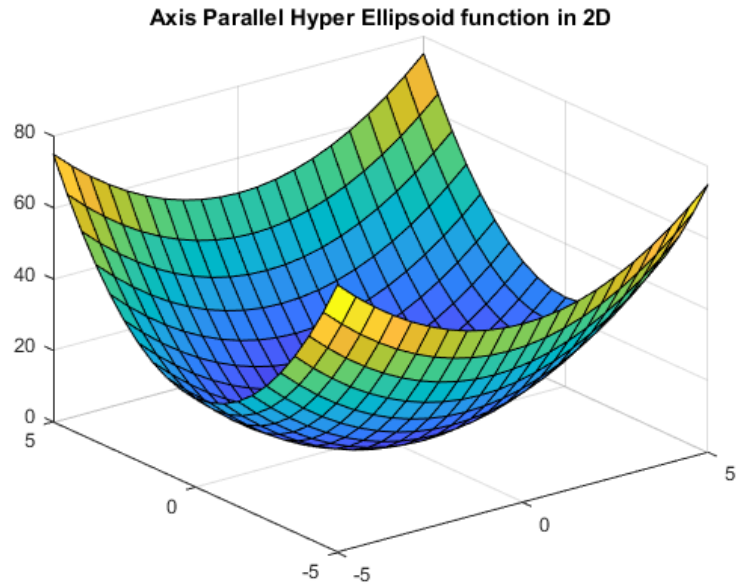
Function	Bipolarity Values (0.3 – 0.5)				
	0.3	**0.35**	**0.4**	**0.45**	**0.5**
f_1	7,01E-06	1,49E-05	4,07E-05	1,32E-05	0,000205
f_2	1,09E-11	**1,04E-13**	5,80E-12	2,53E-11	1,58E-10
f_3	**0,397887**	**0,397887**	**0,397887**	**0,397887**	**0,397888**
f_4	**9,53E-15**	1,70E-13	2,87E-15	4,14E-14	2,18E-10
f_5	**0,998004**	**0,998004**	**0,998004**	**0,998004**	**0,998004**
f_6	**-0,93625**	**-0,93625**	**-0,93625**	**-0,93625**	**-0,93625**
f_7	**3**	**3**	**3**	**3,000001**	**3**
f_8	**0,007396**	**0,007397**	**0,0074**	0,018577	**0,0074**
f_9	-4,12758	-4,12758	**-4,10086**	-4,12758	**-4,10079**
f_{10}	**-1,8013**	**-1,8013**	**-1,8013**	**-1,8013**	**-1,8013**
f_{11}	**0**	**0**	**0**	**0**	**0**
f_{12}	0,002183	0,004203	0,005228	0,005027	0,015096
f_{13}	**3,10E-12**	1,13E-10	5,88E-11	2,37E-10	5,84E-10
f_{14}	**-210,482**	**-210,482**	**-210,482**	**-210,482**	**-210,482**
f_{15}	**-837,966**	**-837,966**	**-837,966**	**-837,966**	**-837,966**
f_{16}	**-1,03163**	**-1,03163**	**-1,03163**	**-1,03163**	**-1,03163**
f_{17}	**8,62E-19**	1,37E-18	2,50E-18	8,13E-18	2,90E-17

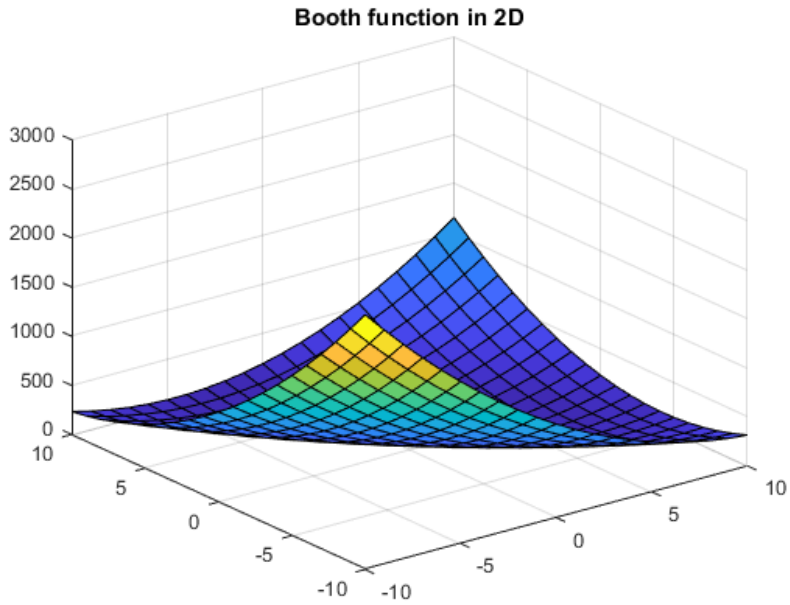
Function	Bipolarity Values (0.55 – 0.75)				
	0.55	**0.6**	**0.65**	**0.7**	**0.75**
f_1	0,000112	0,000428	0,000456	0,0047	0,008917
f_2	1,68E-09	6,15E-09	2,58E-06	1,17E-07	4,31E-05
f_3	**0,397888**	**0,397892**	**0,397907**	**0,397895**	**0,397895**
f_4	1,27E-12	1,25E-09	1,79E-08	9,69E-08	4,59E-07
f_5	**0,998004**	**0,998004**	**0,998004**	**0,998004**	**0,998004**
f_6	**-0,93625**	**-0,93625**	**-0,93625**	**-0,93625**	**-0,93625**
f_7	**3**	**3,000024**	**3,000032**	**3,000026**	**3,000004**
f_8	0,011958	0,009865	0,007764	0,016538	0,010713
f_9	**-4,10086**	-4,12757	**-4,10086**	**-4,10084**	-4,12333
f_{10}	**-1,8013**	**-1,8013**	**-1,8013**	**-1,8013**	**-1,8013**
f_{11}	**0**	**0**	**0**	**0**	**0**
f_{12}	0,002428	0,005027	0,000929	0,005319	0,004028
f_{13}	2,80E-10	7,14E-08	1,46E-06	4,29E-06	0,000135
f_{14}	**-210,478**	**-210,481**	**-210,404**	**-210,254**	**-210,27**
f_{15}	**-837,966**	**-837,965**	**-837,966**	**-837,964**	**-837,961**
f_{16}	**-1,03163**	**-1,03163**	**-1,03163**	**-1,03163**	**-1,03163**
f_{17}	9,94E-15	6,60E-13	3,23E-12	8,18E-12	1,50E-11

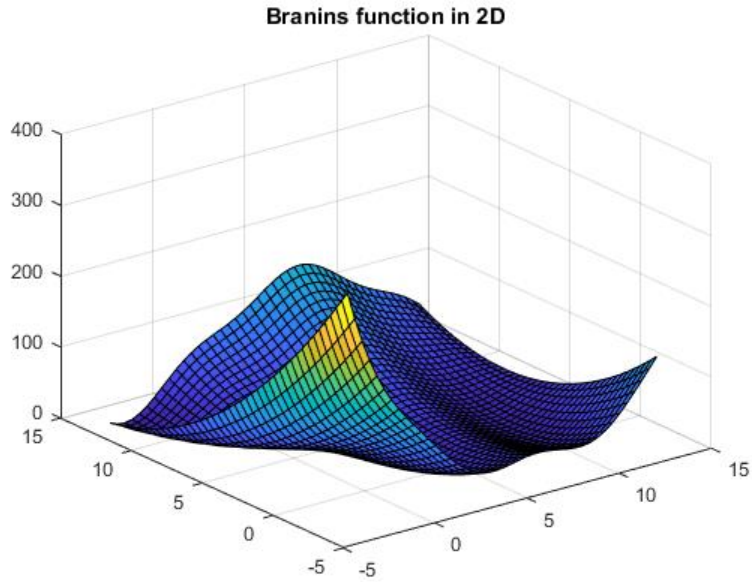
Function	Bipolarity Values (0.8 – 1)				
	0.8	**0.85**	**0.9**	**0.95**	**1**
f_1	0,008023	0,013904	0,015	0,018312	0,04112
f_2	2,64E-06	1,64E-05	0,000214	6,26E-05	0,00011
f_3	**0,398069**	**0,397957**	**0,398114**	**0,39823**	**0,398737**
f_4	1,66E-07	4,90E-08	3,23E-06	8,67E-07	3,27E-05
f_5	**0,998004**	**0,998004**	**0,998004**	**0,998005**	**0,998005**
f_6	**-0,93625**	**-0,93625**	**-0,93625**	**-0,93625**	**-0,93624**
f_7	3,000075	3,000668	3,000147	3,002901	3,00603
f_8	0,020811	0,026475	0,026045	0,026597	0,031746
f_9	-4,08913	-4,09265	-4,09572	-4,08089	-4,07876
f_{10}	**-1,8013**	**-1,80129**	**-1,8013**	**-1,8013**	**-1,80125**
f_{11}	**0**	1,48E-05	0,003986	0,000422	0,007463
f_{12}	0,004094	0,001081	0,002289	0,002541	0,005133
f_{13}	2,50E-05	8,28E-05	0,000436	0,000166	0,000899
f_{14}	**-210,288**	**-210,071**	-209,773	-209,394	-209,622
f_{15}	**-837,957**	**-837,959**	**-837,961**	**-837,954**	-837,876
f_{16}	**-1,03163**	**-1,03161**	**-1,03161**	**-1,03162**	**-1,0316**
f_{17}	5,56E-10	6,06E-11	5,36E-10	1,19E-08	1,28E-08

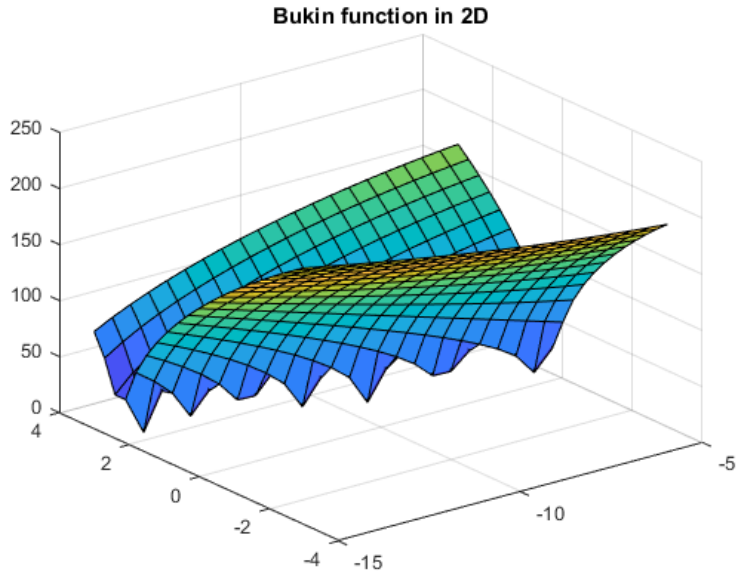
Appendix 2: The visual presentation of the test functions

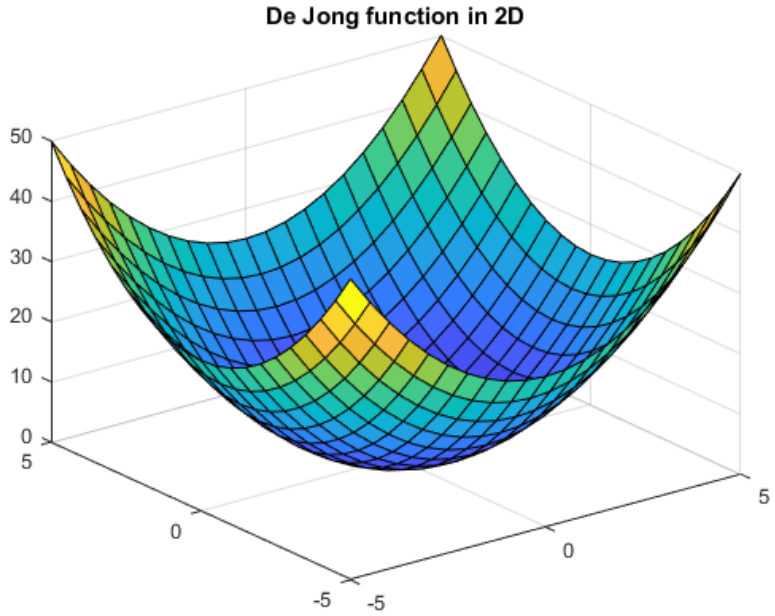


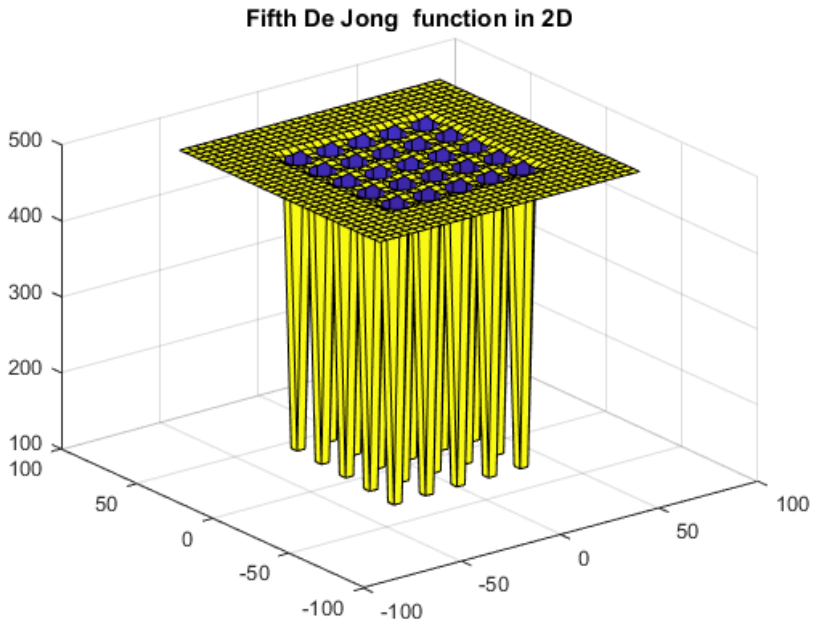


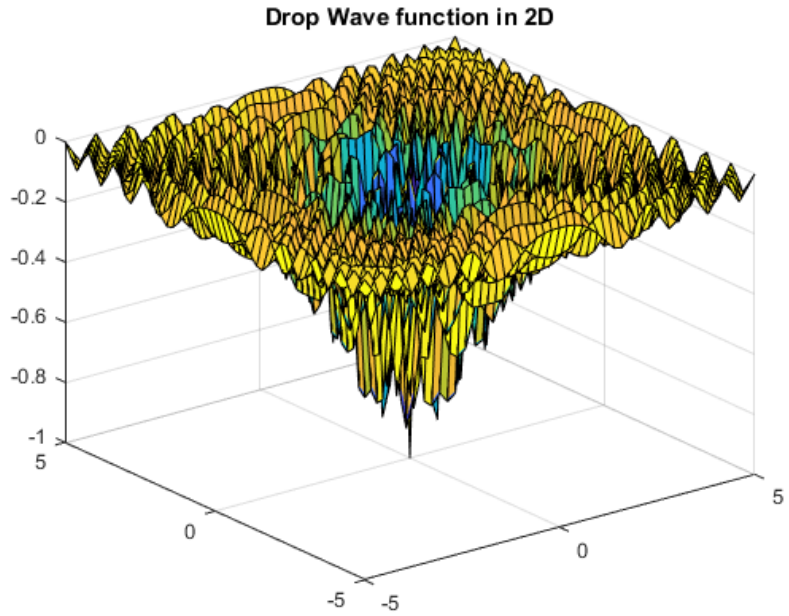

Booth function in 2D
3000
2500
2000
1500
1000
500
0
10
5
0
-5
-10
-10
-5
0
5
10

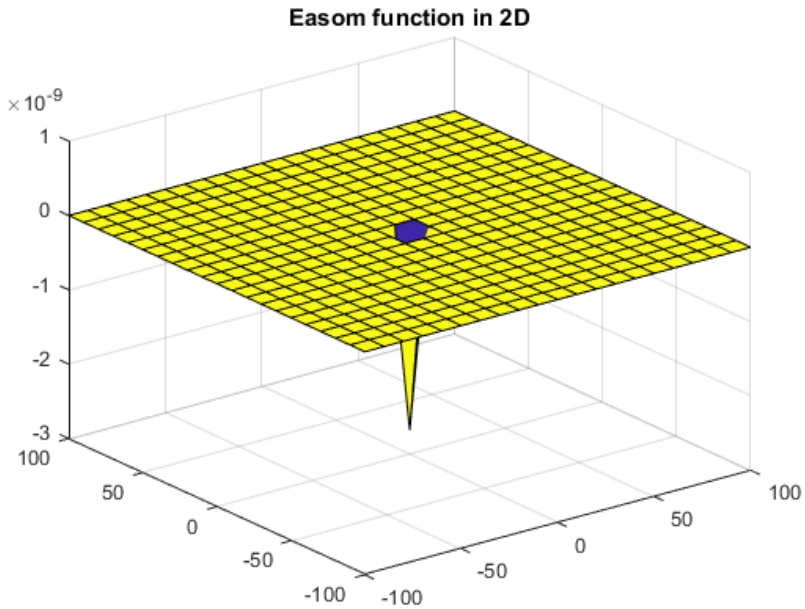

Branins function in 2D
400
300
200
100
0
15
10
5
0
-5
-5
0
5
10
15

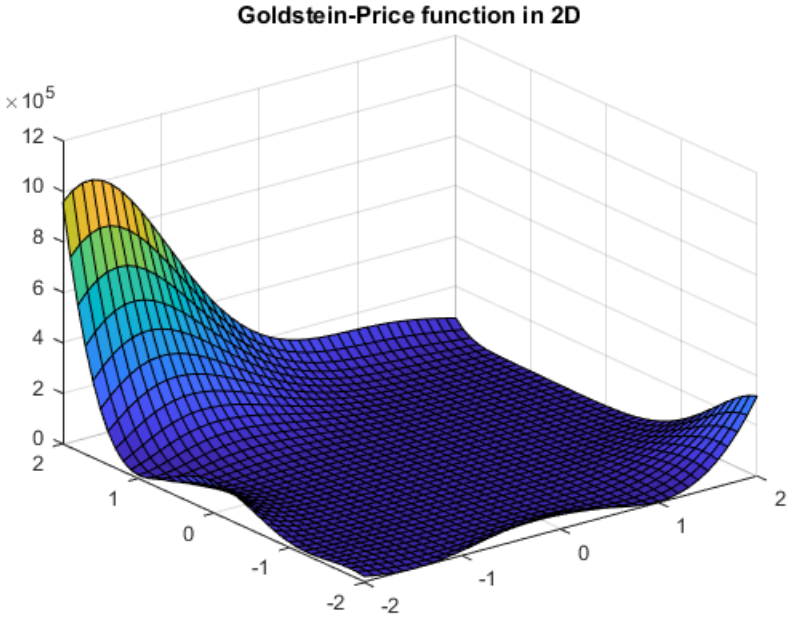

Bukin function in 2D
250
200
150
100
50
0
4
2
0
-2
-4
-15
-10
-5

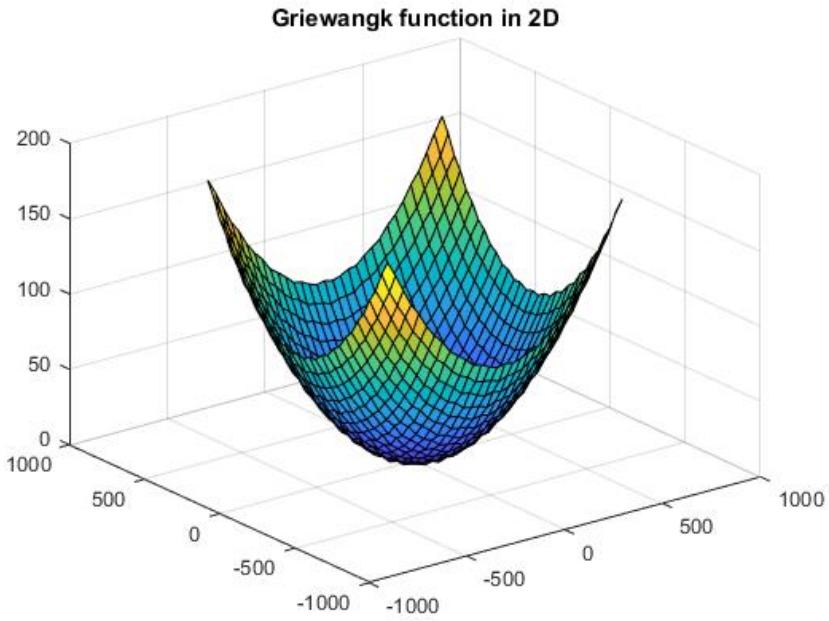

De Jong function in 2D
50
40
30
20
10
0
5
0
-5
-5
0
5

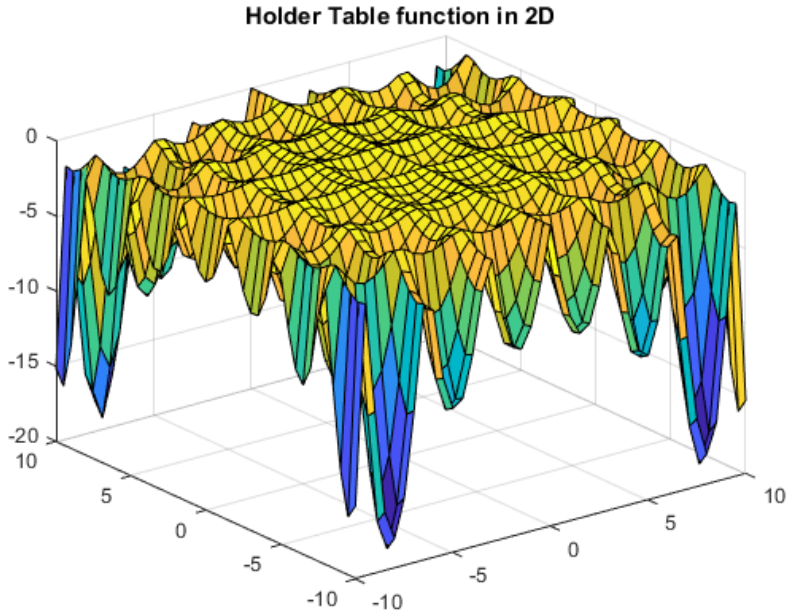

Fifth De Jong function in 2D
500
400
300
200
100
100
50
0
-50
-100
-100
-50
0
50
100

