Adaptive Multiobjective Memetic Optimization: Algorithms and Applications

by

Hieu V. Dang

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

Adaptive Multiobjective Memetic Optimization: Algorithms and Applications

by

Hieu V. Dang

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in fulfillment of the requirements for the degree of

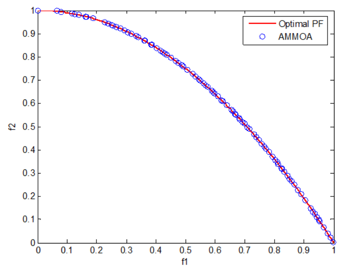
Doctor of Philosophy

Department of Electrical and Computer Engineering

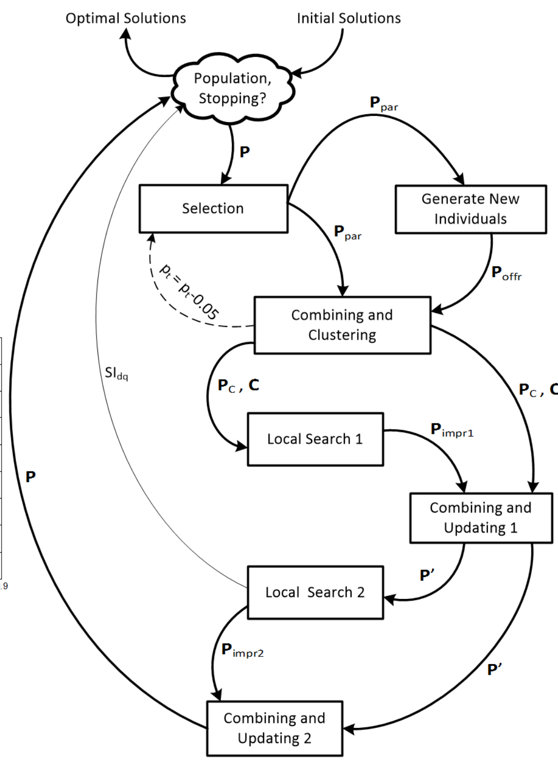
University of Manitoba

Winnipeg

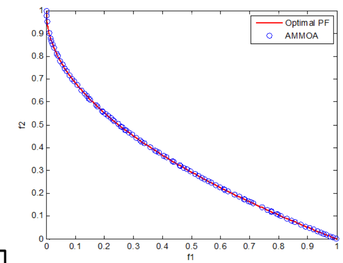
(xxv + 169 + A5 + B5 + C3 + D11 + E3) = 221 pp.



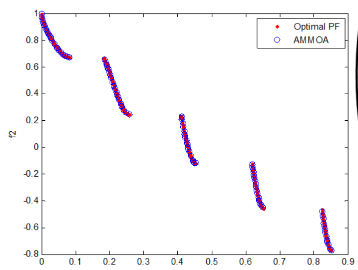
Optimal PF
AMMOA
f1
f2



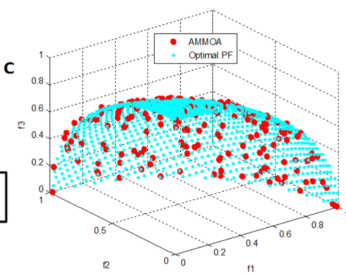
Optimal Solutions
Initial Solutions
Population, Stopping?
P
P_{par}
Selection
Generate New Individuals
P_{par}
P_{offr}
$p_t = p_t - 0.05$
Combining and Clustering
SI_{dq}
P_C , C
P_C , C
Local Search 1
P_{impr1}
Combining and Updating 1
P
P’
Local Search 2
P_{impr2}
P’
Combining and Updating 2



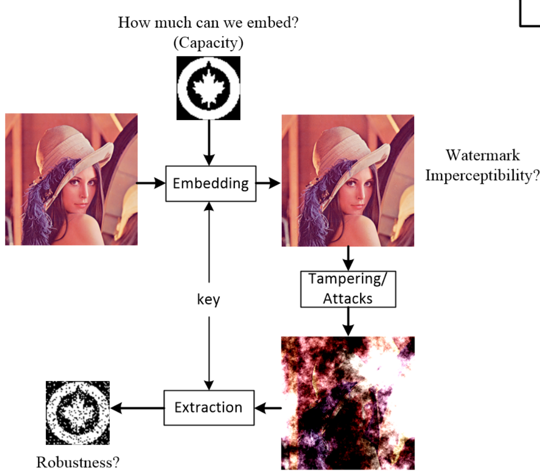
Optimal PF
AMMOA
f1
f2



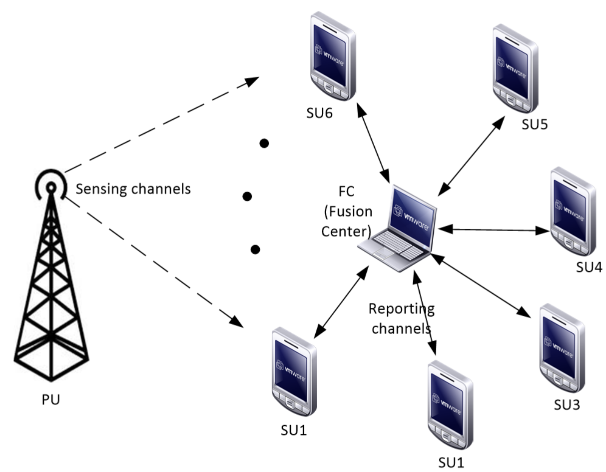
Optimal PF
AMMOA
f1
f2



AMMOA
Optimal PF
f1
f2
f3



How much can we embed?
(Capacity)
Embedding
Watermark
Imperceptibility?
Tampering/
Attacks
key
Extraction
Robustness?



Sensing channels
PU
SU6
SU5
FC
(Fusion
Center)
SU4
Reporting
channels
SU3
SU1
SU1

Abstract

Multiobjective optimization is an important research area because of its broad applicability in science and engineering. The idea of adopting nature-inspired computing, such as bio-inspired computation and culture-inspired computation (memetic), into multiobjective optimization has recently attracted much attention from the research communities. Memetic computing inspired by the theory of meme for the human cultural evolution can favour multiobjective optimization greatly because of its hybrid combination of both global and local learnings for a fast convergence.

The thesis presents research on multiobjective optimization based on memetic computing and its applications in engineering. We have introduced a framework of *adaptive multiobjective memetic optimization algorithms* (AMMOA) with an information theoretic criterion for guiding the selection, clustering, and local refinements. A robust stopping criterion for AMMOA has also been introduced to solve non-linear and large-scale optimization problems. The framework has been implemented for different benchmark test problems with remarkable results.

This thesis also presents two applications of these algorithms. First, an optimal image data hiding technique has been formulated as a multiobjective optimization problem with conflicting objectives. In particular, trade-off factors in designing an optimal image data hiding are investigated to maximize the quality of watermarked images and the robustness of watermark. With the fixed size of a logo watermark, there is a conflict between these two objectives, thus a multiobjective optimization problem is introduced. We propose to use a hybrid between *general regression neural networks* (GRNN) and the adaptive multiobjective memetic optimization algorithm (AMMOA) to solve this challenging problem. This novel image data hiding approach has been implemented for many different test natural images with remarkable robustness and transparency of the embedded logo watermark. We also introduce a perceptual measure based on the relative Rẽnyi information spectrum to evaluate the quality of watermarked images.

The second application is the problem of joint spectrum sensing and power con-

trol optimization for a multichannel, multiple-user cognitive radio network. We investigated trade-off factors in designing efficient spectrum sensing techniques to maximize the throughput and minimize the interference. To maximize the throughput of secondary users and minimize the interference to primary users, we propose a joint determination of the sensing and transmission parameters of the secondary users, such as sensing times, decision threshold vectors, and power allocation vectors. There is a conflict between these two objectives, thus a multiobjective optimization problem is used again in the form of AMMOA. This algorithm learns to find optimal spectrum sensing times, decision threshold vectors, and power allocation vectors to maximize the averaged opportunistic throughput and minimize the averaged interference to the cognitive radio network.

The main contributions of this thesis include: (*i*) A framework of adaptive multiobjective memetic optimization algorithms (AMMOA) is introduced for multiobjective optimization problems with remarkable results; (*ii*) A novel multiobjective image data hiding method is proposed for colour image protections; and (*iii*) A new approach for multiobjective joint spectrum sensing and power control is presented for a multichannel and multiple-user cognitive radio network.

Acknowledgements

I would like to express my great appreciation to my supervisor, Prof. Witold Kinsner for his countless help and guidance throughout my doctoral studying years. His dedication and encouragement have helped me go far beyond my expectations. I have benefited a lot from the valuable discussions with him. Especially, his emphasis on the integrity and the reproducibility in research has greatly guided my research life.

I would like to thank my committee members and examiners, Prof. Ekram Hossain, Prof. Pourang Irani, Prof. Abba Gumel, and Prof. B. John Oommen for many useful discussions and sincere advice.

I would also like to express my gratitude and special thanks to Vietnam International Education Development (VIED) and the University of Manitoba for providing the financial assistance.

Thanks to my parent, brothers, and sisters for their encouragement and motivation in me throughout the years. My deepest grateful goes to my wife and my son who stayed next to me during my Ph.D. pursuit. I express my gratitude and love for their emotional support and for loving me.

Finally, thanks to all my fellows at the Delta Research Group who have helped directly and indirectly to support me throughout this undertaking.

Contents

List of Tables

List of Figures

List of Abbreviations

AMMOA	Adaptive multiobjective memetic optimization algorithm
AWGN	Additive white Gaussian noise
BPNN	Back-propagation neural networks
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DWT	Discrete wavelet transform
EA	Evolutionary algorithm
FCM	Fuzzy c-mean clustering
FPGA	Field programmable gate arrays
GA	Genetic algorithm
GRNN	General regression neural networks
HVS	Human visual system
IDWT	Inverted discrete wavelet transform
IGD	Inverted generational distance
JPEG	Joint photographic experts group
MA	Memetic algorithm
MAC	Medium access control
MMOA	Multiobjective memetic optimization algorithm
MOEA	Multiobjective optimization using evolutionary algorithm
MOGLS	Multi-objective genetic local search
MOP	Multiobjective optimization problem
MSE	Mean squared error
NSGA	Non-dominated sorting genetic algorithm
NSGA-II	Non-dominated sorting genetic algorithm 2

PAES	Pareto archived evolutionary strategy
PF	Pareto front
PNN	Probabilistic neural networks
PS	Particle swarm
PSNR	Peak signal to noise ratio
PSO	Particle swarm optimization
PU	Primary user
RRFDS	Relative Rényi fractal dimension spectrum
RRSE	Rényi relative summation entropy
SBX	Simulated binary crossover
SF	Scaling factor
SNR	Signal to noise ratio
SPEA2	Strength Pareto evolutionary algorithm 2
SU	Secondary user
SVM	Support vector machine
WAR	Watermark accuracy ratio

List of Symbols

$CL\{i\}$	Cluster i^{th}
$\boldsymbol{CL}$	The set of clusters CL
C_N	Colour noises
$d(\bullet)$	Distance (Euclidean distance)
D_{Rq}	R$\acute{e}$nyi fractal dimension spectrums
$F_p(n)$	Fibonacci p-code sequence
$\boldsymbol{f}(\mathbf{x})$	$M-$dimensional objective vector consisting of M objective functions
$f_i(\mathbf{x})$	Objective function
f^p	The p-norm distance function
$f^{Rq}(\bullet)$	R$\acute{e}$nyi divergent function
G	Averaged transmission gain of the cognitive radio network
H_{Rq}	R$\acute{e}$nyi relative entropy
$H_{ri}(k)$	The cross-channel transfer function between the secondary transmitter r and the secondary receiver i
$\mathcal{H}_{k\|0}$	The hypothesis represents the absence of the primary signal over the subcarrier k
$\mathcal{H}_{k\|1}$	The hypothesis represents the presence of at least one primary user
I_i^{SU}	The interference created by the secondary user SU_i
I	The original input image for an image data hiding
I_w	The watermarked image obtained after embedding the watermark into the original image
I_{peak}	The peak value of pixel in the image I
I_w^N	Noise corrupted watermarked image

$I_w^{C_N}$	The watermarked image corrupted by a colour noise
I_H	The homogeneity index of clustering
I_S	The separation index of clustering
I_Q	The clustering quality index
M	Number of objectives
N_p	Number of individuals in the population $\boldsymbol{P}$
n	Number of variables in one decision vector $\mathbf{x}$
N_{CL_i}	The number of individuals in the cluster
N_c	The number of clusters $\boldsymbol{CL}$
$p(\mathbf{x})$	Relative probability of solution $\mathbf{x}$
$\boldsymbol{P}$	Set of feasible solutions (population)
p_m	Probability of mutation
$P_{i,k}^{fa}$	False alarm probability in spectrum sensing of user i at the channel k in cognitive radio network
$\tilde{\boldsymbol{p}}^e$	The optimal power allocation vector obtained from multiobjective optimization for cognitive-radio users
$P_{i,k}^{d}$	Probability of detection in spectrum sensing of user i at the channel k
$P_{i,k}^{md}$	Probability of missed detection in spectrum sensing of user i at the channel k
$\boldsymbol{P}_{par}$	Selected parent population
$\boldsymbol{P}^*$	Pareto-optimal set
$\boldsymbol{P}_f^*$	Pareto-optimal front
$\boldsymbol{P}_{offr}$	Offspring population
$\boldsymbol{P}_C$	Intermediate combined population
$\boldsymbol{P}_{impr}$	Improved population after local refinement
p_t	Probability of picking up a parent for the mating pool in the parent selection process