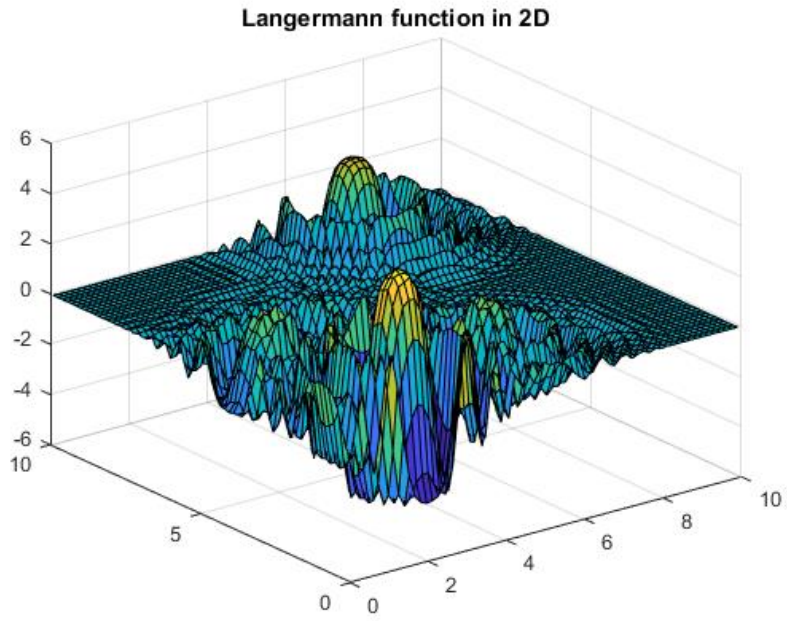

Drop Wave function in 2D
0
-0.2
-0.4
-0.6
-0.8
-1
5
0
-5
-5
0
5

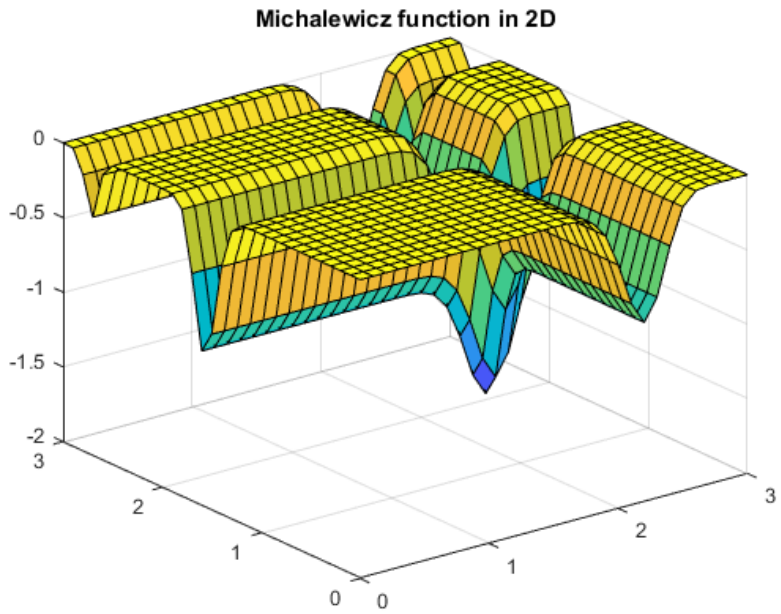

Easom function in 2D
×10^-9
1
0
-1
-2
-3
100
50
0
-50
-100
-100
-50
0
50
100

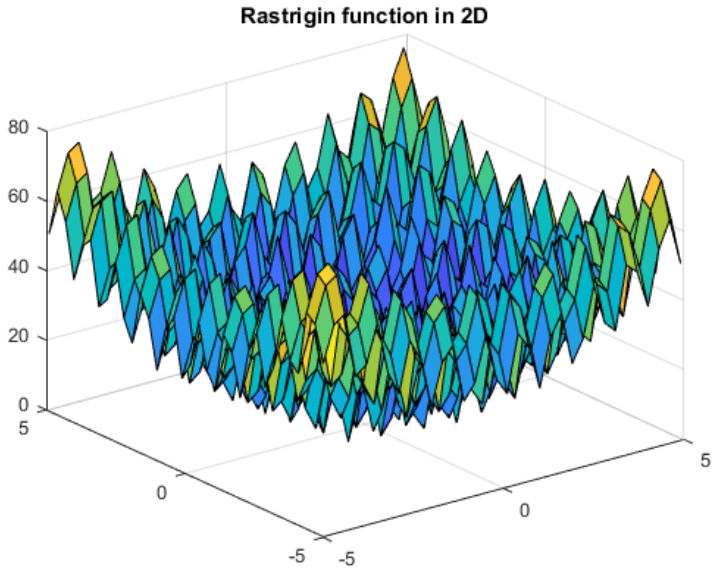

Goldstein-Price function in 2D
×10^5
12
10
8
6
4
2
0
2
1
0
-1
-2
-2
-1
0
1
2

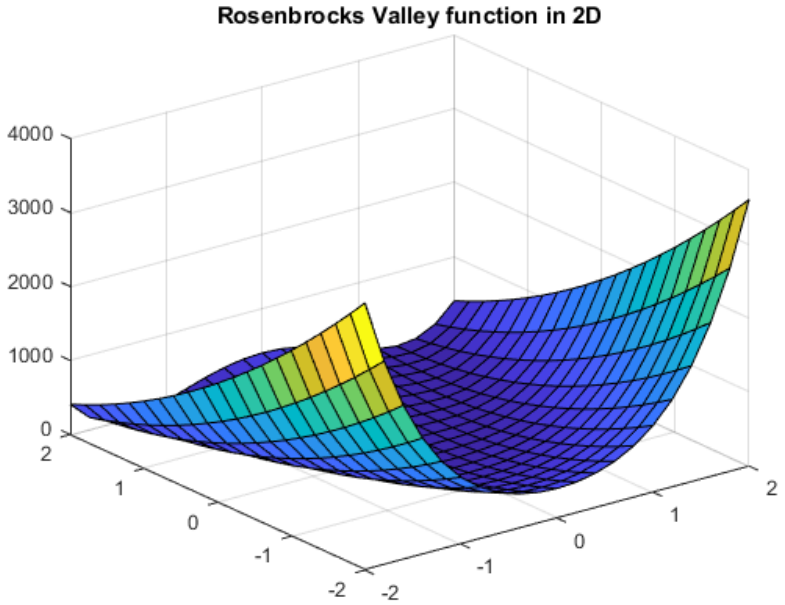

Griewangk function in 2D
200
150
100
50
0
1000
500
0
-500
-1000
-1000
-500
0
500
1000

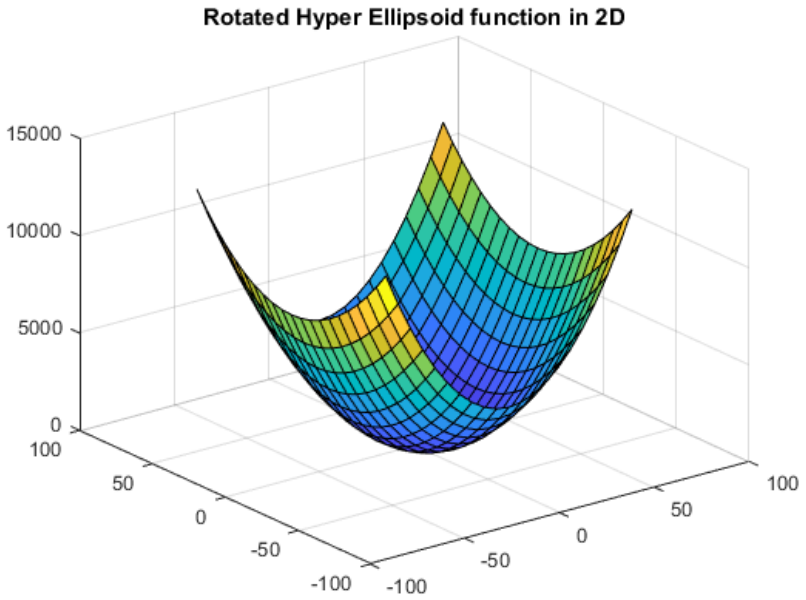

Holder Table function in 2D
0
-5
-10
-15
-20
10
5
0
-5
-10
-10
-5
0
5
10

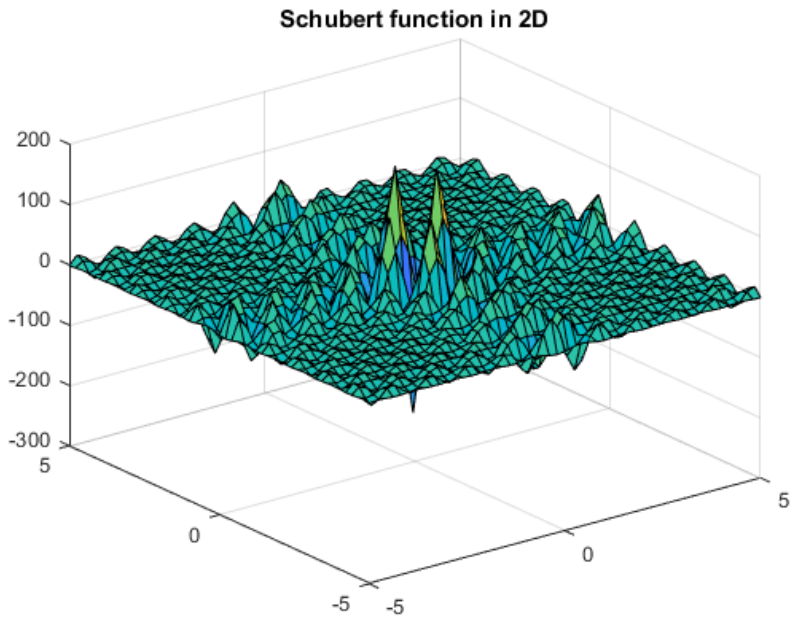

Langermann function in 2D
6
4
2
0
-2
-4
-6
10
5
0
0
2
4
6
8
10

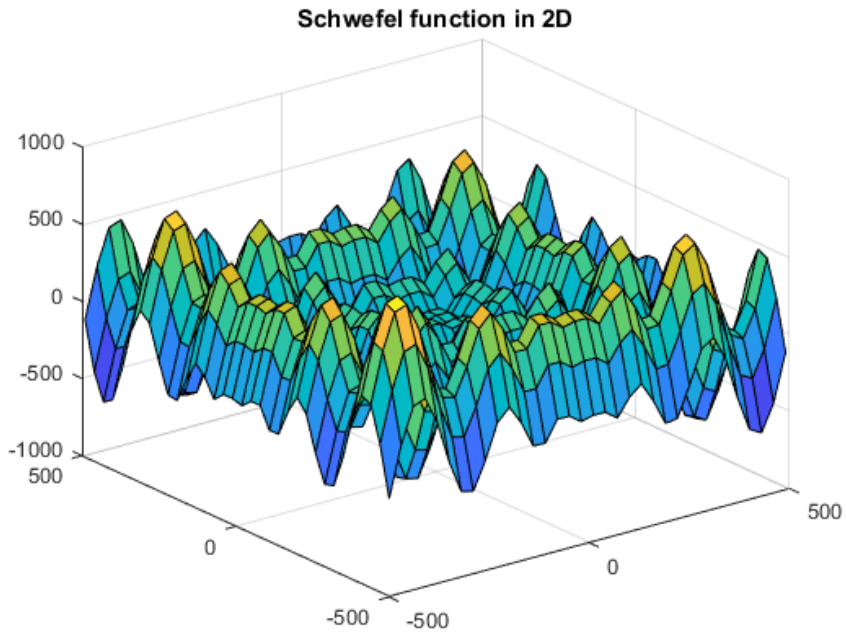

Michalewicz function in 2D
0
-0.5
-1
-1.5
-2
3
2
1
0
0
1
2
3

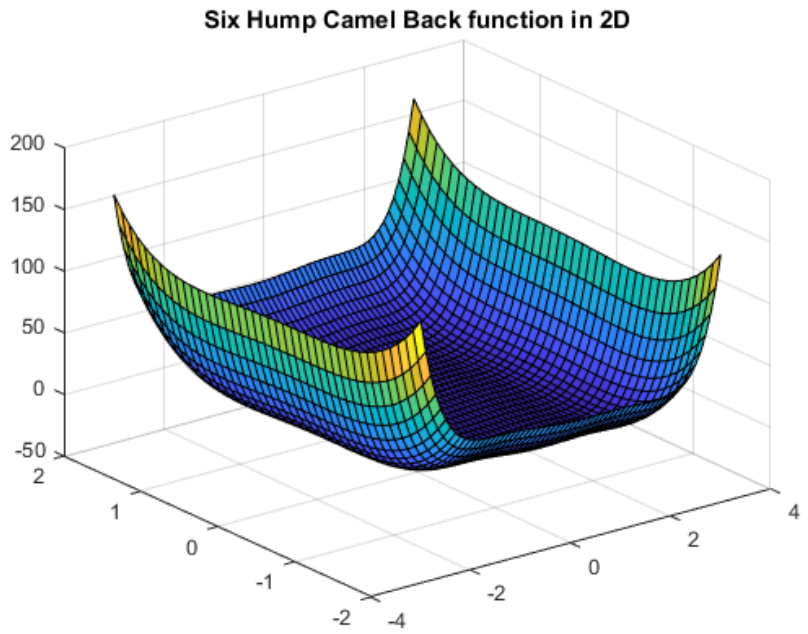

Rastrigin function in 2D
80
60
40
20
0
5
0
-5
-5
0
5

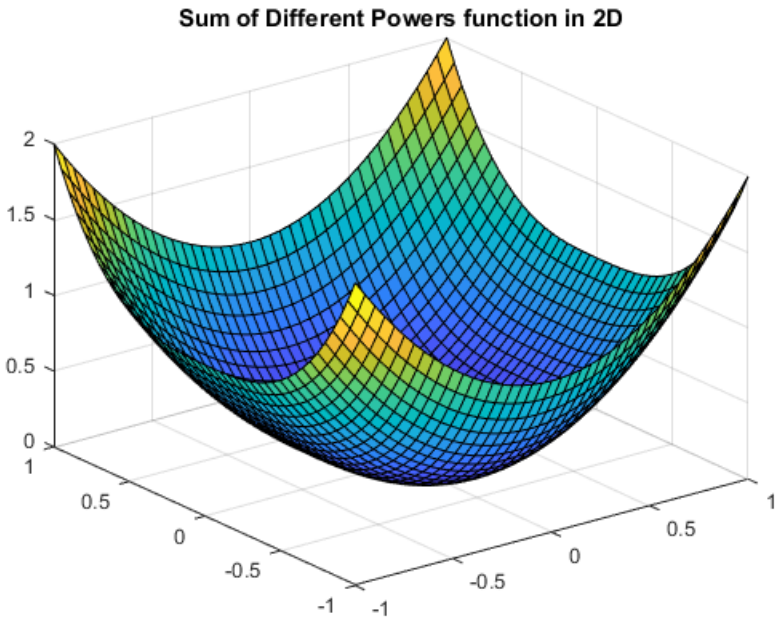

Rosenbrocks Valley function in 2D
4000
3000
2000
1000
0
2
1
0
-1
-2
-2
-1
0
1
2


Rotated Hyper Ellipsoid function in 2D
15000
10000
5000
0
100
50
0
-50
-100
-100
-50
0
50
100


Schubert function in 2D
200
100
0
-100
-200
-300
5
0
-5
-5
0
5


Schwefel function in 2D
1000
500
0
-500
-1000
500
0
-500
-500
0
500


Six Hump Camel Back function in 2D
200
150
100
50
0
-50
2
1
0
-1
-2
-4
-2
0
2
4

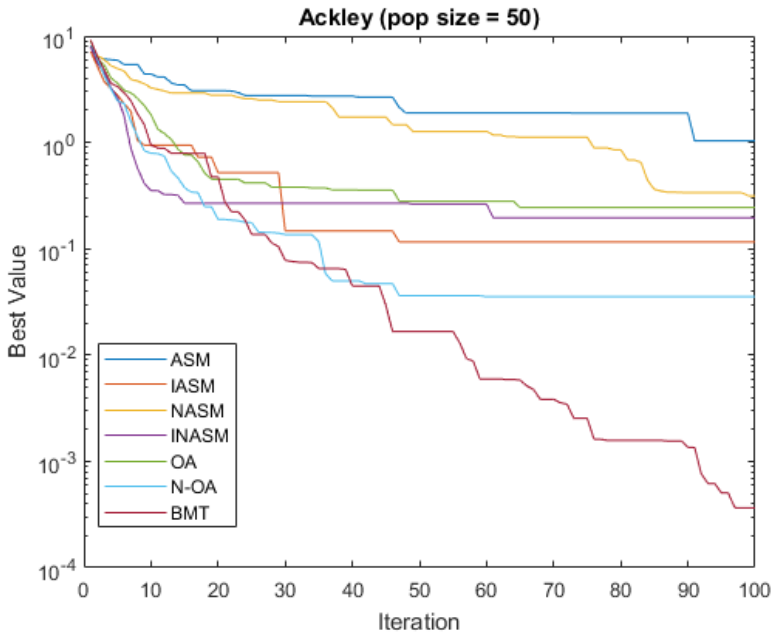

Sum of Different Powers function in 2D
2
1.5
1
0.5
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