$\boldsymbol{P}_{init}$	Initial population of solutions
$\boldsymbol{P}_I$	Intermediate population obtained by combining an improved population and the current population
p_{local}	Probability of local search
p_x	Probability of crossover
q	Moment order of the probabilities in Rényi entropy
R_{RSE}	Rényi relative summation entropy
$\boldsymbol{r}$	Reference vector for relative probability vectors
R	Averaged throughput of the cognitive radio network
$R^{\mathbf{x}}_{RSE}$	Rényi relative summation entropy of the solution $\mathbf{x}$
$S_{i,k}[n]$	The primary signaling fromt the primary user of the cognitve-radio network
T_k	Random integer numbers generated by the Fibo p-code sequence
u_{ki}	Membership function of the data element i in the cluster k
$\boldsymbol{U}$	Matrix of the membership functions
v_k	Center of the cluster k^{th}
W	The logo watermark
$\tilde{W}$	The extracted logo watermark
$\mathbf{x}$	A decision variable vector
$\boldsymbol{\mathcal{X}}$	An n-dimensional search space of decision variable vectors
$\mathbf{x}^*$	Optimal decision variable vector
$x_i^{(t)}$	An i^{th} decision variable of the decision vector $\mathbf{x}$ at the generation t
x_i^U	Upper bound of the variable x_i
x_i^L	Lower bound of the variable x_i

$\mathbb{S}$	The bag denoted for the host signal in the information theoretical model of data hiding
$\boldsymbol{\tau}$	The sensing time vector for cognitive-radio users
$\boldsymbol{\gamma}$	The decision-threshold vector for cognitive-radio users
$\tilde{\boldsymbol{\tau}}$	The optimal sensing time obtained from multiobjective optimization for cognitive-radio users
$\tilde{\boldsymbol{\gamma}}$	The optimal decision-threshold vector obtained from multiobjective optimization for cognitive-radio users
$\mathbb{M}$	The bag denoted for the hidden message (watermark)
$\mathbb{K}$	The bag denoted for the secret key shared between the encoder and the decoder
$\mathbb{U}$	The bag denoted for the embedded signal
$\mathbb{V}$	The bag denoted for the received (attacked) signal
$\hat{\mathbb{M}}$	The bag denoted for the extracted hidden message
$\boldsymbol{s}_{\mathbb{S}}$	The subbag denoted for an instance of the host signal
$\boldsymbol{m}_{\mathbb{M}}$	The subbag denoted for an instance of the hidden message
$\boldsymbol{k}_{\mathbb{K}}$	The subbag denoted for an instance of the secret key
$\boldsymbol{u}_{\mathbb{U}}$	The subbag denoted for an instance of the embedded signal
η_c	Distribution index of a crossover operator
$\boldsymbol{v}_{\mathbb{V}}$	The subbag denoted for an instance of the received (attacked) signal
$\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}}$	The subbag denoted for an instance of the extracted hidden message
$\vartheta(\bullet)$	Embedding (encoding) function
$\varphi(\bullet)$	Extracting (decoding) function
σ	Standard deviation of a Gaussian distribution

η_m	Distribution index of a mutation operator
$\eta(i)$	The embedding factor for the i^{th} watermark bit
$\bar{\oplus}$	The not XOR (XNOR) logic operator
$\boldsymbol{\lambda}$	Random normalized weights

Publications

- <u>Journal Publications</u>:

    1. Hieu V. Dang and Witold Kinsner, "A multiscale information theoretic criterion for adaptive multiobjective memetic optimization," Submitted to *IEEE Trans. Evolutionary Computation,* (Submitted for publication, 2015).
    2. Hieu V. Dang and Witold Kinsner, "Multiobjective multivariate optimization of joint spectrum sensing and power control in cognitive wireless networks," *International Journal of Cognitive Informatics and Natural Intelligence* (Accepted for publication, Aug. 2015).
    3. Hieu V. Dang, Witold Kinsner, and Yingxu Wang, "Multiobjective image data hiding based on neural networks and memetic optimization," *WSEAS Trans. Signal Processing*, vol. 10, pp. 645–661, Dec. 2014.
    4. Hieu V. Dang and Witold Kinsner, "A perceptual data hiding mathematical model for color image protection," *Journal of Advanced Mathematics and Applications*, vol. 1, no. 2, pp. 218–233, Dec. 2012.

- <u>Conference Publications</u>:

    1. Hieu V. Dang and Witold Kinsner, "An analytical multiobjective optimization of joint spectrum sensing and power control in cognitive radio networks," To appear in *Proc. of the 14th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing*, ICCI*CC 2015, (China, Beijing; July 6–8, 2015), pp. 39–48, 2015.
    2. Hieu V. Dang and Witold Kinsner, "Multiobjective memetic optimization for spectrum sensing and power allocation in cognitive wireless networks," in *Proc. of the Canadian Conf. on Electrical and Computer Engineering*, CCECE 2014, (Torronto, Canada; May 4–7), pp. 1–6 , 2014.

3. Hieu V. Dang, Witold Kinsner, and Yingxu Wang, "Optimal colour image watermarking using neural networks and multiobjective memetic optimization," in *Proc. of the 2014 Intern. Conf. on Neural Networks and Fuzzy Systems*, ICNN-FS 2014, (Venice, Italy; March 15-17, 2014), pp. 63–74, 2014.

4. Hieu V. Dang and Witold Kinsner, "An intelligent digital color image watermarking approach based on wavelet transform and general regression neural network," in *Proc. of the 11th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing*, ICCI*CC 2012, (Kyoto, Japan; August 22-24, 2012), pp. 115-123, 2012.

Chapter 1

Introduction

1.1 Motivation

Optimization is one of the most important areas in engineering, computer science, and many other fields. In engineering, dealing with uncertainty is critical. The need for a reliable decision in every engineering designs is highly demanding, especially, a decision process with multiple objectives and constraints. It is a challenge for engineers to find an optimal solution of a multiple-objective designing problem. It is a difficult issue because this decision making is a non-linear optimization problem with many variables and multiple objectives. Such multiobjective optimization problems have multiple optimal solutions instead of one optimal solution as in single-objective optimization problems. Based on the designing missions and constraints, one can select an optimal solution in this set of optimal solutions for a specific design problem. Therefore, designing a framework to help engineers solve those multiple-objective decision problems is my foremost motivation for this thesis.

Stochastic optimization plays an important role in dealing with uncertainty in science and engineering. There exist different stochastic optimization approaches in the literature. A very important and effective approach is a population-based optimization. In the population-based optimization approach, the initial population consisting of the number of variables is created randomly first, the algorithm refines the population several times to find the best solution among the individuals in the population. *Evolutionary algorithms* (EAs) constitute one example in this category. Evolutionary algorithms are nature-inspired computing approaches, and include *genetic algorithms* (GAs), *particle swarm optimization* (PSO), and *memetic algorithms* (MAs). They are developed based on the mechanisms found in nature, such as gene evolution in GA, animal society in PSO, and recently the human cultural evolution in MA. The theory of memes introduced by Richard Dawkin for the human cultural evolution has been recently brought to the algorithmic field for MA. Current literature shows that MA outperforms GA in specific single-objective optimization problems. The main idea of MA is to adopt a local search into an EA global search to improve the convergent speed of the searching algorithm. However, when applying MAs to multiobjective optimization problems, MAs find it difficult to maintain the diversity of the solutions or the population in each iteration. The performance of MAs can be improved much by applying the idea of the human cultural evolution carefully. Thus, designing an effective framework (i.e., improving the convergent speed and maintaining the diversity of solutions) of multiobjective memetic optimization based on the human cultural evolution is also my important motivation behind this thesis.

In order to demonstrate the operational characteristics of the new approach, two application areas have been selected: information hiding and cognitive radio. Information hiding is a technique of embedding information (watermark) into a carrier signal (video, image, audio, text) such that the watermark can be extracted or detected later for copyright protection, content authentication, identity, fingerprinting, access control, copy control, covert communications, and broadcast monitoring. Information hiding has recently attracted so much attention because of its broad applications. Designing an optimal information hiding is a decision process under uncertainty considerations. It requires to consider many perspectives, such as signal quality and integrity, embedding capacity, security, and the robustness of watermark. Therefore, it is a challenging multi-objective decision making process. I am motivated in designing an optimal and adaptive image data hiding as one of the applications of my work in multiobjective optimization based on memetic computing.

Cognitive radio is a major invention and development in the wireless communication field. The cognitive radio concept has been proposed to be the next generation wireless devices that can share underutilized spectrum. Spectrum sensing and dynamic spectrum access are main principles of cognitive radio. In spectrum sensing, cognitive-radio users (*secondary users* - SUs) sense the spectrum of licensed users (*primary users* -PUs) to detect and utilize spectrum holes within the PUs' spectrum. The challenge for a reliable sensing algorithm is to identify suitable transmission opportunities without compromising the integrity of the PUs. The efficiency of the employed spectrum sensing technique plays a key role in maximizing the cognitive radio network throughput, while protecting the PUs from interference. Hence, I also get motivated to design an efficient model that is able to maximize the throughput of the cognitive network based on *adaptive multiobjective memetic optimization algorithms* (AMMOA).

1.2 Criteria for Selecting Applications

Designing optimal image data hiding and designing effective spectrum sensing and power control in cognitive radio networks are the selected applications for demonstrating the applicability of AMMOA. Our motivation and objective of selecting applications are beyond the verification as what can be done by adopting simple benchmark test problems. The selected applications should be able to demonstrate the applicability and ability of

AMMOA to cope with the difficulties of nonlinear, complex, and evolutionary learning systems in fast-changed environments. In addition to our experience in these two applications, the following groups of selection criteria have been considered: (*i*) partially conflicting objective spaces; (*ii*) nonlinear and adaptive learning systems; (*iii*) complexity of optimization process; and (*iv*) orthogonality and diversity.

1.2.1 Partially Conflicting Objective Spaces

Multiobjective optimization problems can be categorized as non-conflicting multiobjective problems, totally-conflicting multiobjective problems, or partially-conflicting multiobjective problems. For a non-conflicting multiobjective problem, the various objectives are correlated and the optimization of one objective leads to the subsequent improvement of the other objectives. Thus, a single objective optimization can be used to solve this problem by aggregating different objectives into a scalar function. For a totally-conflicting multiobjective problem, all feasible solutions are also optimal, and no optimization is required. The partially-conflicting multiobjective problems are perhaps the most common real world problems. In this class of multiobjective problems, there is a set of solutions representing the trade-offs between the different objectives instead of an unique solution. This category is the most challenging of the three, and requires for an effective multiobjective optimization technique.

Both designing optimal image data hiding and designing effective spectrum sensing and power control problems are partially-conflicting multiobjective optimization problems. In image data hiding, with a fixed size of a logo watermark, there is a conflict between the transparency (the quality of the watermarked image) and robustness of the watermark. In cognitive radio networks, maximizing the throughput of the network and minimizing the interferences are the objectives of network designers. These two objectives are partially-conflicting and requiring a joint determination of the sensing and transmission parameters. These two applications are very challenging multiobjective optimization problems.

1.2.2 Nonlinear and Adaptive Learning Systems

We are interested in nontrivial engineering applications featured by nonlinear and adaptive learning systems. Those problems require learning abilities to adapt the system parameters/structure to the changes of environments. Wide-range and dynamic searching space is so common for those problems that are always challenging to be solved. Besides, the sensitivity of learning systems to environments is also needed to investigate carefully.

Both designing optimal image data hiding and designing effective spectrum sensing and power control in cognitive radio networks are nonlinear and adaptive learning systems. In image data hiding, the embedding system learns from the environment by extracting the image and environmental attacking features to adapt watermark embedding parameters to maximize the transparency, robustness, and capacity objectives. Any changes of those environmental features lead to change the parameters of the watermarking system. This adaptive and evolutionary learning concept is more important in the spectrum sensing and power control of cognitive radio networks, where the sensing and transmission parameters are required to be updated cognitively with the changes of the environment to maximize the throughput of the network and minimize interference to the network.

1.2.3 Complexity of Optimization Process

From our perspective, the complexity of a optimization process is featured by the complexity of searching landscapes, the high nonlinearity and sensitivity, and the scale of the problems. It is very challenging if the problem has a dynamic searching landscape and a high level of sensitivity. The selected applications should possess a high level of complexity, sensitivity, and scalability.