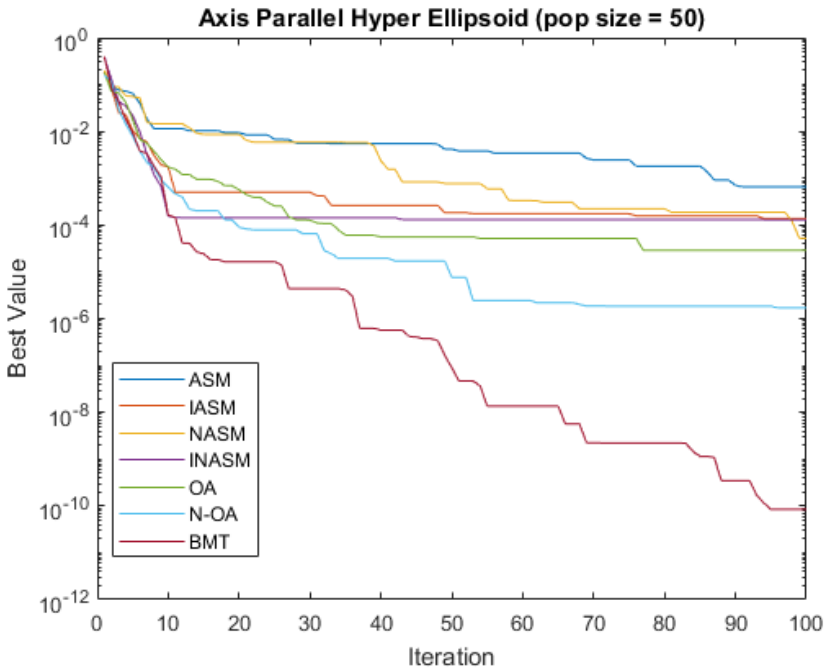
Appendix 3: The results of preliminary tests when population size is 50

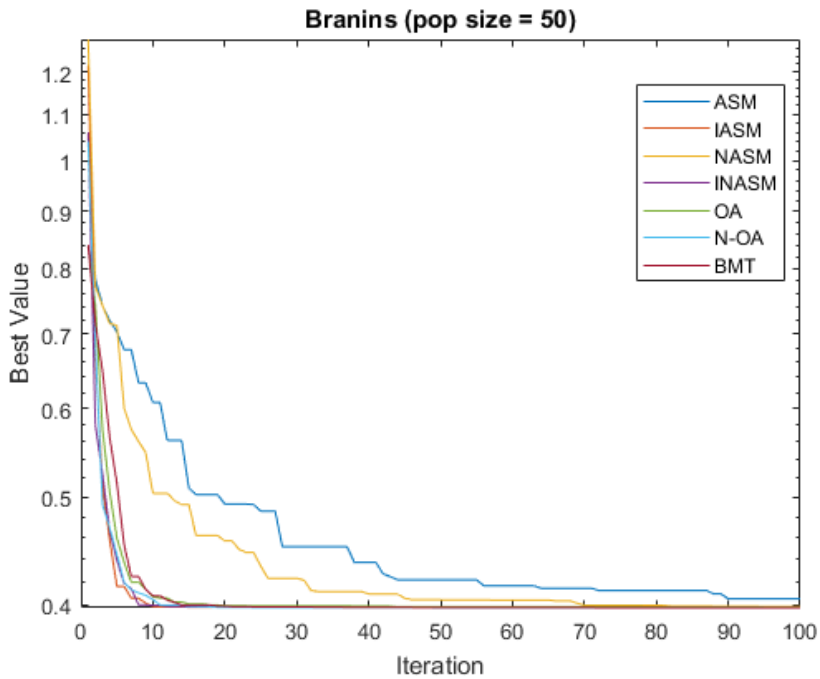
	Axis Parallel Hyper Ellipsoid		De Jong		Goldstein-Price		Rosenbrock's Valley		Sum of Different Powers	
METHOD	Median	IQR	Median	IQR	Median	IQR	Median	IQR	Median	IQR
ASM	2,84E-02	1,89E-01	1,39E-02	2,63E-02	3,49E+00	1,48E+00	3,25E-02	8,32E-02	3,34E-06	8,02E-06
NASM	4,64E-03	1,80E-02	2,59E-04	3,24E-03	3,11E+00	7,08E-01	1,31E-02	4,39E-02	7,36E-08	7,07E-07
IASM	6,02E-03	5,84E-02	4,91E-04	3,39E-03	3,11E+00	5,88E-01	1,21E-02	7,54E-02	1,08E-07	5,72E-06
INASM	1,46E-03	8,78E-03	1,06E-04	1,14E-03	3,01E+00	4,61E-02	4,06E-02	1,56E-01	1,04E-07	4,52E-07
RW	2,24E-02	5,08E-02	1,22E-03	3,87E-03	3,03E+00	3,16E-01	6,00E-02	1,36E-01	5,42E-07	2,41E-06
LR	2,82E-03	1,35E-02	8,05E-05	7,83E-04	3,04E+00	1,99E-01	4,21E-02	1,33E-01	2,47E-08	1,38E-06
Tour.	1,29E-02	7,69E-02	1,26E-03	2,75E-03	3,19E+00	4,32E-01	5,02E-02	1,60E-01	2,15E-06	1,05E-05
OA	1,26E-03	5,41E-03	1,72E-04	4,07E-04	**3,00E+00**	**2,88E-03**	1,73E-03	6,04E-03	3,67E-08	1,39E-06
N-OA	**6,45E-05**	**9,59E-04**	**7,79E-06**	**5,39E-05**	**3,00E+00**	**7,44E-04**	**9,04E-04**	**1,07E-02**	**1,21E-09**	**5,91E-09**

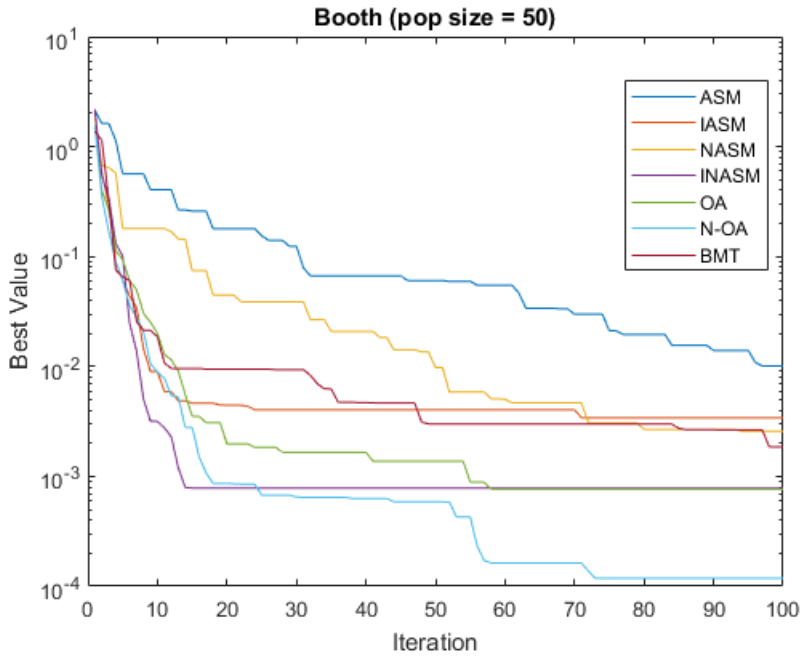
	Ackley		Langermann		Rastrigin		Schwefel		Branins	
METHOD	Median	IQR	Median	IQR	Median	IQR	Median	IQR	Median	IQR
ASM	1,03E+00	2,05E+00	-4,04E+00	3,48E-01	1,47E+00	8,05E-01	-7,16E+02	1,76E+02	4,05E-01	3,71E-02
NASM	3,14E-01	1,01E+00	-4,02E+00	4,24E-01	9,96E-01	1,60E-01	-8,25E+02	1,19E+02	3,99E-01	6,36E-03
IASM	1,15E-01	9,28E-01	-4,02E+00	2,60E-01	2,08E-01	7,25E-01	-8,37E+02	1,80E+00	**3,98E-01**	**9,61E-03**
INASM	1,95E-01	4,64E-01	-4,08E+00	7,51E-02	1,22E-01	5,41E-01	**-8,38E+02**	**8,84E-01**	**3,98E-01**	**3,15E-03**
RW	1,61E-01	3,75E-01	-3,03E+00	1,21E+00	3,12E-02	1,29E-01	-6,84E+02	1,61E+02	**3,98E-01**	**1,22E-02**
LR	8,93E-02	3,22E-01	-4,04E+00	1,66E-01	1,44E-01	9,81E-01	**-8,38E+02**	**1,28E+00**	3,99E-01	6,26E-03
Tour.	2,31E-01	1,21E+00	-4,03E+00	9,30E-02	1,39E-01	6,72E-01	**-8,38E+02**	**1,40E+00**	4,03E-01	2,61E-02
OA	2,44E-01	5,15E-01	-4,10E+00	4,78E-02	**0,00E+00**	**2,32E-03**	**-8,38E+02**	**7,36E-01**	**3,98E-01**	**2,33E-03**
N-OA	**3,54E-02**	**1,20E-01**	**-4,13E+00**	**5,98E-02**	**0,00E+00**	**0,00E+00**	**-8,38E+02**	**2,11E-01**	**3,98E-01**	**5,63E-04**

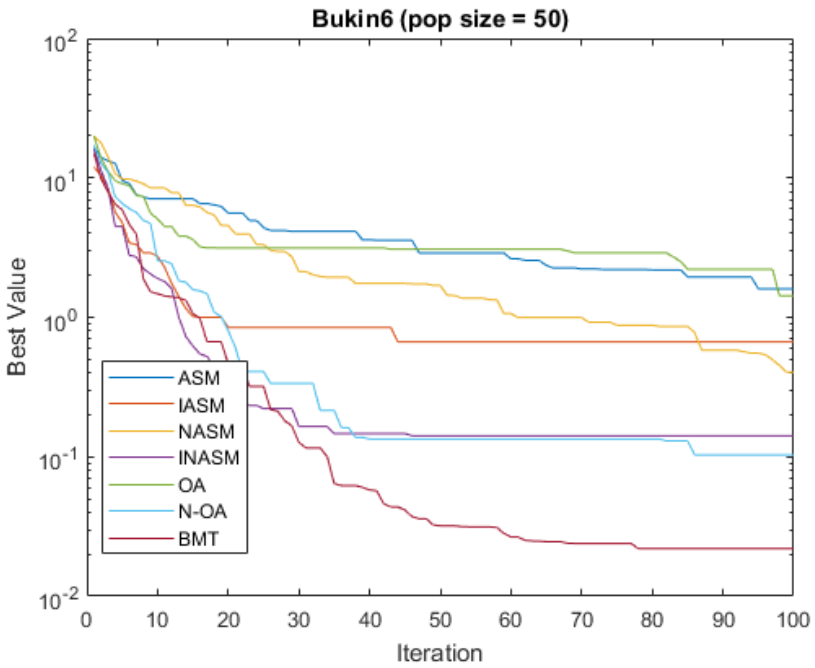
Appendix 4: The visual presentation of obtained results