Both designing optimal image data hiding and designing effective spectrum sensing and power control in cognitive radio networks are highly-complex optimization problems. They are nonlinear and large-scale multiobjective problems where the searching landscapes are dynamical and sensitive with the environments' changes. While scalability of image data hiding problem mainly depends of the size of input images, the scalability of the spectrum sensing and power control problem depends on the structure and the size of the networks (the number of users and the number of operating channels).

1.2.4 Dissimilarity

Dissimilarity means that the computational features of the selected applications should be different. For instance, while both the design of optimal image data hiding and the design of effective spectrum sensing and power control in cognitive radio networks are complex multiobjective optimization problems, the complexity of the later is very scalable with respect to the size and structure of the network.

While the multiobjective function of the image data hiding problem is highly complex since it is obtained through watermark embedding and extraction processes which relate to different computational blocks (e.g., wavelet transform, neural networks, and evolutionary learning systems), the multiobjective function of the spectrum sensing and power control problem can be obtained through numerical analyses.

While the multiobjective function of the image data hiding problem is non-differentiable, the multiobjective function of the spectrum sensing and power control problem is differentiable. In our design, while the image data hiding problem does not require real-time computation ability, the spectrum sensing and power control problem requires real-time computation capacity. These requirements lead to different designs regarding different speeds of convergence.

1.3 Problem Definition and Research Questions

The work presented in this thesis relates to three following research areas: (*i*) designing adaptive multiobjective memetic optimization; (*ii*) developing an optimal image data hiding; and (*iii*) designing an effective spectrum sensing and power control mechanism for cognitive radio networks. Thus, the research problems and questions which the thesis tries to solve can be grouped into the following three categories.

1.3.1 Designing Adaptive Multiobjective Memetic Optimization

There are different approaches to solve a multiobjective optimization problem (MOP). The first approach is to scalarize the M objective functions into a single-objective function. In this approach, the multiobjective optimization problem is scalarized to a single-objective optimization problem. The final solution of the MOP based on this method is a single solution. The main challenge of this approach is to improve the convergent speed of the optimization method. Another approach is to choose one objective function out of M to be minimized; the remaining objectives are constrained to given target values. This process is repeated M times for the M objective functions. With this approach, the algorithm obtains one solution for each optimization process. The optimal solution is the best solution from the M resulting solutions. This approach may not lead to a global optimal solution for partially-conflicting multiobjective optimization problems. Besides, it is challenging to setup wise constrained target values for the objective functions since we do not know the objective search landscapes in practice. Furthermore, this approach is too expensive in computation because it has to implement M optimization processes. The third approach is to find a set of optimal solutions instead of a single solution in the scalarization approach. This approach is usually mentioned as the Pareto-based optimization approach. The Pareto-based approaches are more advanced than the scalarization approaches since they are more cognitive-oriented. However, the main challenges are both for increasing the convergent speed and maintaining the diversity of the optimal solutions in the Pareto-optimal set. In this work, we focus on the Pareto-based optimization approach to solve large-scale multiobjective optimization problems.

The evolutionary algorithms (EAs) are suitable for multiobjective optimization because they are able to obtain a set of improved solutions in a single run. Multiobjective optimization using evolutionary algorithms (MOEAs) have been introduced and applied successfully in the literature, including NSGA-II [DPAM02, Deb01], and SPEA2 [ZiLT01]. The memetic algorithms (MAs) has been recently proposed to improve the convergent speed of EAs for single-objective optimization problem by employing a strategy of both local learning (exploitation) and global learning (exploration). However, when applying this memetic strategy for a multiobjective optimization, this strategy destroys the diversity of the population seriously with an improper strategy for local and global searches.

The challenging research questions in the design of an effective memetic algorithm for solving a multiobjective optimization problem are: (*i*) What is the best strategy for the hybrid between global and local searches (i.e., compromise of exploration and exploitation) in multiobjective optimization based on memetic algorithms (MOMA)? (*ii*) Can an MOMA detect the convergence to the Pareto optimal front by itself? (or what are the online stopping criteria for MOMA?) (*iii*) Can we design MOMAs able to converge in high-dimensional objective spaces, while maintaining good diversity? (*iv*) Can an MOMA learn during the evolution to reveal previous unknown information about the structure of the black-box optimization problems? (*v*) Can we design an MOMA that can be close to the cultural evolution in which an individual can interact fully and learn from its neighbours (a small community) and the population (a big community) to grow? and (*vi*) Why do the multiobjective optimization algorithms based on non-dominated sorting techniques (e.g., Pareto ranking methods) scale poorly with respect to the number of objectives?

1.3.2 Designing Optimal Image Data Hiding

Designing an optimal image data hiding is a multiobjective decision process under uncertainty considerations because it must consider many perspectives, such as signal quality and integrity, embedding capacity, security, and the robustness of watermark. The important requirements for image data hiding systems are imperceptibility, robustness, capacity, and security under different attacks and varying conditions. These requirements can vary under different applications. It is challenging to design a blind watermarking for a logo data hiding where a logo watermark is embedded into the host signal, and then extracted from the embedded host signal without a reference from the original host signal. These visual watermarks are not only assessed by machines but also by humans through their ability to recognize visual patterns through *human visual system* (HVS). It is very difficult to extract a high-quality visual logo from the embedded host image without any reference from the original host signal.

Transparency and robustness are two main challenges in logo watermarking techniques since the logo consists of much information that is not easy to embed perceptually into a host signal. The transparency means that the watermark should be invisible in the host signal. The robustness refers to the ability of the hidden watermark to survive common attacks such as signal processing operations (compression, filtering, noise addition, desynchronization, cropping, and information insertions). In logo watermarking, the robustness is so strict because it requires satisfactory recognition by human beings.

With a fixed size of a logo watermark, there is a conflict between the transparency and robustness of the watermark. Increasing the transparency of watermark (or the quality of the watermarked image) decreases the robustness of the watermark and vice versa. A good logo watermarking is a robust data hiding with the acceptable quality of watermarked image. Thus, an optimal logo watermarking is a challenging multiobjective optimization problem.

The challenging research questions in the design of an optimal image data hiding, which we focus in this work, are: (*i*) What are the trade-off factors and conflict objectives in designing an optimal logo watermarking for colour images? (*ii*) Why is wavelet decomposition suitable for perceptual image data hiding? (*iii*) Can we design an optimal wavelet function and transform for an optimal multiobjective image data hiding? (*iv*) Why are the energy based measures, including *signal to noise ratio* (SNR) and *peak signal to noise ratio* (PSNR) not sufficient for evaluating the imperceptibility (transparency) of an image data hiding technique? and (*v*) Can we design an adaptive data hiding technique that is able to self-adjust its parameters in regards with the attacks and environmental changes?

1.3.3 Designing Effective Spectrum Sensing and Power Control in Cognitive Radio Networks

In cognitive radio networks, it is challenging to design a reliable sensing strategy to identify suitable transmission opportunities for cognitive radio users (SUs) without compromising the integrity of the network primary users (PUs). An effective spectrum sensing technique plays a key role in maximizing the throughput of the cognitive network. However, maximizing the throughput of the cognitive radio users leads to increase interference to the primary users. Maximizing the throughput of the network and minimizing the interference to the primary users are the objectives of the network designer. However, these two objectives are seriously conflicting and requiring a joint determination of the sensing and transmission parameters. Therefore, designing an effective spectrum sensing and power control in cognitive radio networks is a challenging joint multiobjective decision making problem with a large number of variables. There exists a set of optimal solutions instead of only one solution for this joint optimization. Depend on the network's context and operational budgets, the best-suited solution is selected to configure the network.

The challenging research questions in the design of an effective spectrum sensing and power control strategy in cognitive radio networks, which we consider in this work, are: (*i*) What are the trade-off factors and conflicting objectives in designing an efficient spectrum sensing mechanism for cognitive wireless networks? (*ii*) Why does spectrum sensing have to be considered together with power control in cognitive wireless networks? and (*iii*) Can we design an efficient cooperative joint spectrum sensing and power control in cognitive wireless networks?

1.4 Thesis Statement

This thesis investigates the following issues: (*i*) the speed of convergence and the diversity of Pareto-optimal solutions of multiobjective optimization using evolutionary algorithms (MOEAs) can be improved significantly by applying the theory of memetic learning in human cultural evolution to design a framework of adaptive multiobjective memetic optimization algorithms (AMMOA); (*ii*) an information-theoretic criterion based on the polyscale relative Rényi entropy can be applied to guide the adaptive processes of AMMOA, and the online convergence detection of AMMOA effectively; (*iii*) image data hiding can be formulated as a multiobjective optimization problem which can be solved effectively based on wavelet transforms, neural networks, and AMMOA; and (*iv*) spectrum sensing and power control in cognitive radio networks can be formulated as a multiobjective joint optimization problem which can be solved effectively by using AMMOA.

1.5 Thesis Objectives

The main part of the thesis is the study of multiobjective memetic optimization algorithms and their applications. The theory of multiobjective optimization and memetic computing is studied to design a framework for adaptive multiobjective memetic optimization algorithms (AMMOA). The framework and its implementation are then verified by its applications to solve challenging problems in information hiding and cognitive wireless networks. In summary, our main objectives cover the followings:

- To study and apply the theory of memes for the human cultural evolution to the memetic computing. This is important to gain an understanding of the mechanism of the human cultural evolution to bring this knowledge into algorithmic design.

- To design an efficient framework for adaptive multiobjective optimization based on memetic computing (AMMOA). In particular, the framework is able to model the human cultural evolution wisely, and is more efective than currently-used multiobjective optimization based on evolutionary algorithms. The framework should be also able to detect the convergence to the optimal front with online stopping criteria.

- To apply the framework of adaptive multiobjective optimization in designing an optimal image data hiding for copyright protection and authentication applications. In particular, we are going to apply AMMOA to find an optimal solution to maximize the quality of the watermarked image and the robustness of the watermark.

- To apply the AMMOA framework in designing an efficient cooperative joint spectrum sensing and power control in cognitive radio networks. We are going to apply AMMOA to find the optimal trade-off factors in spectrum sensing and power control to maximize the network throughput and to reduce the interferences.

1.6 Thesis Scope and Contributions

The thesis presents an investigation of memetic evolutionary algorithms to design and implement a framework of adaptive multiobjective memetic optimization algorithm (AMMOA). AMMOA is implemented and tested with variety of multiobjective optimization benchmark test problems with remarkable results.

The thesis also presents a study of perceptual and robust image data hiding techniques deeply. An optimal logo watermarking is proposed for colour image protection based on wavelet transforms, neural networks, and AMMOA. A perceptual multiscale measure based on the relative Rényi fractal dimension is also proposed to evaluate the transparency of the logo watermarking method with remarkable results.

The thesis also presents a research on the problem of designing an optimal spectrum sensing and power control in cognitive radio networks. We have formulated this problem as a multiobjective joint optimization problem. AMMOA is proposed to solve this problem with good results.

The major contributions of this thesis are listed as follows:

- *A novel and efficient framework of adaptive multiobjective memetic optimization algorithms* (AMMOA). Based on the theories of memetic computation and multiobjective optimization, we develop a framework of AMMOA. In this framework, we develop a new information theoretic criterion for guiding the adaptive selection, clustering, and local searches in each evolution of AMMOA. An online stopping criterion is also introduced to guide AMMOA to detect its convergence. This novel framework is implemented for different multiobjective benchmark test problems and real-world engineering design applications with remarkable results. The comparison results show that AMMOA is superior to the well-known multiobjective optimization NSGA-II for both high convergent speed and diversity of the obtained optimal set.

- *A novel multiobjective image data hiding method.* The trade-off factors and conflicting objectives in designing an optimal image data hiding are studied carefully to formulate a multiobjective optimization problem of image data hiding. We then develop a novel multiobjective image data hiding method for colour images based on wavelets, general regression neural networks (GRNN), and AMMOA. The proposed method obtains a high level of robustness and imperceptibility of logo watermark, and is more superior to the well-known image watermarking techniques in the literature. We also introduce a multiscale perceptual measure for measuring the transparency (imperceptibility) of the watermark with remarkable results.

- *A new approach for multiobjective joint spectrum sensing and power control in cognitive radio networks.* The trade-off factors and conflicting objectives in designing an effective spectrum sensing and power control method is studied carefully. We introduce a multiobjective joint optimization problem of spectrum sensing and power allocation for a multichannel, multiple-user cognitive radio network. We then develop a novel multiobjective joint spectrum sensing and power allocation technique based on AMMOA. This is the first approach to solve the joint optimization problem of spectrum sensing and power allocation using multiobjective optimization. The experimental results show the practical applicability of the method to increase the adaptiveness and cognition of cognitive radio networks.

1.7 Thesis Organization

The thesis is organized into six chapters:

- Chapter 2 introduces a brief background on multiobjective optimization. This chapter gives short introductions to the field of multiobjective optimization, including the definitions, theories, and the literature review.

- Chapter 3 presents our proposed framework of adaptive multiobjective memetic optimization (AMMOA). Additionally, an implementation of this framework is also described in this chapter. The experimental results and comparisons with the well-known multiobjective optimization based on non-dominated sort genetic algorithm (NSGA-II) are also presented.