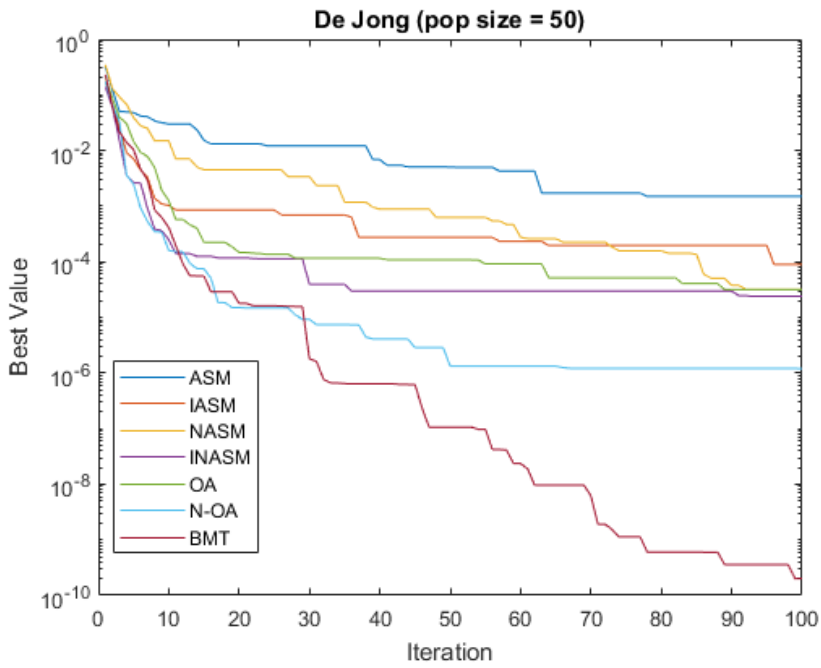


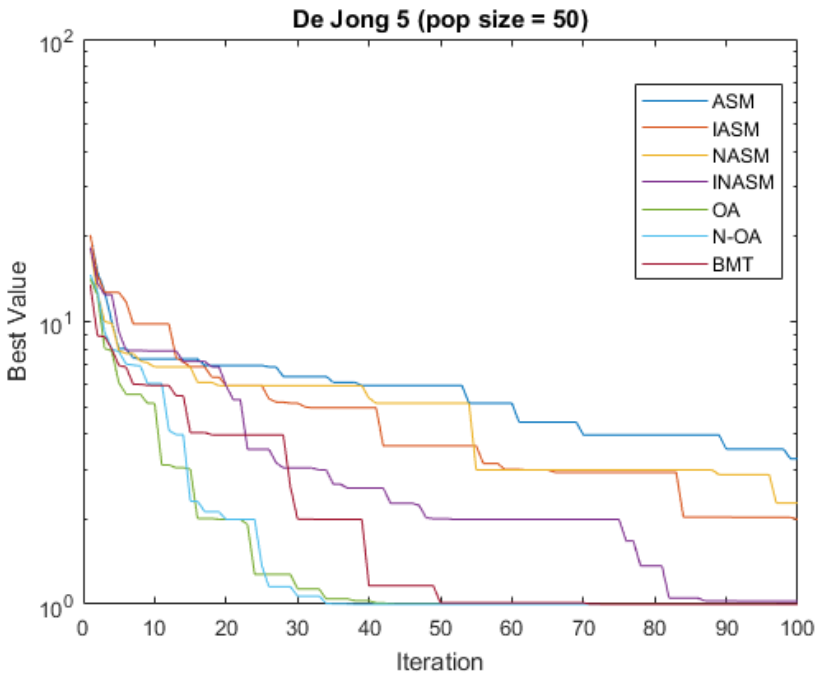


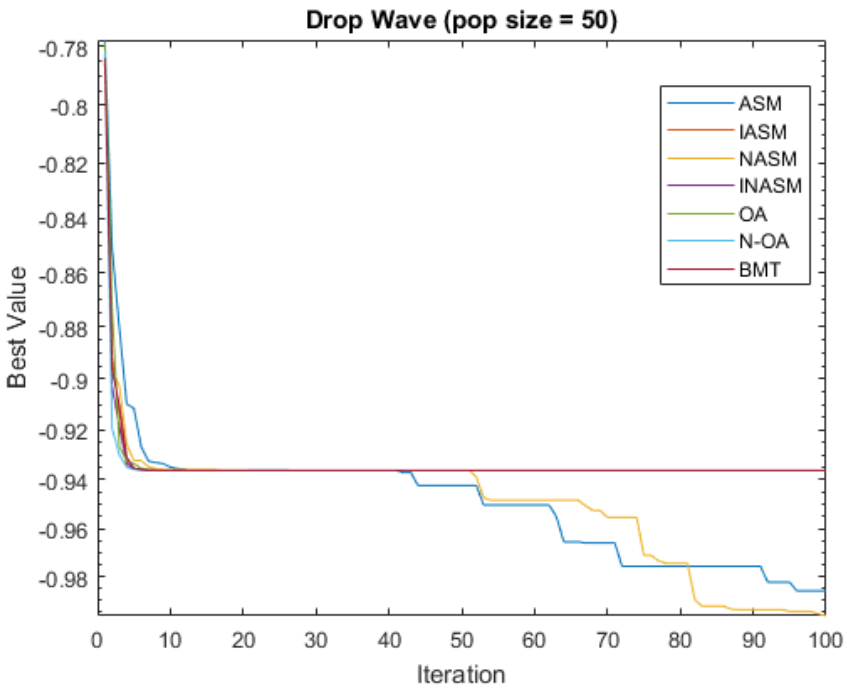

Branins (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

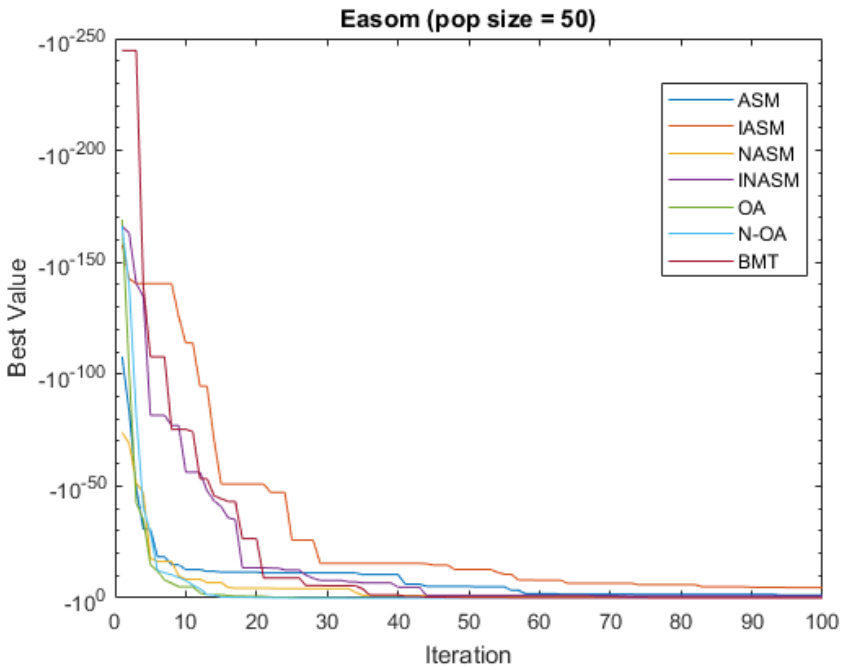

Booth (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

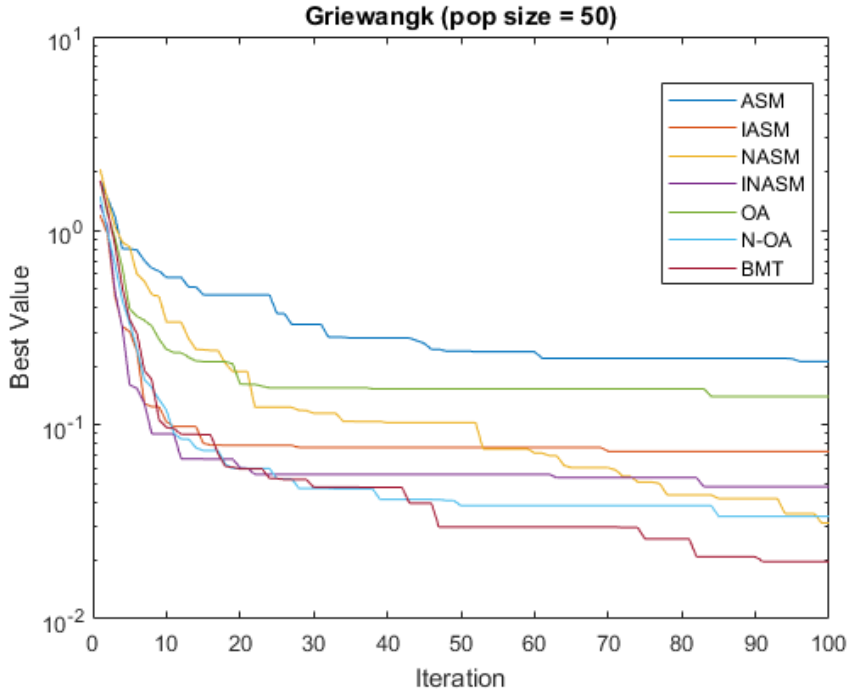

Bukin6 (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

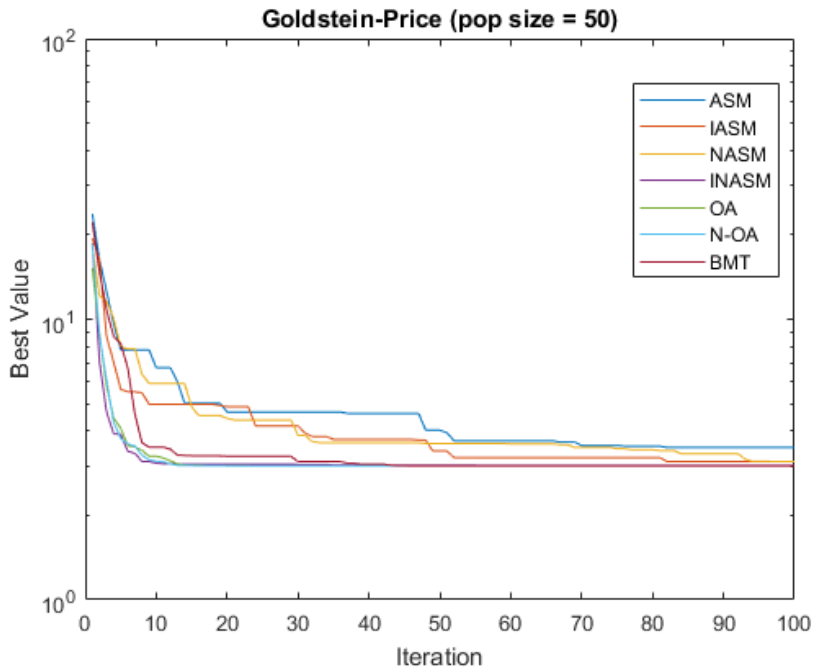

De Jong (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

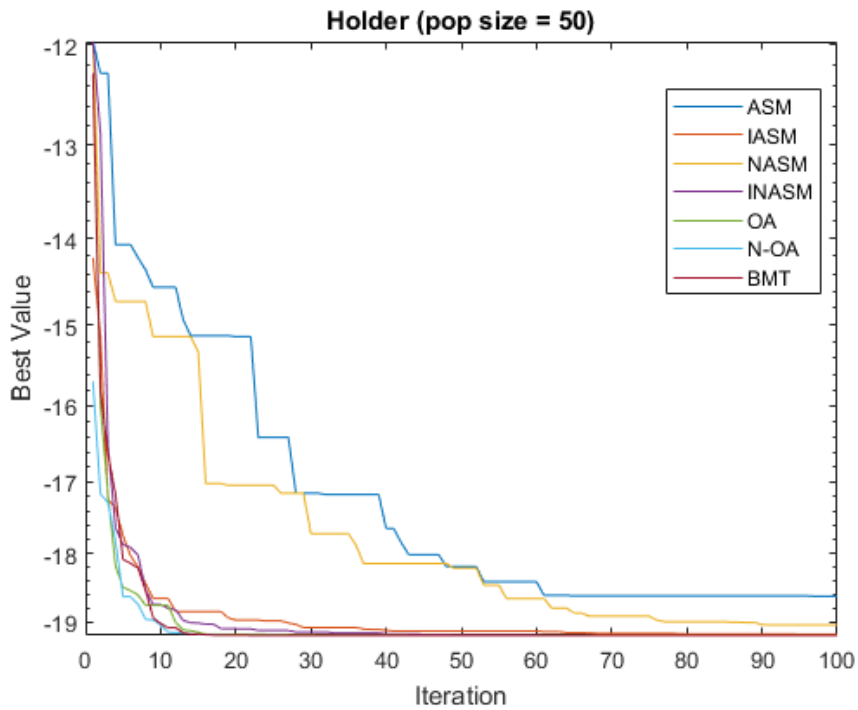

De Jong 5 (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

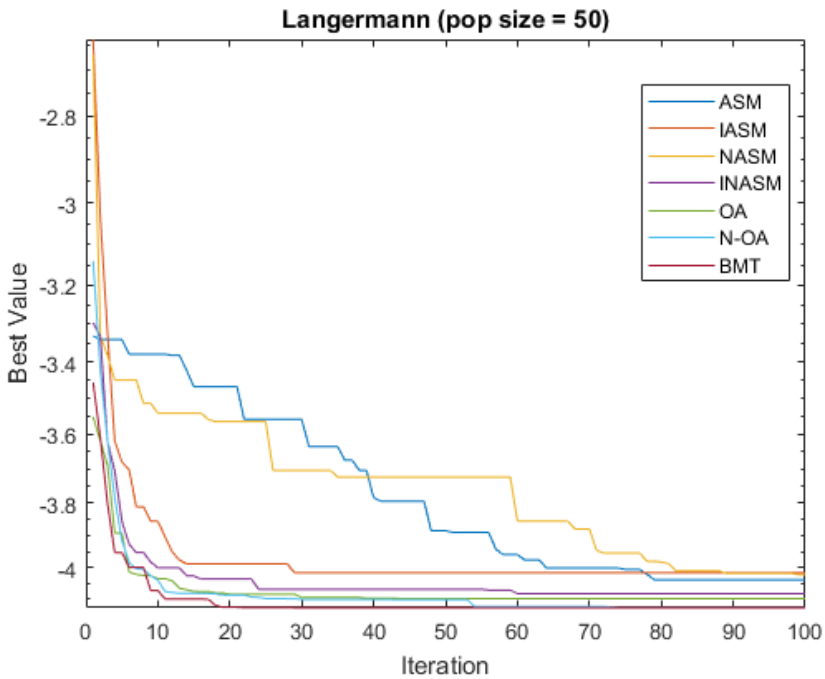

Drop Wave (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

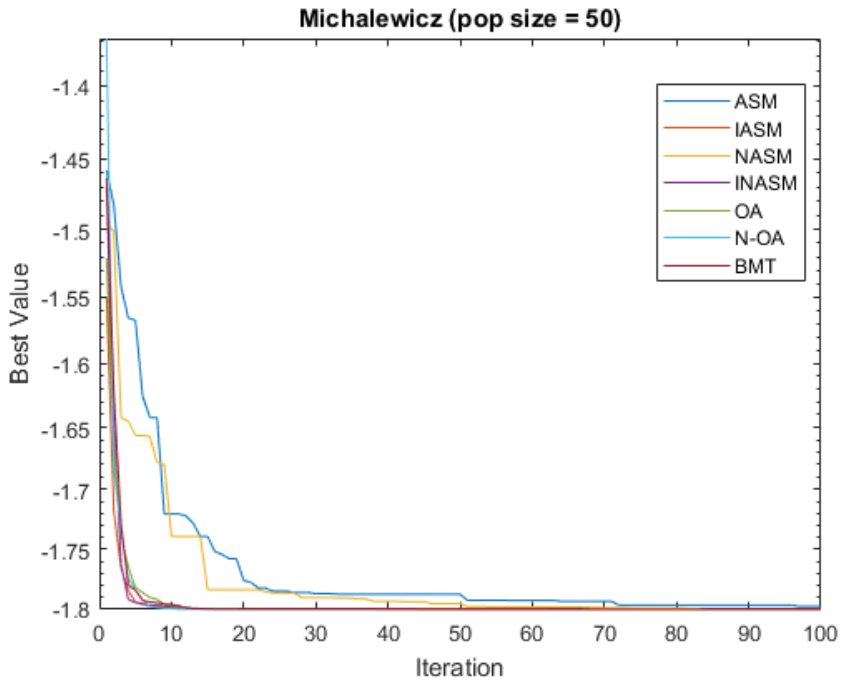

Easom (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

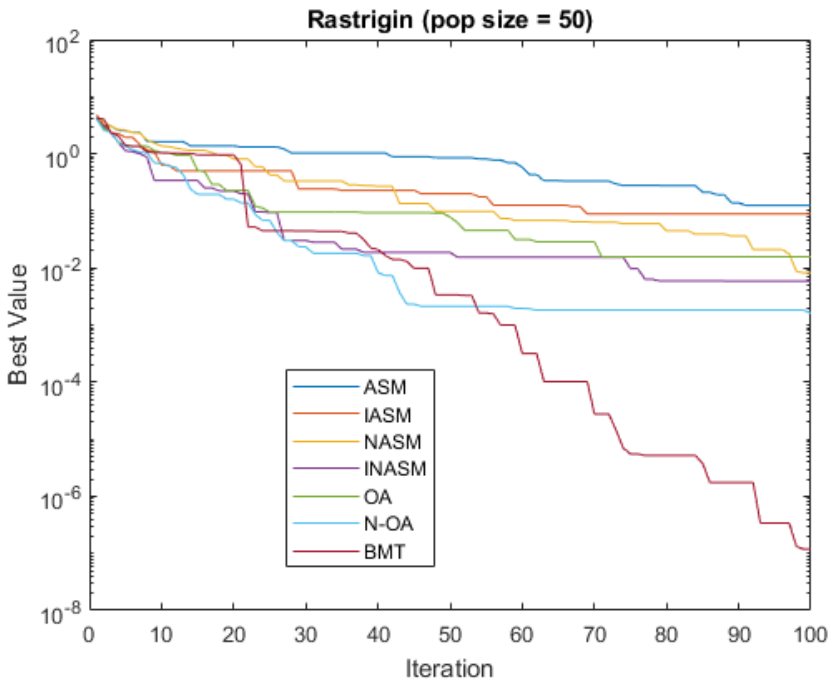

Griewangk (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

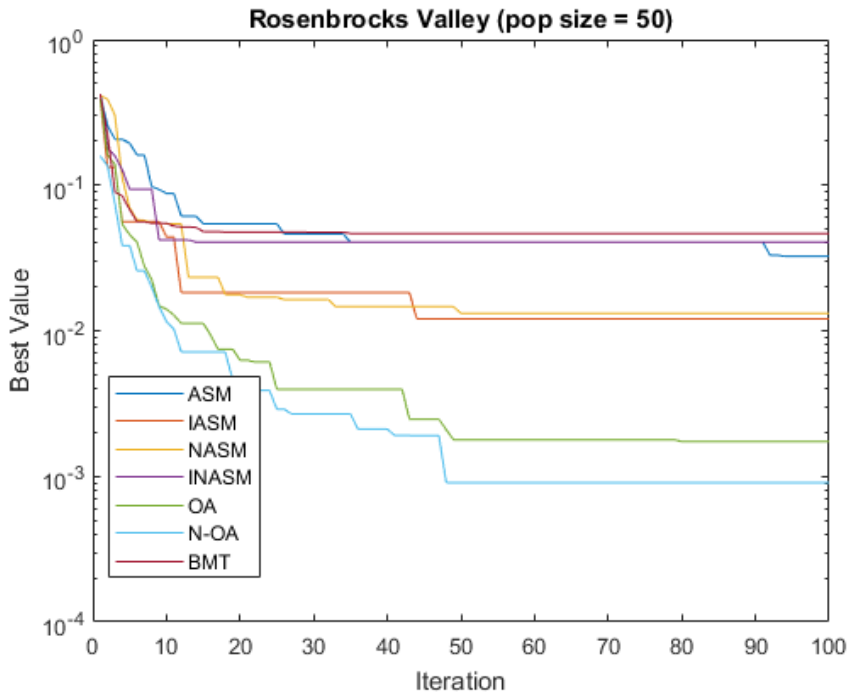

Goldstein-Price (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

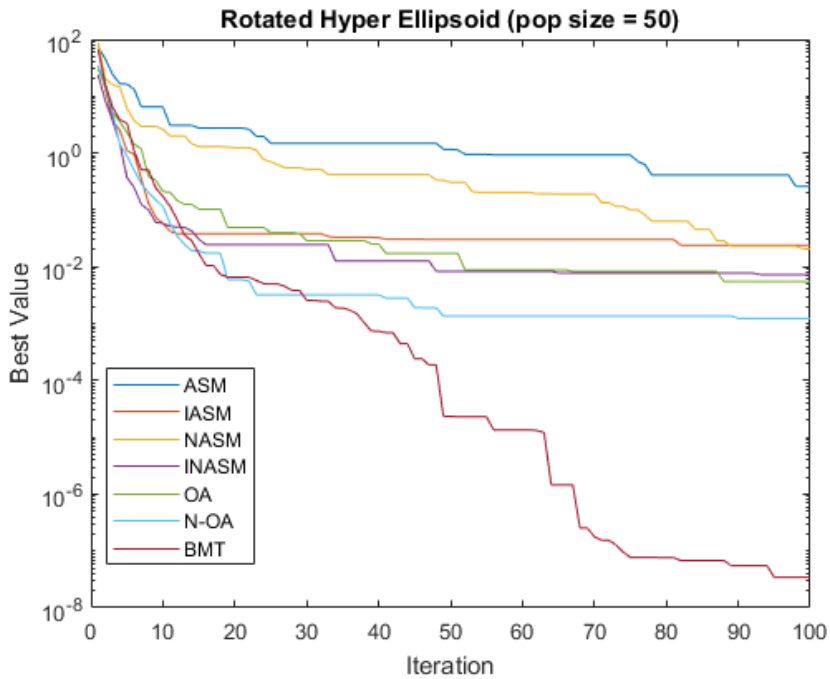

Holder (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

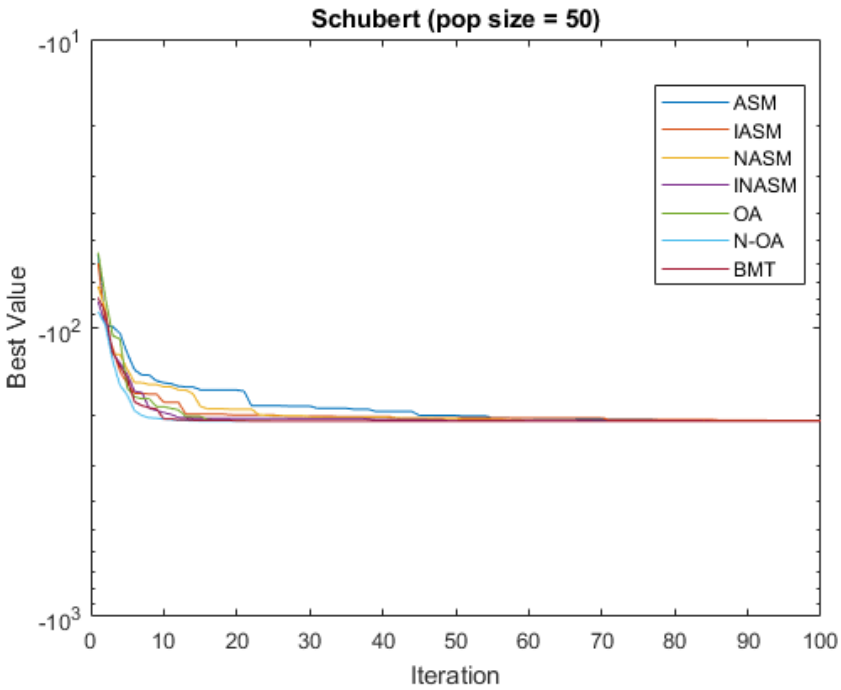

Langermann (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT
-2.8
-3
-3.2
-3.4
-3.6
-3.8
-4
0
10
20
30
40
50
60
70
80
90
100

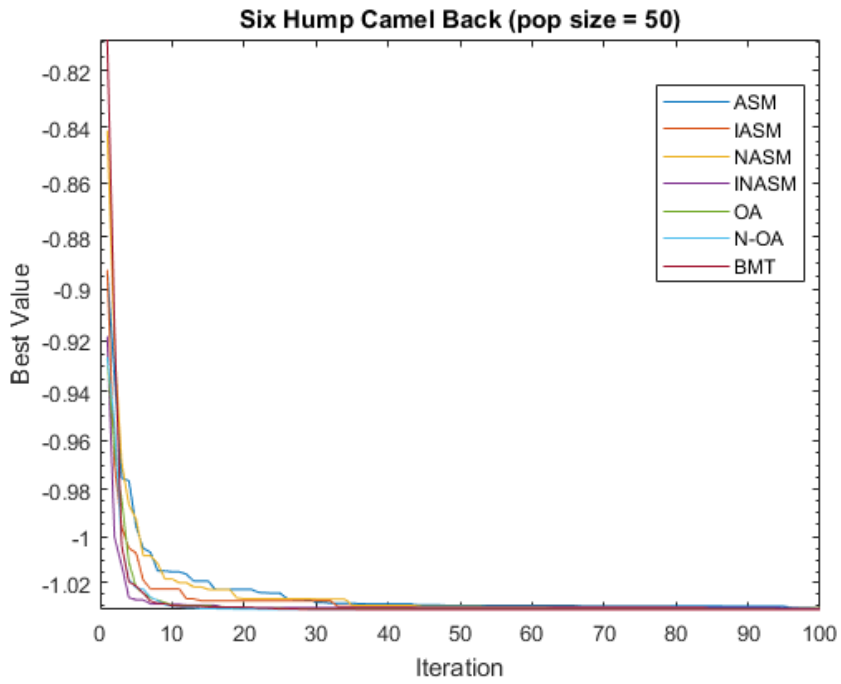

Michalewicz (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT
-1.4
-1.45
-1.5
-1.55
-1.6
-1.65
-1.7
-1.75
-1.8
0
10
20
30
40
50
60
70
80
90
100