- Chapter 4 presents a novel application of AMMOA in designing an optimal multiobjective image data hiding. The problem of perceptual and robust data hiding for colour images is first introduced. A novel image data hiding based on neural networks, wavelets, and AMMOA is then presented. The experimental results and comparisons with discussions are also presented in this chapter.

- Chapter 5 introduces a new approach for joint spectrum sensing and power control in cognitive wireless networks. The problem of joint spectrum sensing and power allocation in cognitive radio is discussed with a careful investigation of trade-off factors. A novel cooperative joint spectrum sensing and power allocation approach based on AMMOA is presented with experimental results and discussions.

- Chapter 6 is with the conclusions and the possible extensions of this research.

- Finally, the thesis ends with 124 references covering from 1970 to 2015 with the mode at 2012 (the reference statistics are shown in Figure. 1.1), followed by five appendixes.

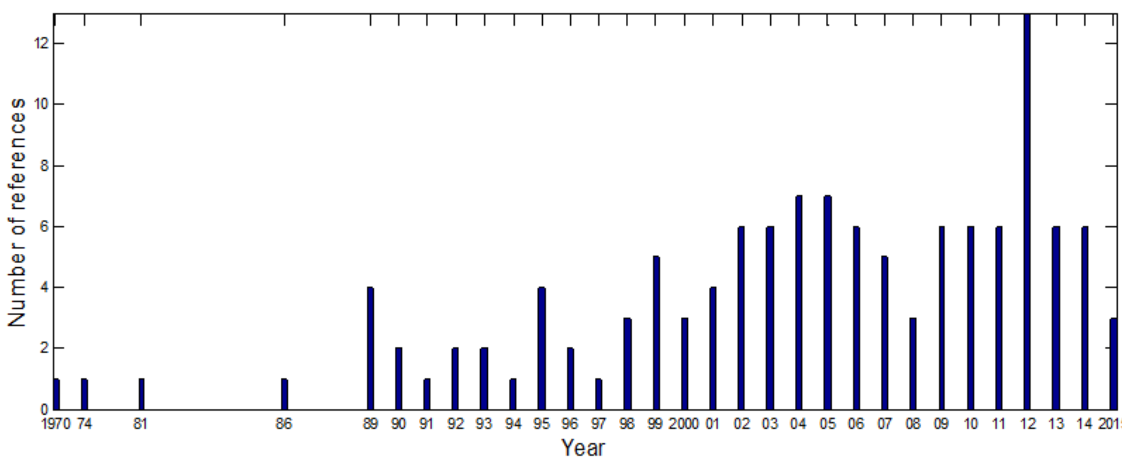


Figure 1.1: Statistics of the references used in this thesis.

Chapter 2

Literature Review on Multiobjective Optimization

This chapter presents some background on multiobjective optimization including definitions, theories, and literature reviews. The theoretic issues and applications of multiobjective optimization based on evolutionary algorithms are also presented.

2.1 Introduction

Multiobjective optimization is considered as a decision making process to get the most optimal solution with the maker's preference from the best obtained results. Multiobjective optimization deals with the function with more than two objectives. In most practical decision making problems, there are multiple objectives or multiple criteria as evidences. For instance, in engineering design, we do want to design a smart, long life, energy efficiency, and lower cost machine. However, to design a smart, long life, and energy efficiency machine, we need to have high quality of materials, component, and intellectual properties. These things are very expensive. Thus, these objectives are conflict. A good designer should be able to find a solution that can be balance between these objectives or follows the preference of the investors. Therefore, a multiobjective optimization problem should be adopted to find the optimal solutions.

Unfortunately, these real world problems are often difficult, if not possible, to be solved without advanced and efficient optimization techniques. This is because these problems are characterized by objectives that are much more complex as compared to the single-objective optimization problems. Because of that, a multiobjective optimization problem has been mostly combined and solved as a single-objective optimization problem. However, this method is just looking for one solution instead of a set of optimal solutions.

In the literature, *evolutionary algorithms* (EAs) are widely used to solve nonlinear optimization problems. Due to the nature of EA that has the capability of finding a set of Pareto solutions in a single simulation run, EA is suitable for multiobjective optimization.

In this chapter, we provide a brief introduction to the multiobjective optimization problems, evolutionary algorithms (EAs), *memetic algorithms* (MAs), multiobjective optimization using EAs (MOEA), and the performance evaluations of MOEA.

2.2 Multiobjective Optimization Problems

2.2.1 Problem Formulation

Multiobjective optimization is sometimes referred to vector optimization. Let $\boldsymbol{\mathcal{X}}$ be an n-dimensional search space of decision variable vectors $\mathbf{x} = \{x_1, x_2, ..., x_n\}$. A multiobjective optimization problem can be generally formulated as

$$\begin{aligned} \underset{\mathbf{x}\in\boldsymbol{\mathcal{X}}}{\text{maximize/minimize}}\ \boldsymbol{f}(\mathbf{x}) &= \{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), ..., f_M(\mathbf{x})\} \\ \text{subject to } \boldsymbol{g}(\mathbf{x}) &= \{g_1(\mathbf{x}), g_2(\mathbf{x}), ..., g_K(\mathbf{x})\} \geq 0 \\ \boldsymbol{h}(\mathbf{x}) &= \{h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_L(\mathbf{x})\} = 0 \end{aligned} \tag{2.1}$$

There are K inequalities and L equalities constraints. Functions $\boldsymbol{g}(\mathbf{x})$ and $\boldsymbol{h}(\mathbf{x})$ are called constraint functions. The objective function $\boldsymbol{f}(\mathbf{x})$ is an M-dimensional vector that consists of M objective functions $\{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), ..., f_M(\mathbf{x})\}$. The optimization problem can be to maximize or minimize the objective function vector. According to the duality principle [Deb01], we can convert a maximization problem into a minimization problem by multiplying the objective function vector by -1.

If all objective functions and constraint functions are linear, the multiobjective optimization problem is called a multiobjective linear problem. Otherwise, if any of the objective or constraint functions are nonlinear, the problem is called a nonlinear multiobjective optimization problem [Miet99, Deb01]. For a multiobjective linear problem, the principle of linear programming can be used to solve this problem with the proof of convergence; however, for nonlinear multiobjective optimization problem the current solvers do not have convergence proofs. Unfortulately, most real world problems are nonlinear multiobjective problems in nature. In this thesis we focus on desining multiobjective optimization methods to solve nonlinear multiobjective optimization problems.