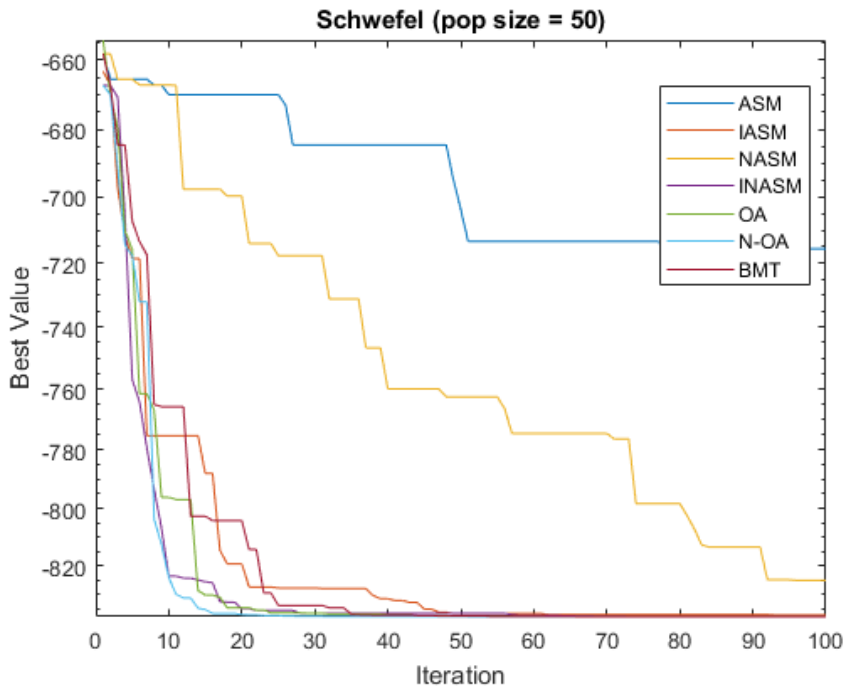

Rastrigin (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

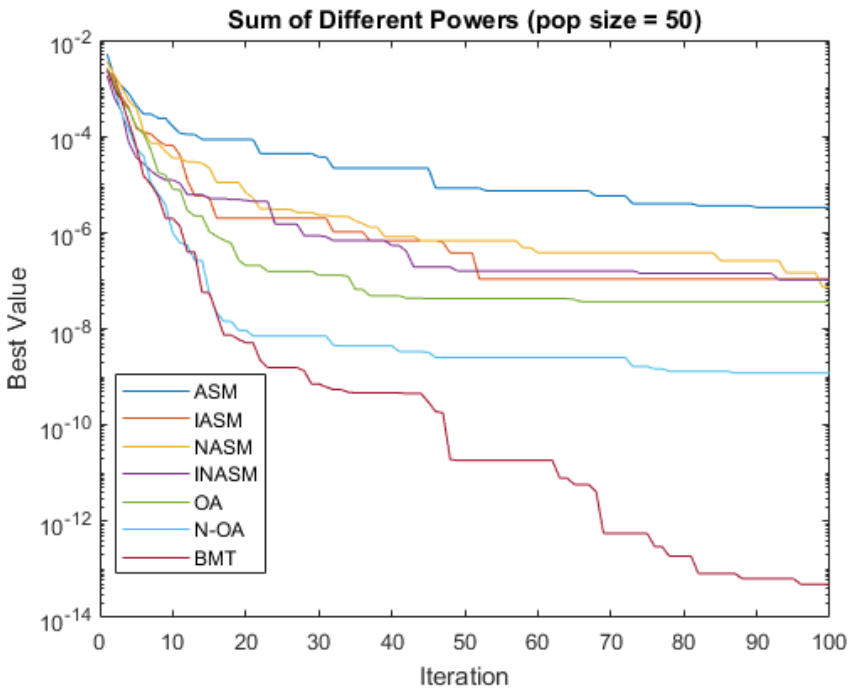

Rosenbrocks Valley (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

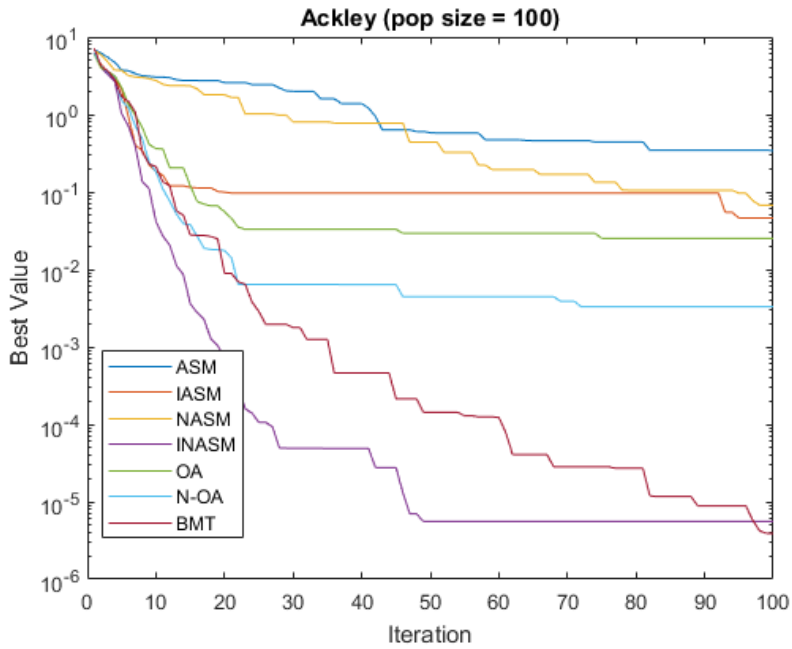

Rotated Hyper Ellipsoid (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

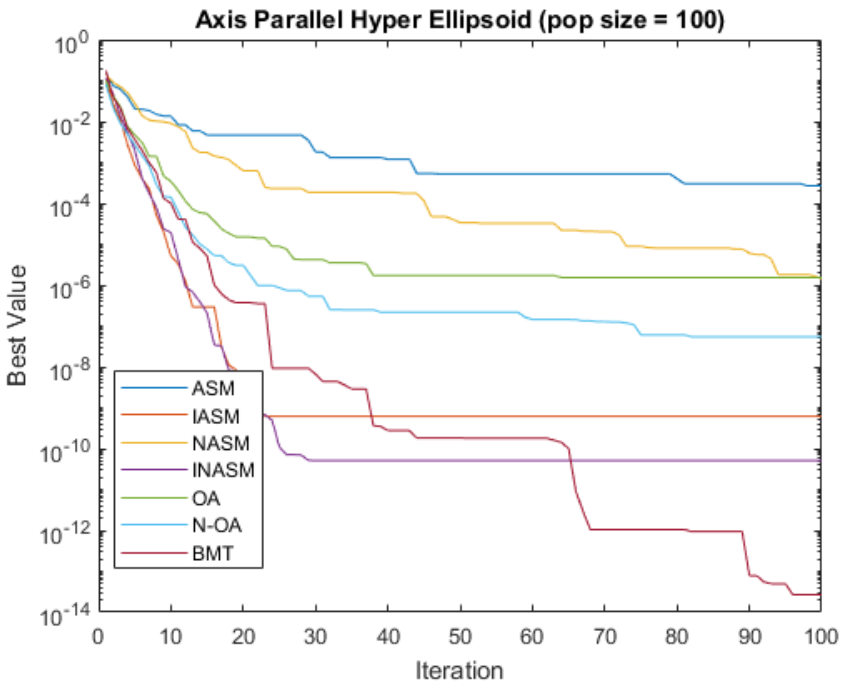

Schubert (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

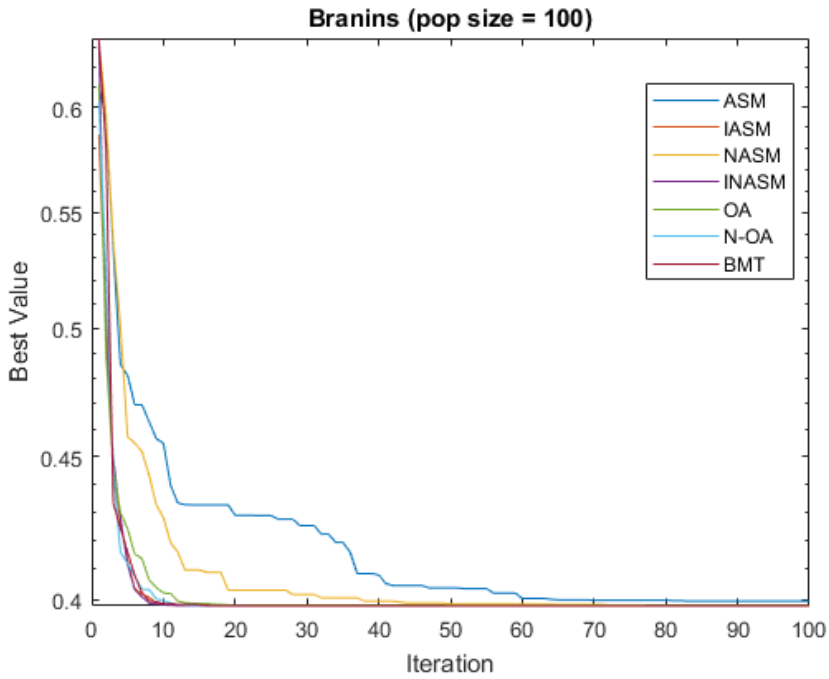

Six Hump Camel Back (pop size = 50)
Best Value
Iteration
-0.82
-0.84
-0.86
-0.88
-0.9
-0.92
-0.94
-0.96
-0.98
-1
-1.02
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

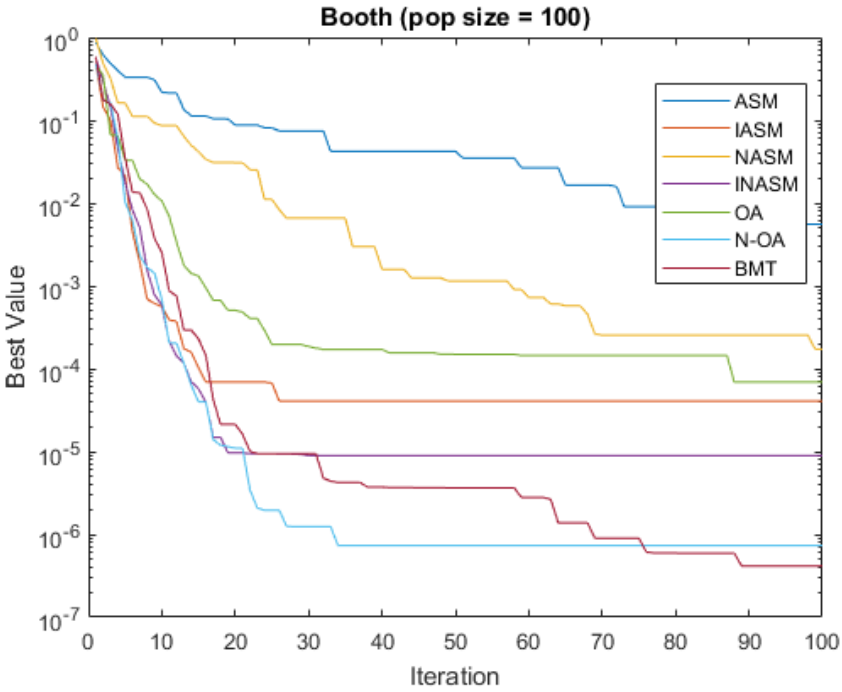

Schwefel (pop size = 50)
Best Value
Iteration
-660
-680
-700
-720
-740
-760
-780
-800
-820
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

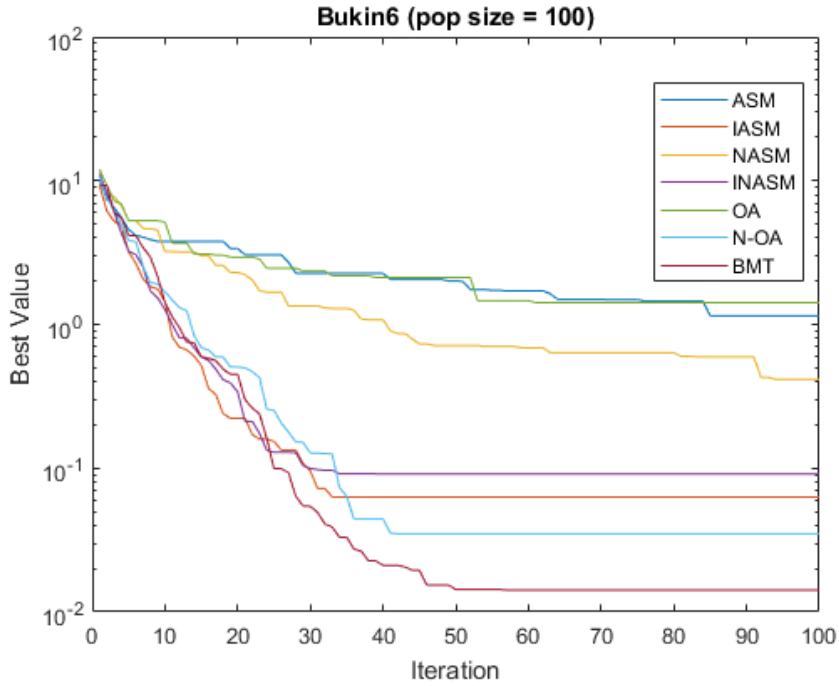

Sum of Different Powers (pop size = 50)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

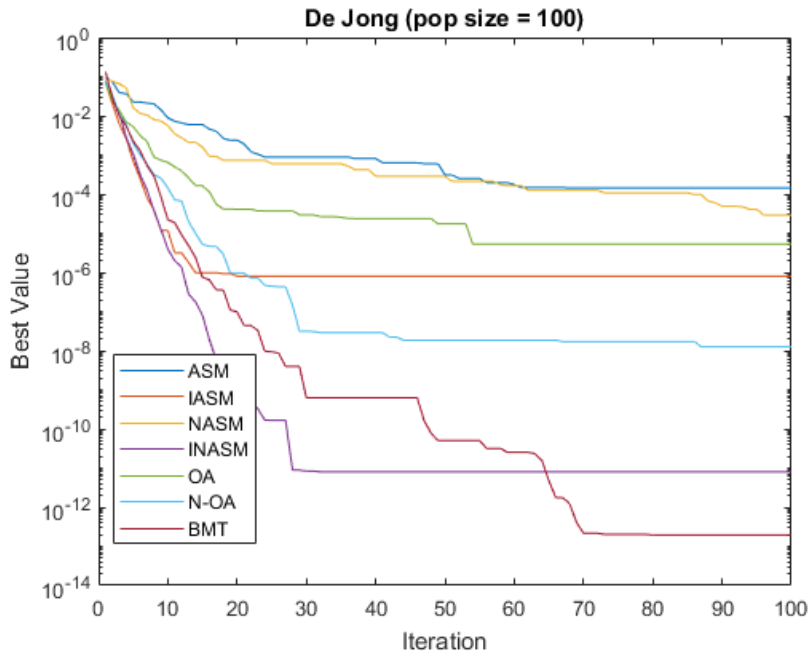

Ackley (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

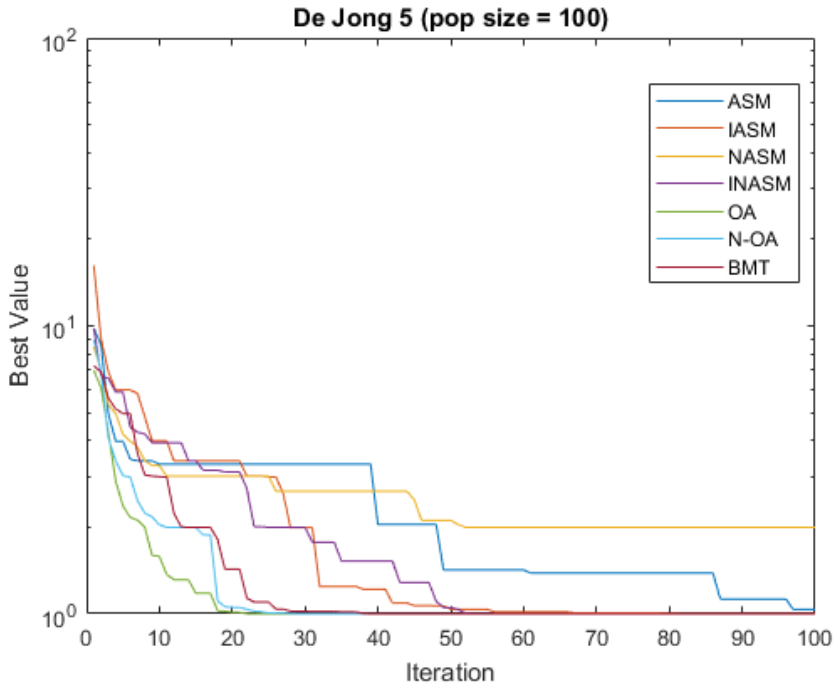

Axis Parallel Hyper Ellipsoid (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

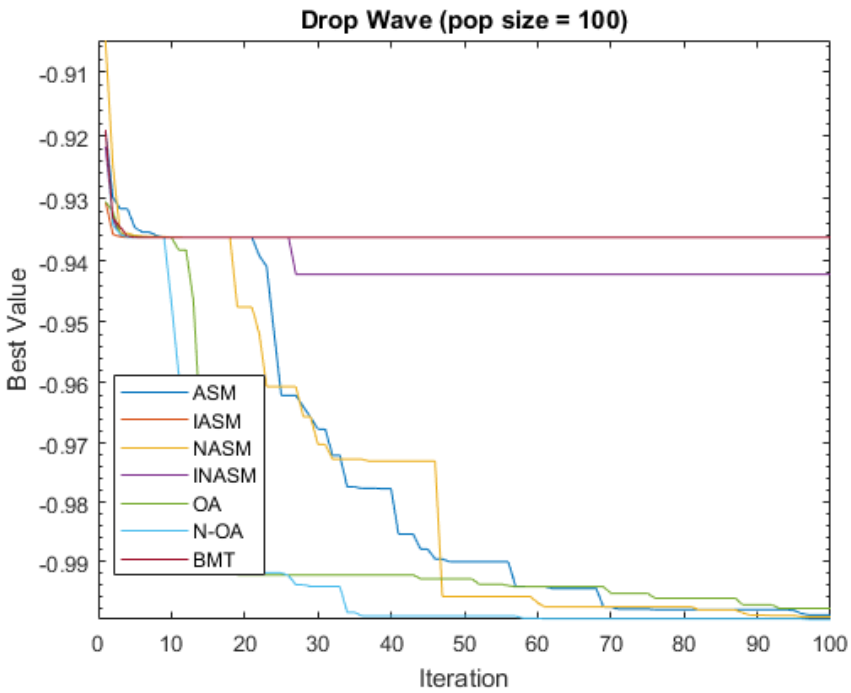

Branins (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

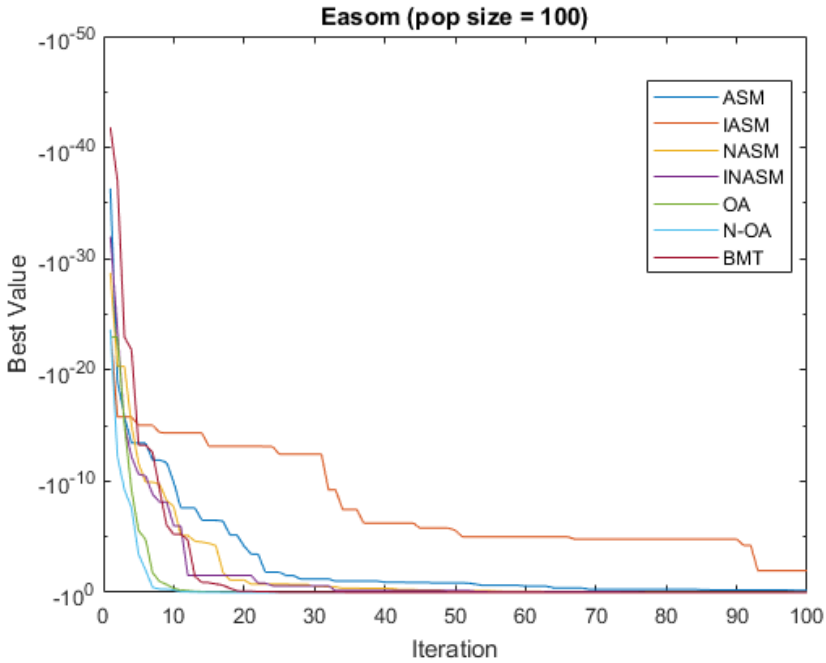

Booth (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

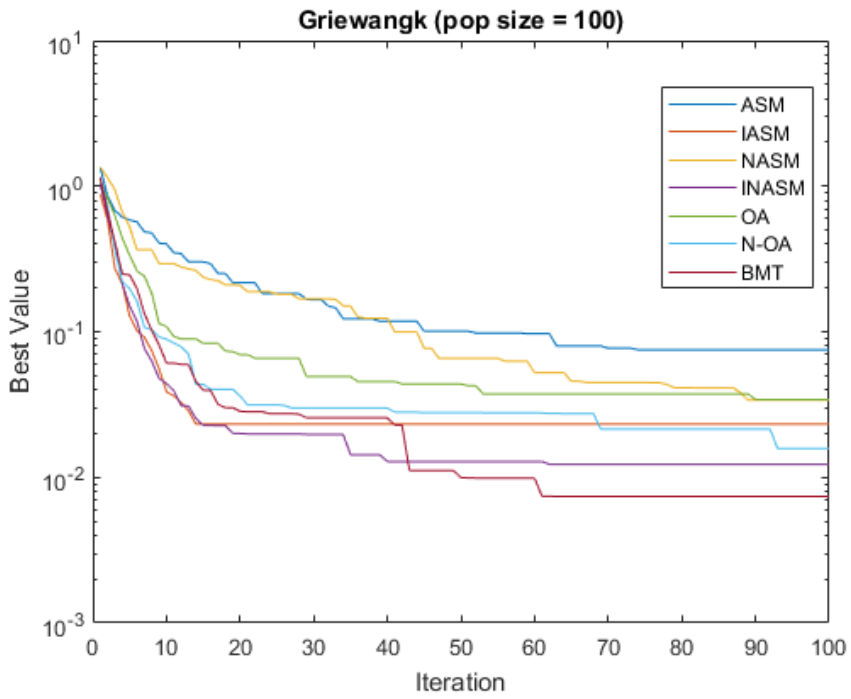

Bukin6 (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

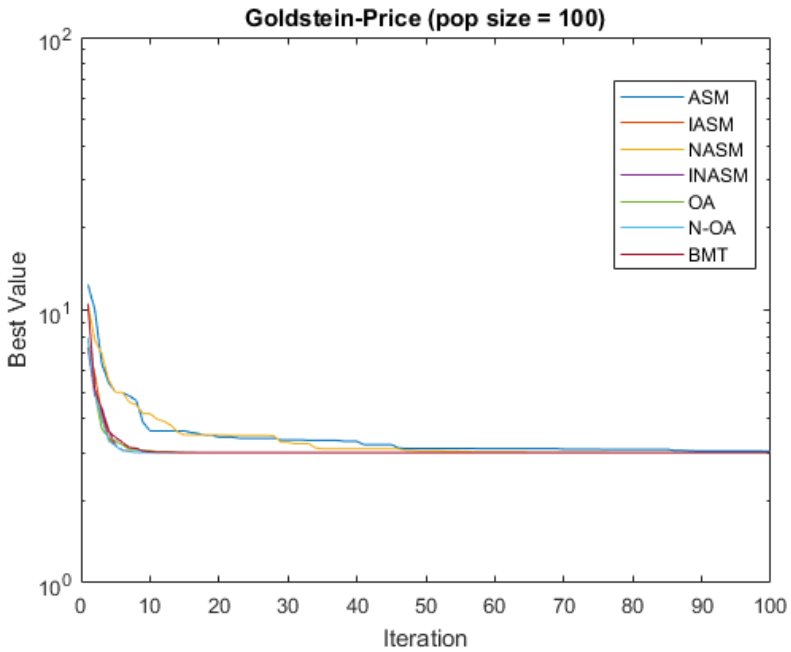

De Jong (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

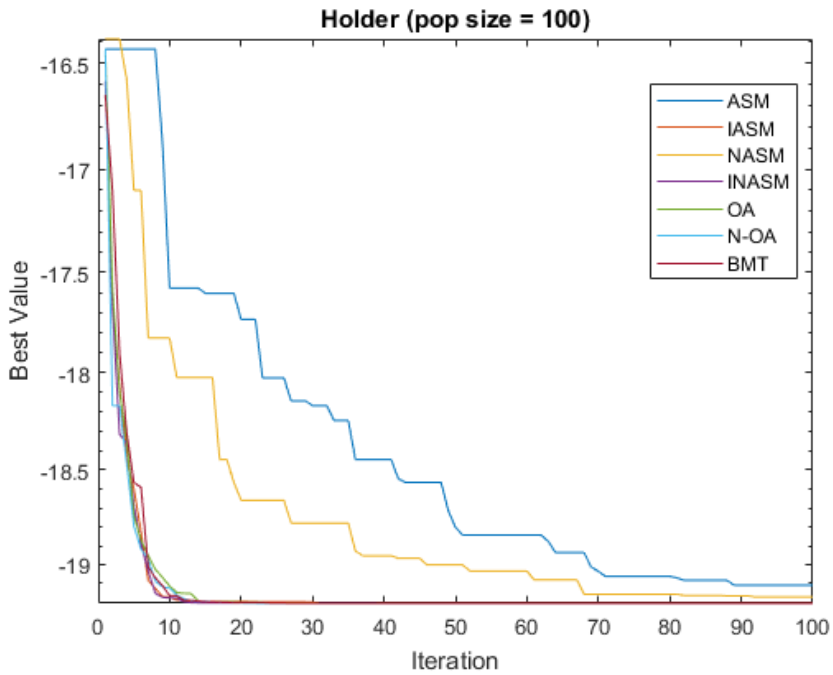

De Jong 5 (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

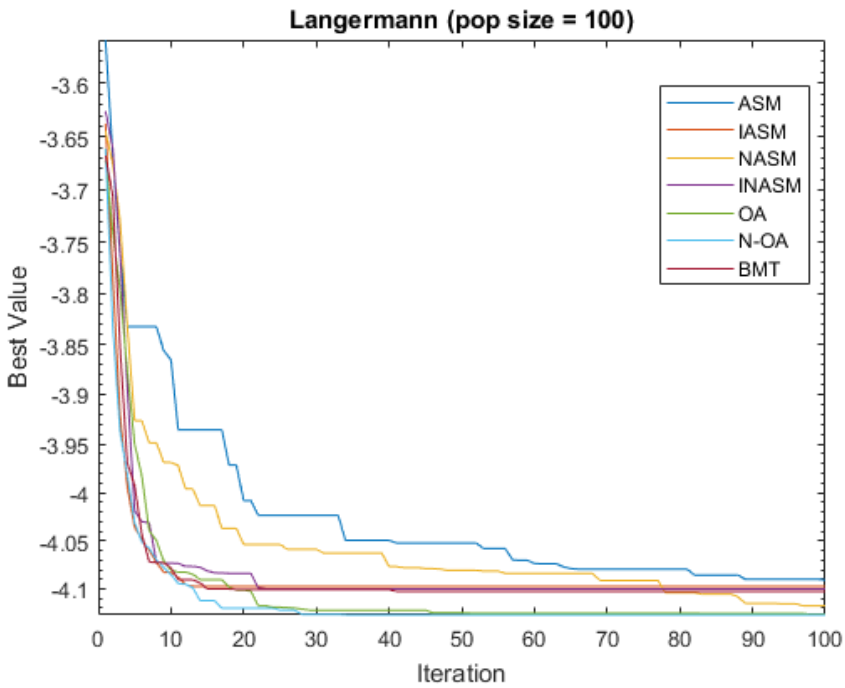

Drop Wave (pop size = 100)
Best Value
Iteration
-0.91
-0.92
-0.93
-0.94
-0.95
-0.96
-0.97
-0.98
-0.99
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

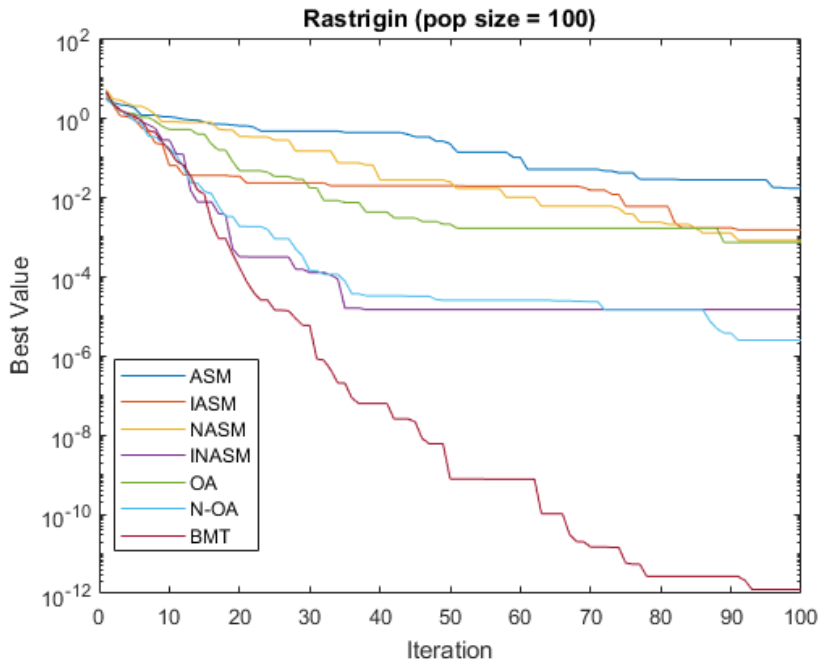

Easom (pop size = 100)
Best Value
Iteration
-10^-50
-10^-40
-10^-30
-10^-20
-10^-10
-10^0
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

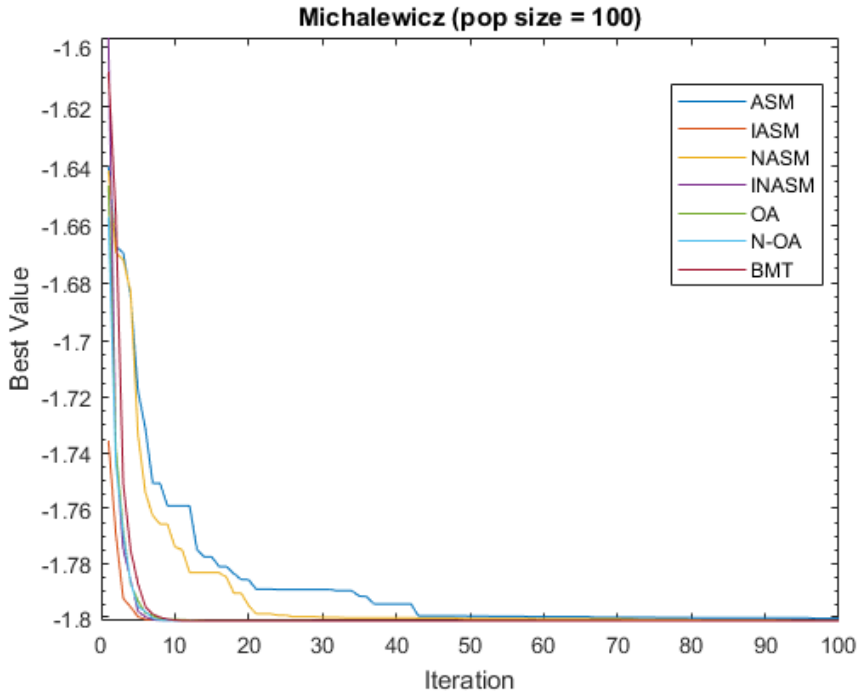

Griewangk (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

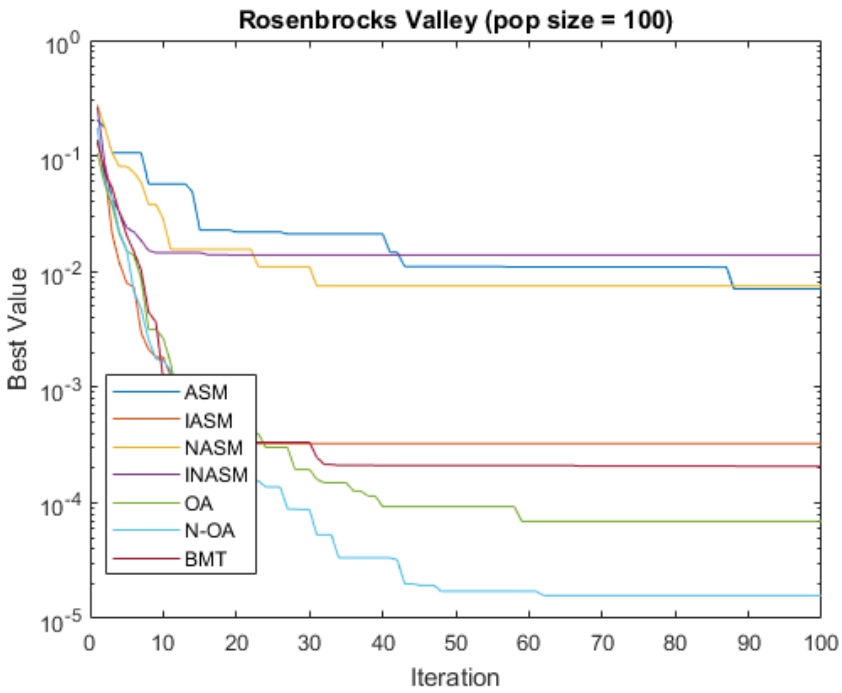

Goldstein-Price (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

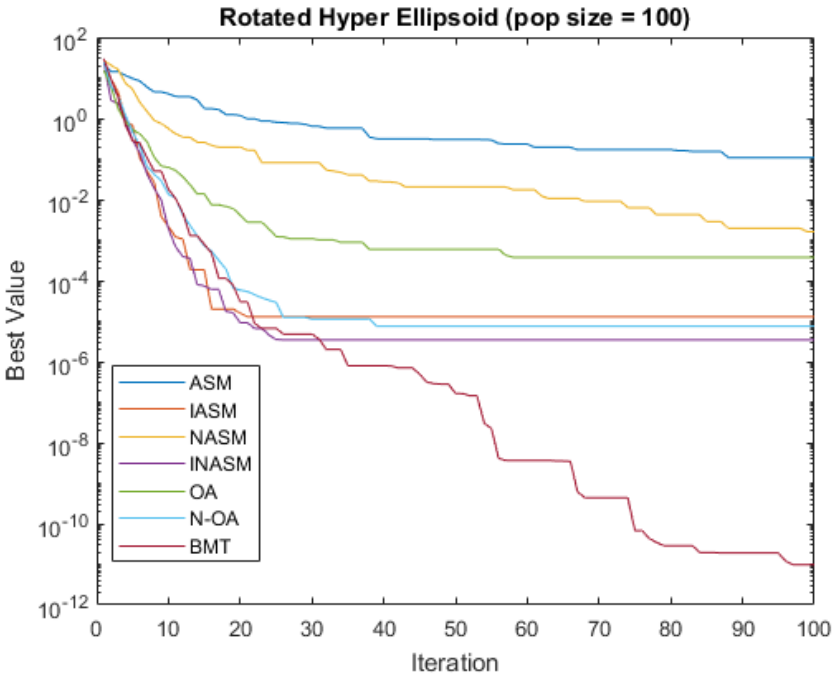

Holder (pop size = 100)
Best Value
Iteration
-16.5
-17
-17.5
-18
-18.5
-19
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

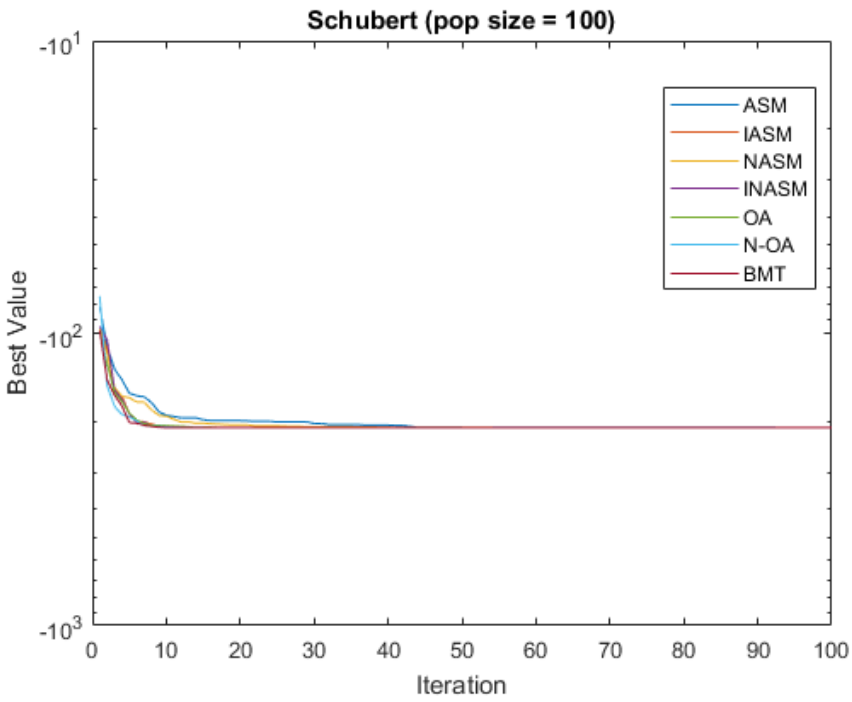

Langermann (pop size = 100)
Best Value
Iteration
-3.6
-3.65
-3.7
-3.75
-3.8
-3.85
-3.9
-3.95
-4
-4.05
-4.1
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

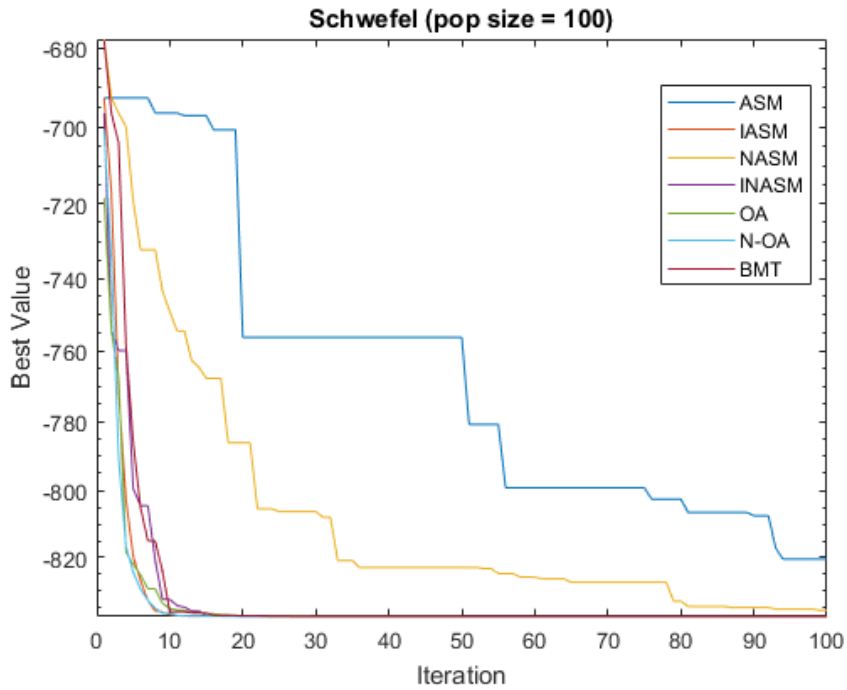

Rastrigin (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

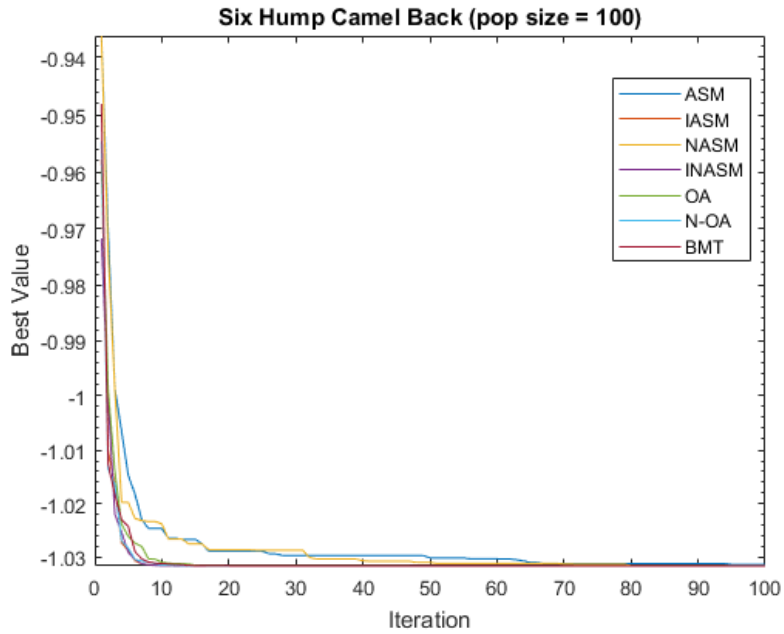

Michalewicz (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

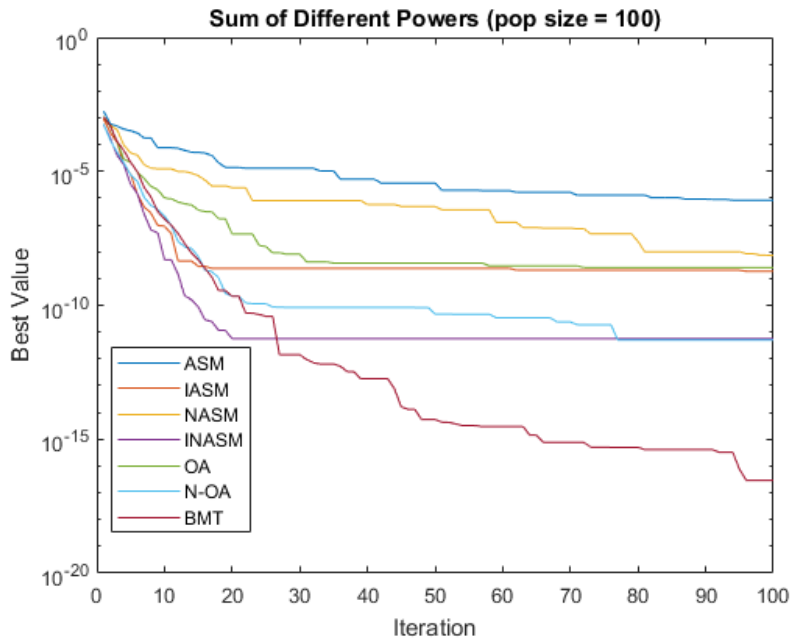

Rosenbrocks Valley (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

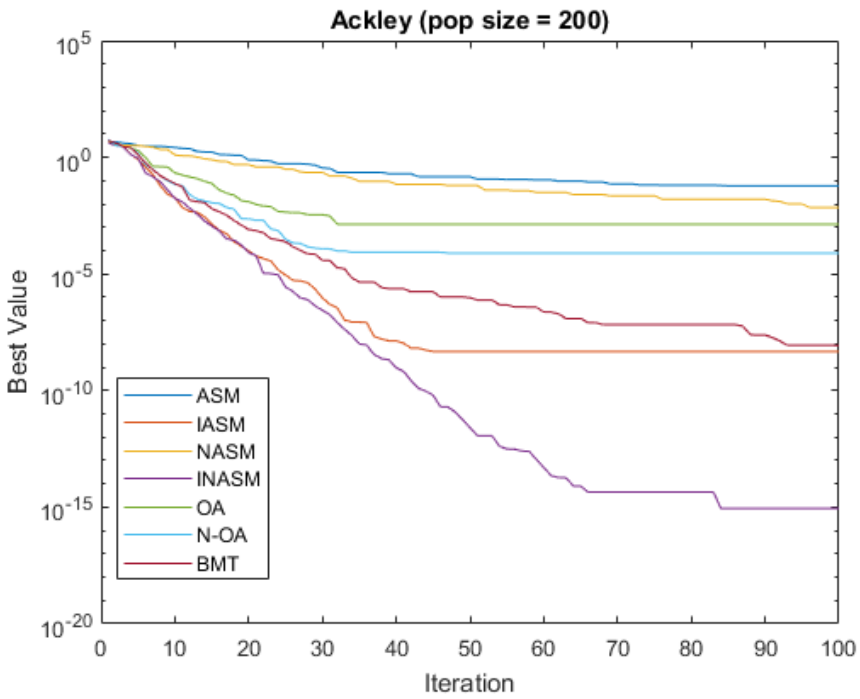

Rotated Hyper Ellipsoid (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

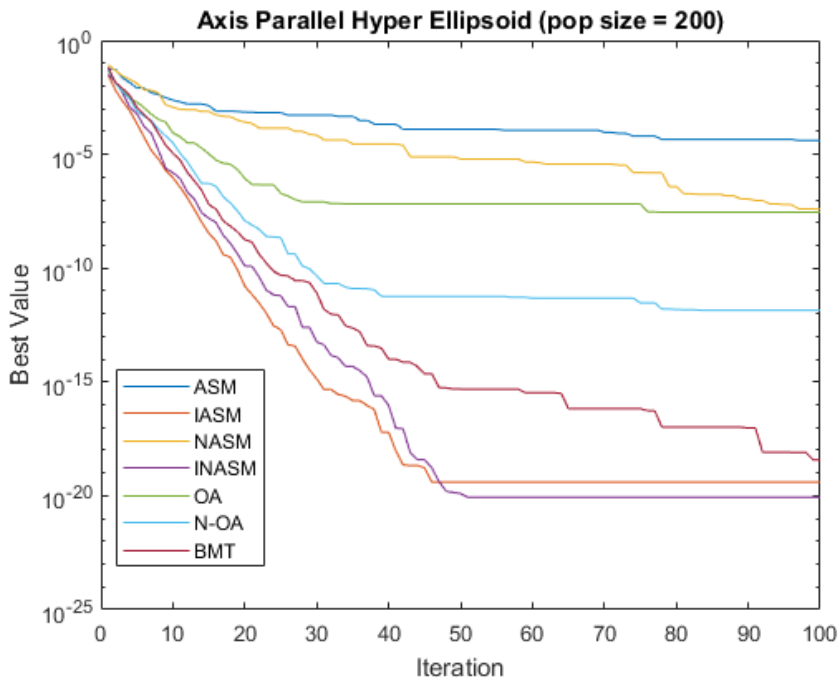

Schubert (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

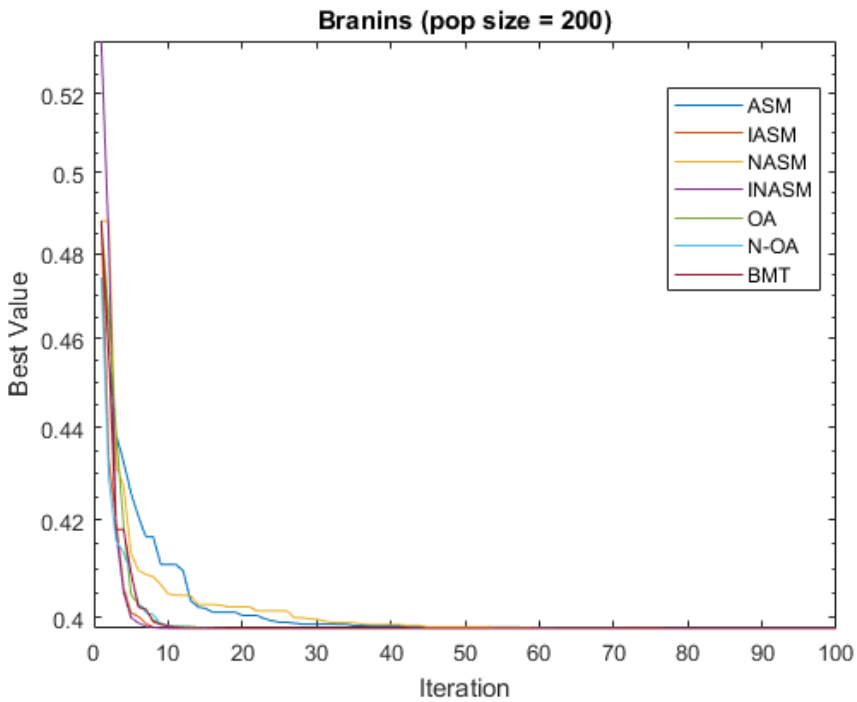

Schwefel (pop size = 100)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

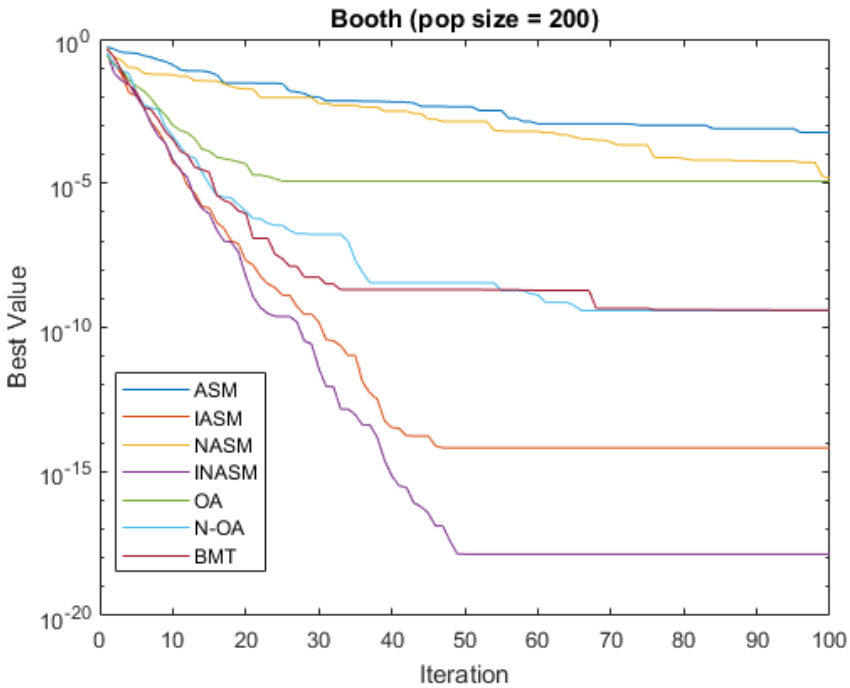

Six Hump Camel Back (pop size = 100)
Best Value
Iteration
-0.94
-0.95
-0.96
-0.97
-0.98
-0.99
-1
-1.01
-1.02
-1.03
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

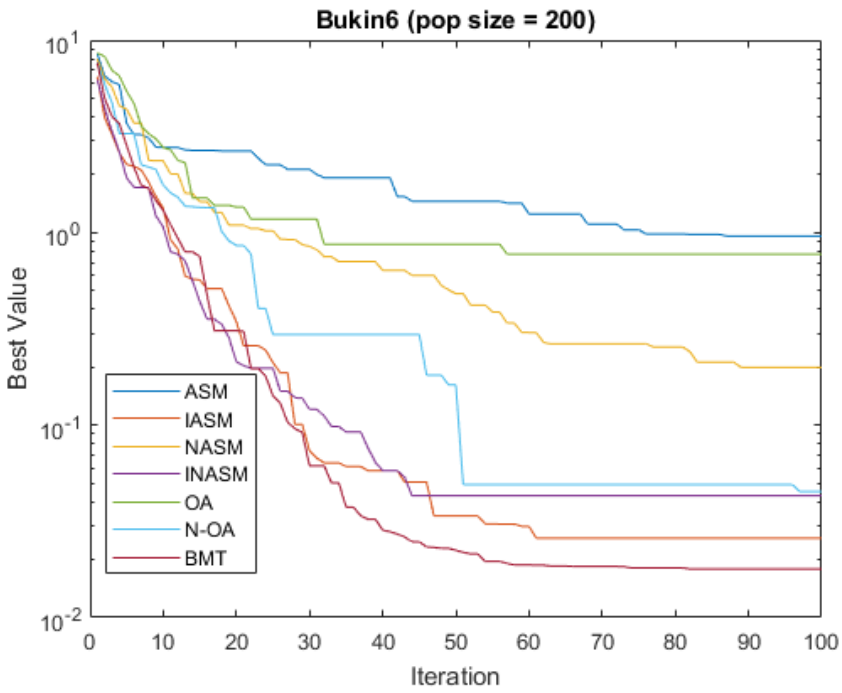

Sum of Different Powers (pop size = 100)
Best Value
Iteration
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-20}
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

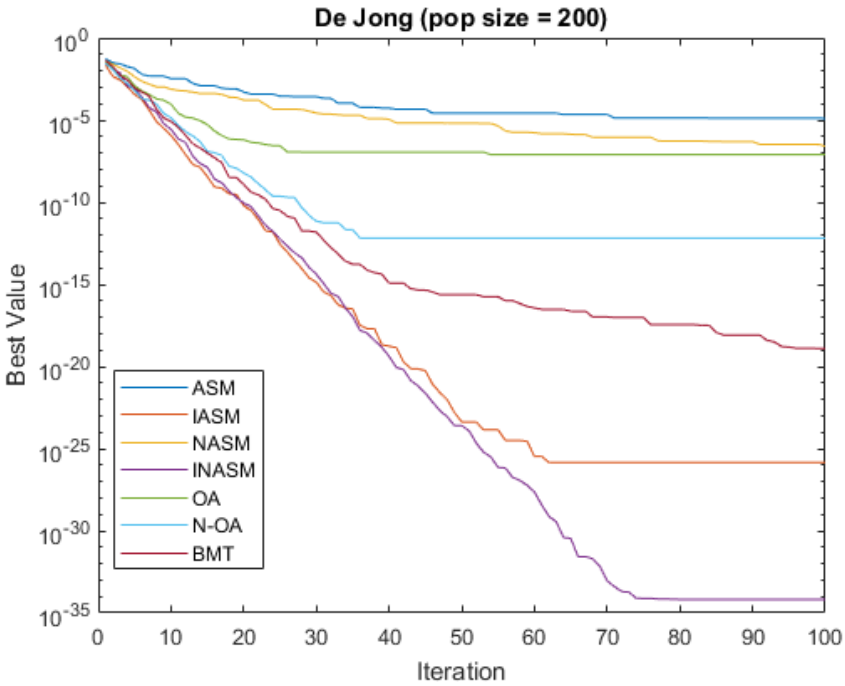

Ackley (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

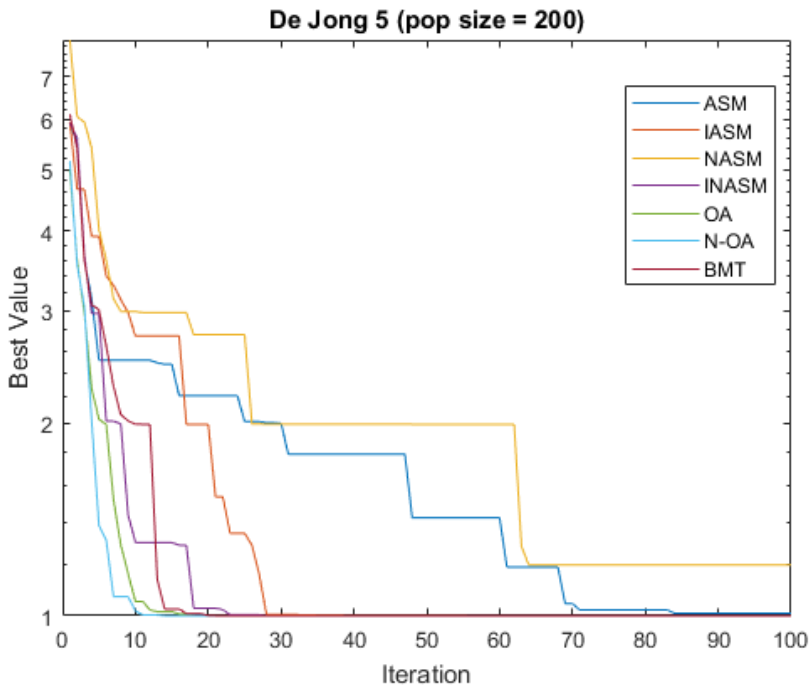

Axis Parallel Hyper Ellipsoid (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

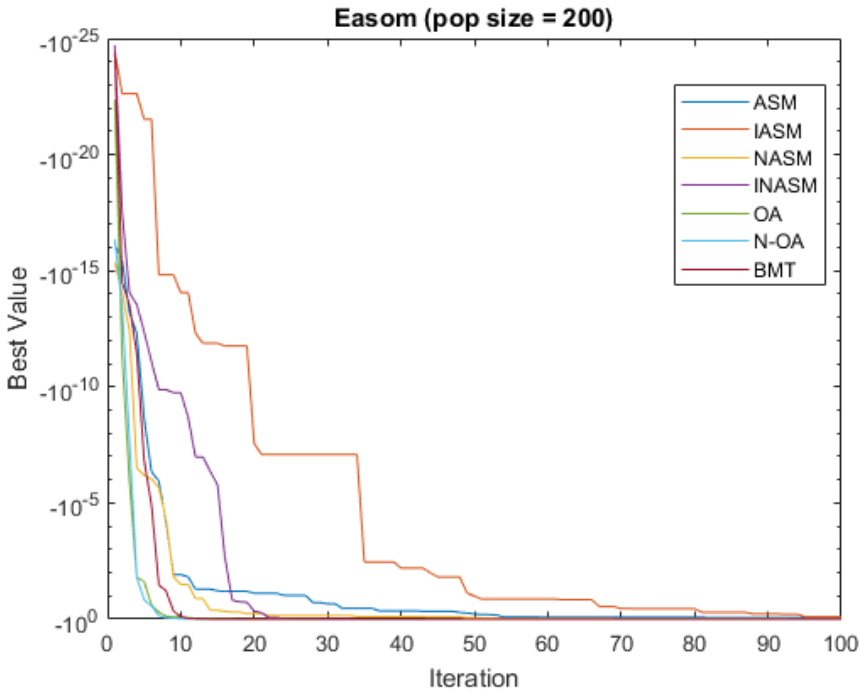

Branins (pop size = 200)
Best Value
Iteration
0.52
0.5
0.48
0.46
0.44
0.42
0.4
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

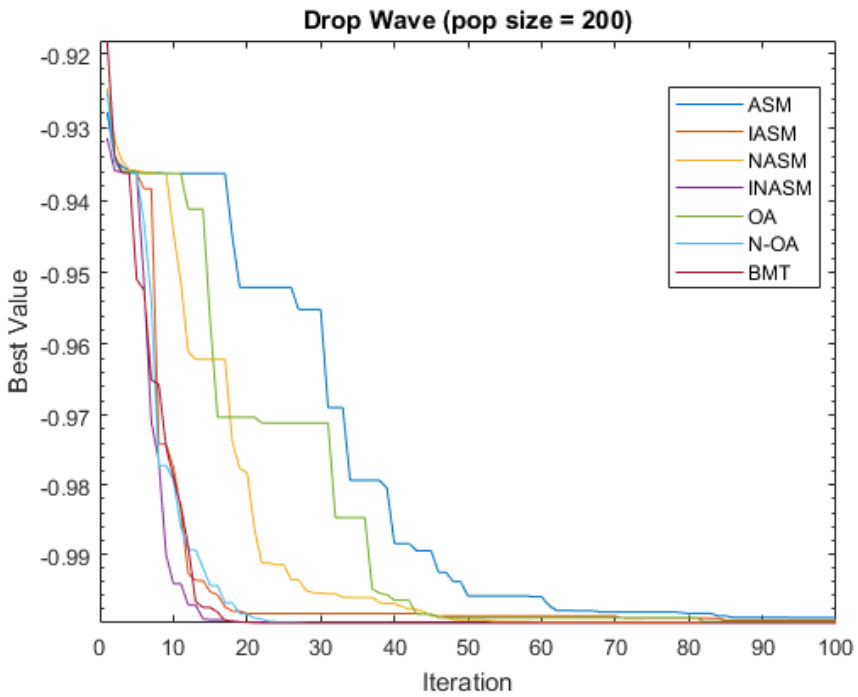

Booth (pop size = 200)
Best Value
Iteration
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-20}
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

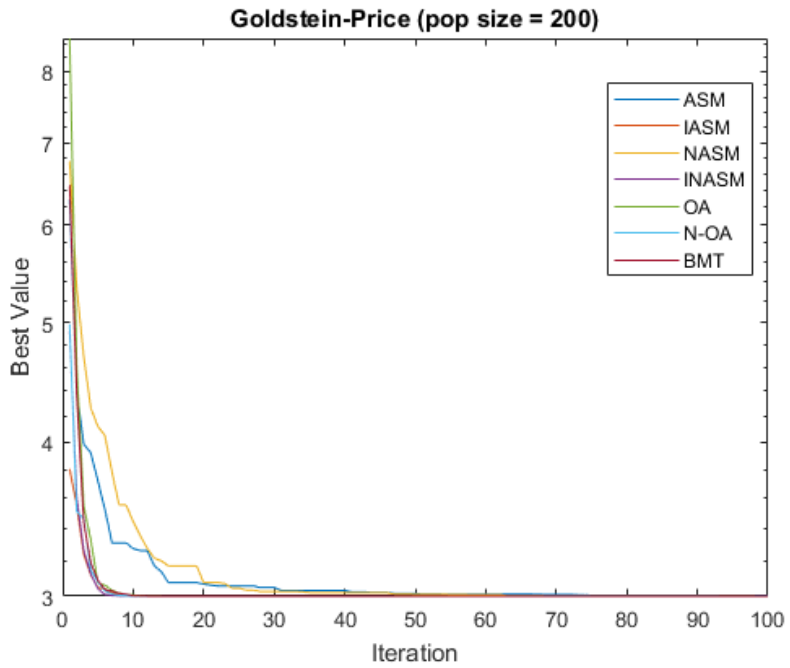

Bukin6 (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

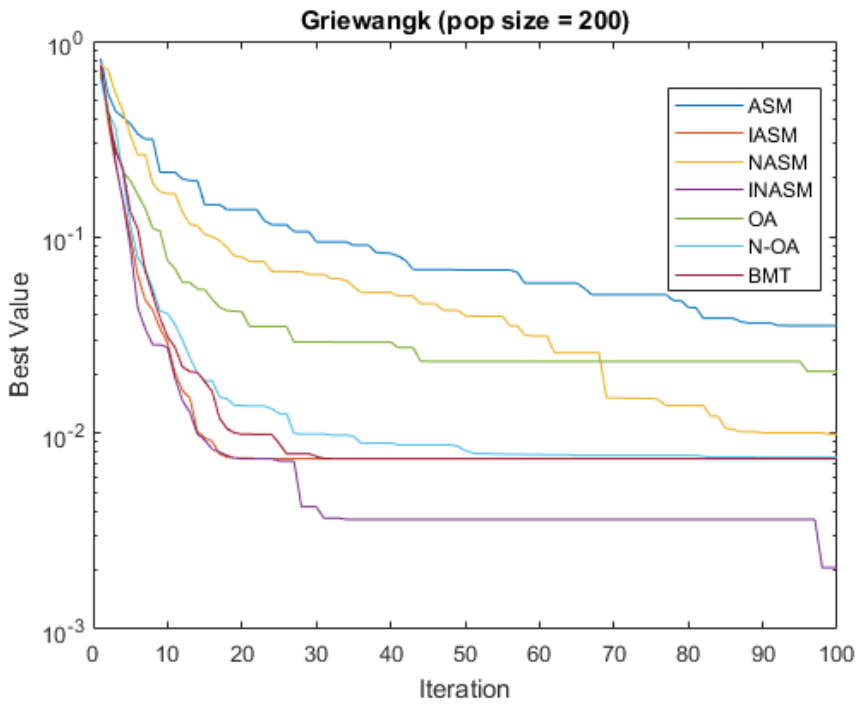

De Jong (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

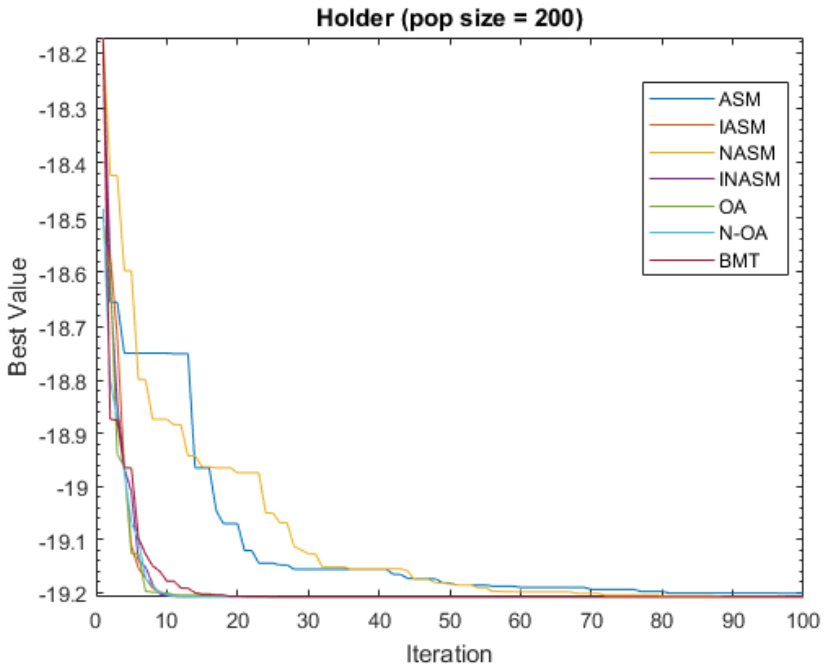

De Jong 5 (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

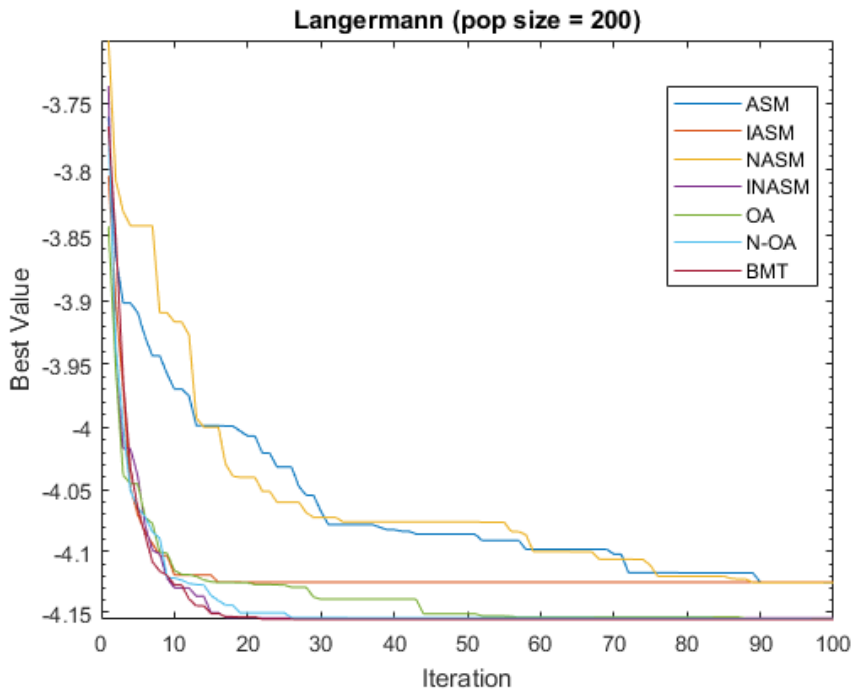

Easom (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

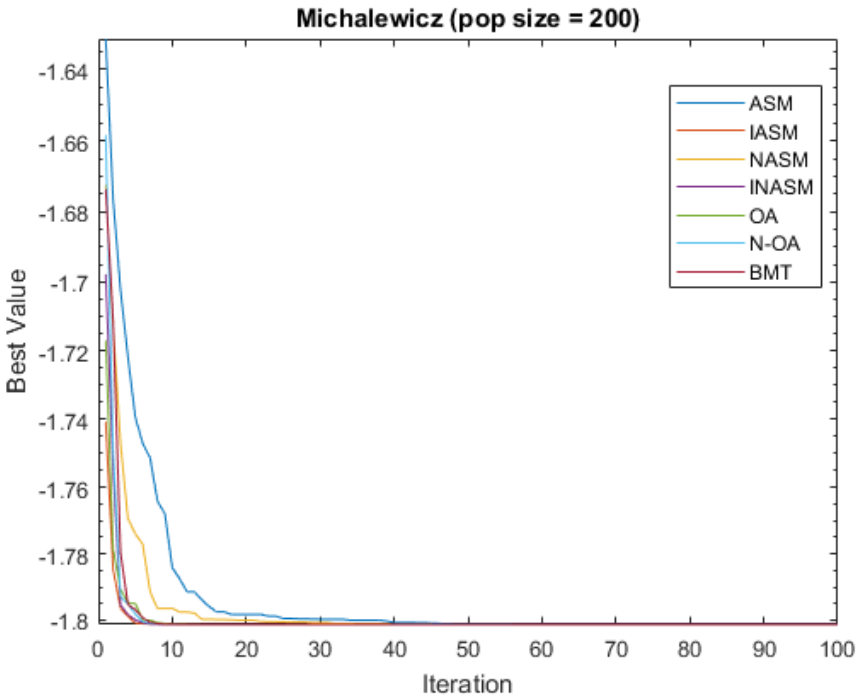

Drop Wave (pop size = 200)
Best Value
Iteration
-0.92
-0.93
-0.94
-0.95
-0.96
-0.97
-0.98
-0.99
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

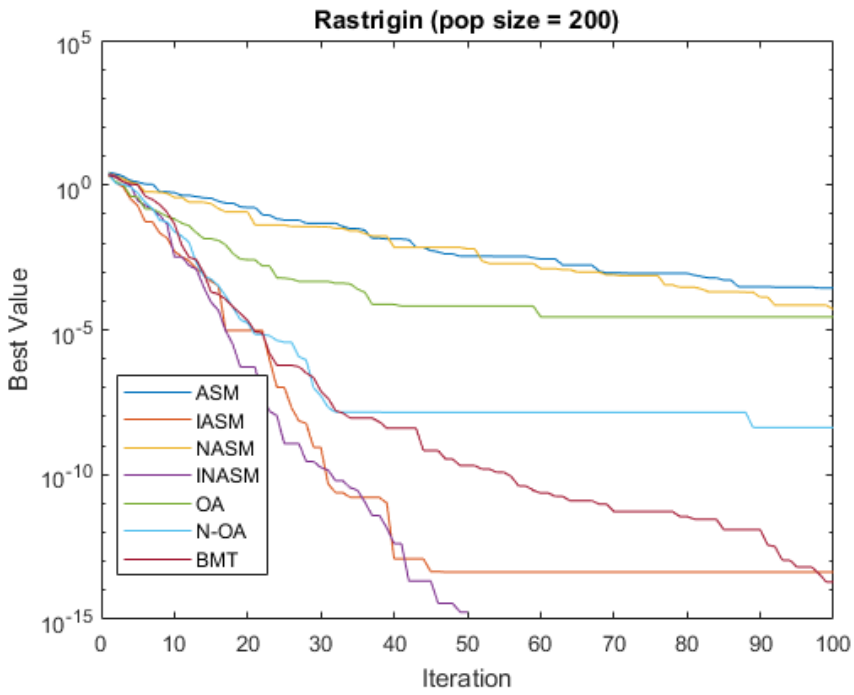

Goldstein-Price (pop size = 200)
Best Value
Iteration
8
7
6
5
4
3
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

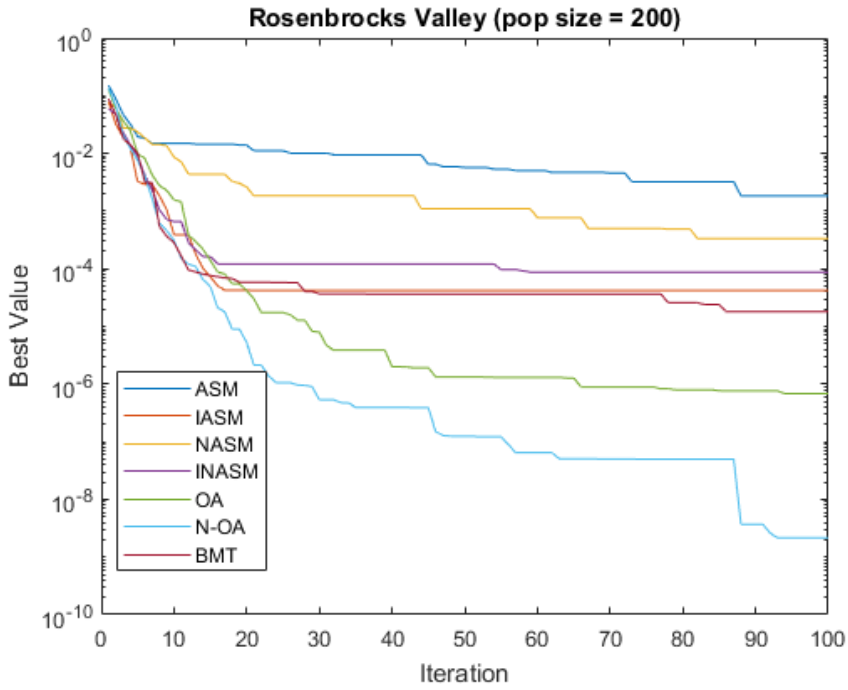

Griewangk (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

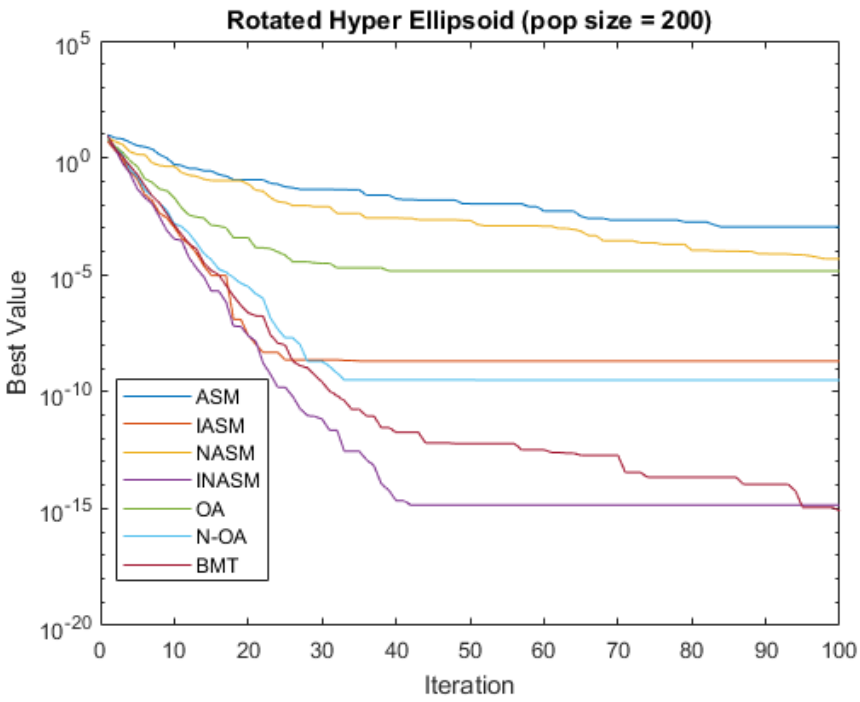

Holder (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

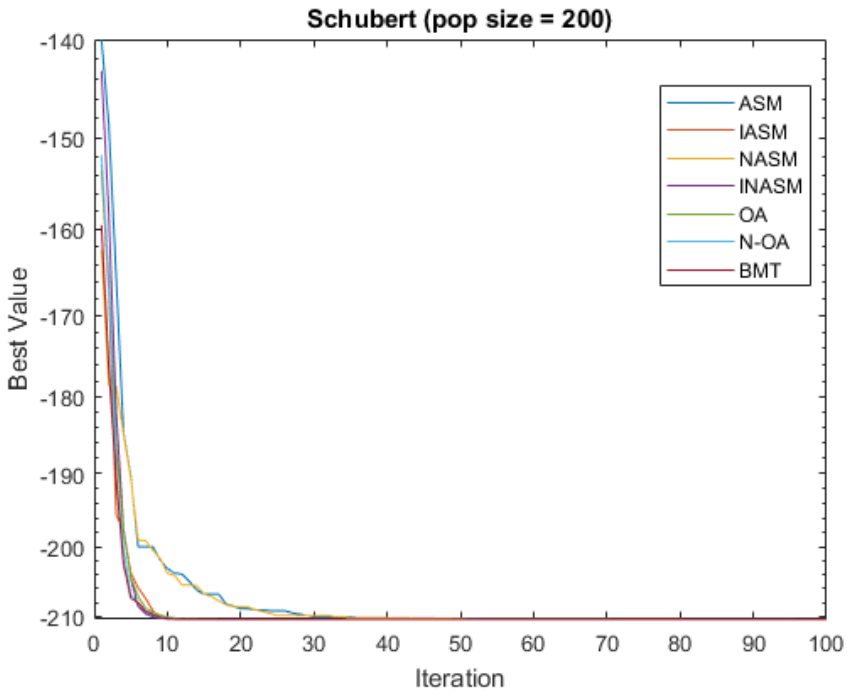

Langermann (pop size = 200)
Best Value
Iteration
-3.75
-3.8
-3.85
-3.9
-3.95
-4
-4.05
-4.1
-4.15
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

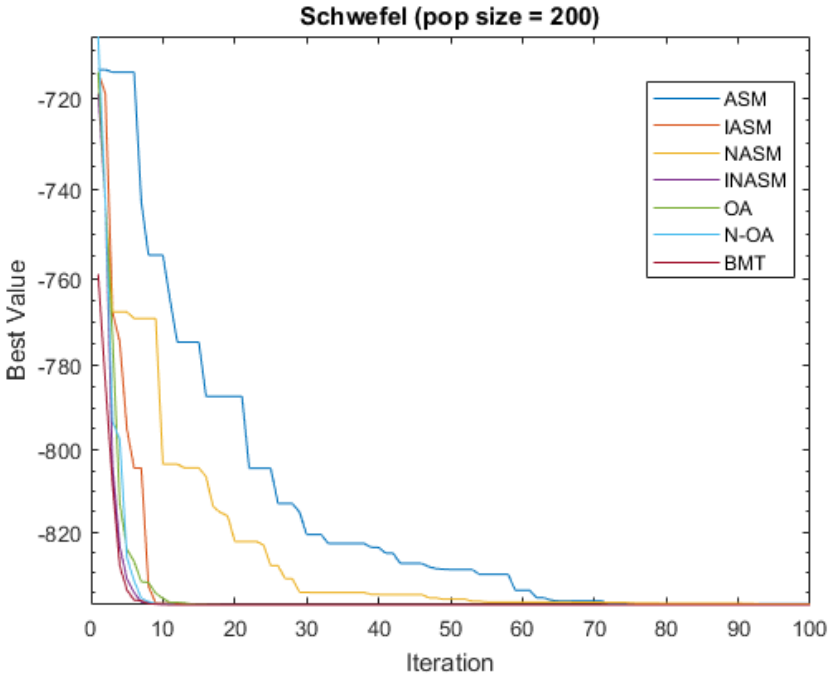

Michalewicz (pop size = 200)
Best Value
Iteration
-1.64
-1.66
-1.68
-1.7
-1.72
-1.74
-1.76
-1.78
-1.8
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT

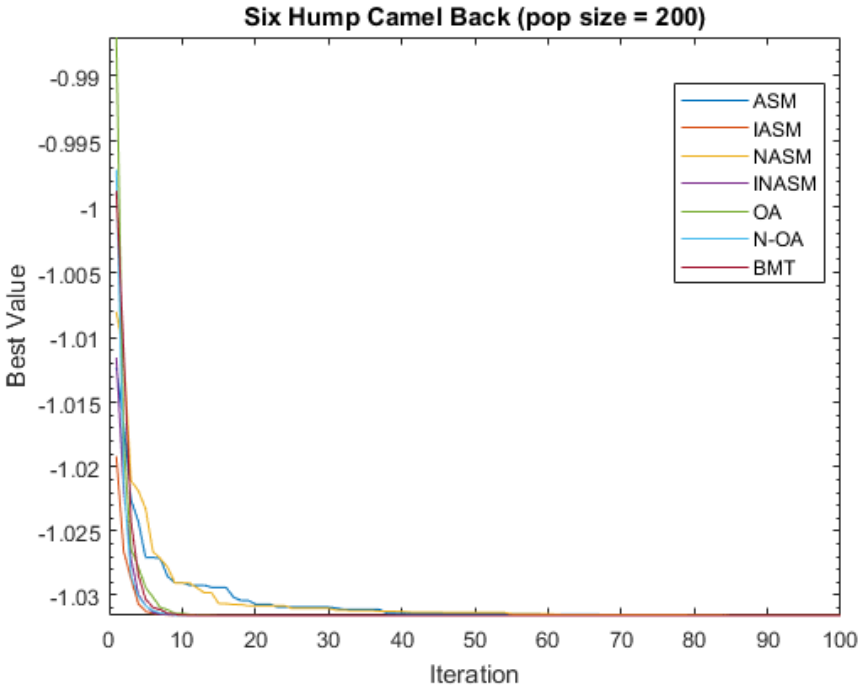

Rastrigin (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT

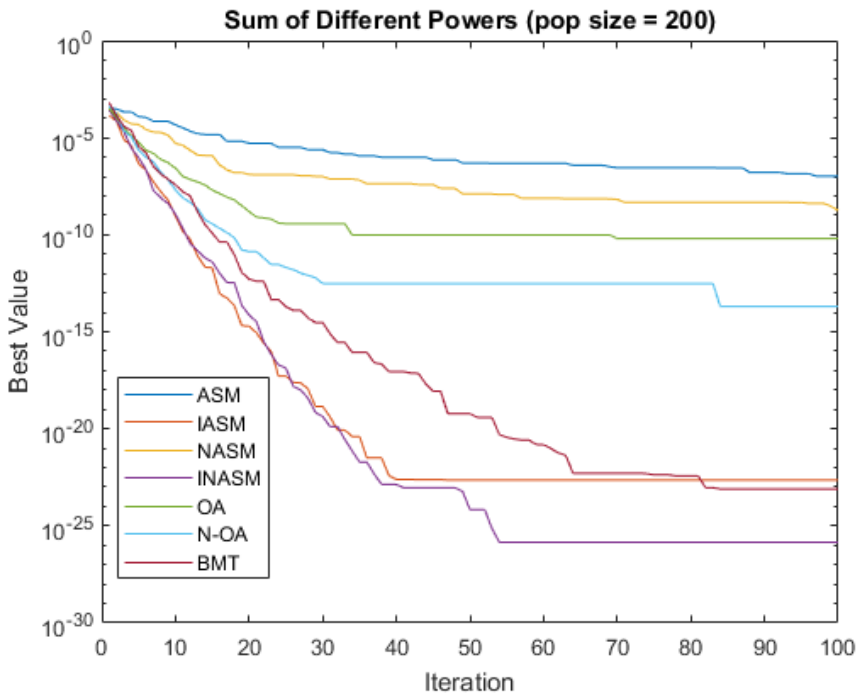

Rosenbrocks Valley (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT


Rotated Hyper Ellipsoid (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT


Schubert (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT


Schwefel (pop size = 200)
Best Value
Iteration
-720
-740
-760
-780
-800
-820
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT


Six Hump Camel Back (pop size = 200)
Best Value
Iteration
-0.99
-0.995
-1
-1.005
-1.01
-1.015
-1.02
-1.025
-1.03
0
10
20
30
40
50
60
70
80
90
100
ASM
IASM
NASM
INASM
OA
N-OA
BMT


Sum of Different Powers (pop size = 200)
Best Value
Iteration
ASM
IASM
NASM
INASM
OA
N-OA
BMT