Multiobjective optimization problems have a multi-dimensional objective space. Depending on whether the objectives are conflicting or non-conflicting, a multiobjective problems can be categorized as [GoTa09]: (*i*) totally conflicting multiobjective problem; (*ii*) non-conflicting multiobjective problem; and (*iii*) partially conflicting multiobjective problem. For the first category, no improvement of objectives can be made without violating any constraints. That means all feasible solutions are also optimal. Thus, totally conflicting multiobjective problems are perhaps the simplest since no optimization is required. On the other hand, in a non-conflicting multiobjective problem, the various objectives are correlated and the optimization of one objective leads to the subsequent improvement of the other objectives. A single objective optimization can be used to solve this problem by aggregating different objectives into a scalar function. The solution is a single optimal solution. The third category, partially conflicting multiobjective problem, is perhaps the most common real-world problems. In this class of multiobjective problems, there is a set of solutions representing the tradeoffs between the different objectives instead of an unique optimal solution. This category is the most challenging of the three.

There are two fundamental differences between single-objective optimization and multiobjective optimization. The first difference is that in single-objective optimization, the search for an optimum global solution is the only one goal. However, in multiobjective optimization, the Pareto-optimal solutions (Pareto front) and the diversity of the optimal solutions are two important goals. The second difference is that in single-objective optimization, there is only one search space (the decision variable space). Nonetheless, in multiobjective optimization, there are two search spaces (the objective space and the decision variable space).

There exist different techniques to solve multiobjective problems. Without losing generality, multiobjective optimization algorithms can be categorized into two groups: (i) algorithms that use the combinations of objectives to select new solutions; (ii) algorithms that do not combine objectives and do the selection by means of dominance based criteria [NeCo12]. In the first category, the multiple objectives are combined to create a single objective by adopting a weight values. Thus, the algorithm does not detect an optimal front or a set of optimal solutions, but only one solution. This class of algorithms has the drawback that the selection of a proper set of weights must be performed to allow a natural dispersion of the solutions. The outcome of such an optimization strategy depends on the chosen weights. In the second approach, the selection is based on dominance-based ranking of all the solutions. The search result is a set of optimal solutions or the so called Pareto front. Based on the knowledge of such multiple optimal solutions a designer can compare and choose a compromised optimal solution. The idea of the Pareto's optimality and Pareto front is discussed in the next section.

2.2.2 Pareto Dominance and Optimality

The concept of domination is very important in multiobjective optimization algorithms. The domination concept is used to rank the solutions in the population. Specifically, two solutions are compared on the basis of whether one dominates the other solution or not [Deb01]. The following definitions of dominance and Pareto optimality are presented for the minimization problem.

Definition 2.1. Dominance: A solution $\mathbf{x}_1$ is said to dominate the other solution $\mathbf{x}_2$ if $f_i(\mathbf{x}_1) \preceq f_i(\mathbf{x}_2)$, $\forall i \in \{1, ..., M\}$, and $\exists j \in \{1, ..., M\}$: $f_j(\mathbf{x}_1) \prec f_j(\mathbf{x}_2)$.

If $\mathbf{x}_1$ and $\mathbf{x}_2$ satisfy Definition 2.1, we can say $\mathbf{x}_2$ is dominated by $\mathbf{x}_1$; or $\mathbf{x}_1$ is non-dominated by $\mathbf{x}_2$. Intuitively, if a solution $\mathbf{x}_1$ dominates the solution $\mathbf{x}_2$, the solution $\mathbf{x}_1$ is better than the solution $\mathbf{x}_2$ in the multiobjective optimization problem.

Definition 2.2. Weak Dominance: A solution $\mathbf{x}_1$ weakly dominates a solution $\mathbf{x}_2$ if $f_i(\mathbf{x}_1) \preceq f_i(\mathbf{x}_2)$, $\forall i \in \{1, ..., M\}$.

Definition 2.3. Strong Dominance: A solution $\mathbf{x}_1$ strongly dominates a solution $\mathbf{x}_2$ if $f_i(\mathbf{x}_1) \prec f_i(\mathbf{x}_2)$, $\forall i \in \{1, ..., M\}$.

From Definition 2.3, we see that the solution $\mathbf{x}_1$ strongly dominates the solution $\mathbf{x}_2$ if solution $\mathbf{x}_1$ is strictly better than solution $\mathbf{x}_2$ in all M objectives.

Definition 2.4. Non-Dominated Set: The non-dominated set of solutions $\boldsymbol{P}'$ includes non-dominated solutions that are not dominated by any member solution among the set of solutions.

Definition 2.5. Weak Non-Dominated Set: Among a set of solutions $\boldsymbol{P}$, the weak non-dominated set $\boldsymbol{P}'$ are those solutions that are not strongly dominated by any other member solution of the set $\boldsymbol{P}$.

The Definition 2.5 suggests that a weak non-dominated set contains all member solutions of the non-dominated set, which is obtained by the concept of Definition 2.1.

Definition 2.6. Pareto Optimal Set: The non-dominated set of the entire feasible search space $\mathcal{S}$ is the Pareto-optimal set $\boldsymbol{P}^*$. In other word, the Pareto-optimal set $\boldsymbol{P}^*$ is the set of solutions that are non-dominated in the objective space such that $\boldsymbol{P}^* = \{\mathbf{x}_i^* — \nexists\ f_k(\mathbf{x}_j) \prec f_k(\mathbf{x}_i^*),\ \forall i, j \in \{1, ..., n\}, \forall k \in \{1, ..., M\}\}$.

Definition 2.7. Pareto Optimal Front: The Pareto-optimal front $\boldsymbol{P}_f^*$ is the set of non-dominated solutions with respect to the objectives in the objective space such that $\boldsymbol{P}_f^* = \{f_i^* | \nexists f_j \prec f_i^*, \forall i, j \in \{1, ..., n\}\}$

These above principles can be explained by a graphical presentation in the Fig. 2.1, where these relationships of Pareto front, Pareto optimal set, the non-dominated set, and the dominated set for a bi-objective minimization problem are presented graphically.

2.3 Multiobjective Optimization Using Evolutionary Algorithms (MOEA)

In classical methods of solving multiobjective optimization, the obtained result is a single optimal solution. Since the real world optimization problems are multiobjective optimization with conflicting objectives, a single optimized solution is not sufficient to characterize the problem and does not help the decision maker much.

The *Evolutionary algorithms* (EAs) mimic the nature's evolutionary principles to constitute search and optimization procedures. EAs are population based algorithms that use a population of solutions in each iteration, instead of a single solution in classical methods. The outcome of EA is also a population of solutions. Thus, an EA can be

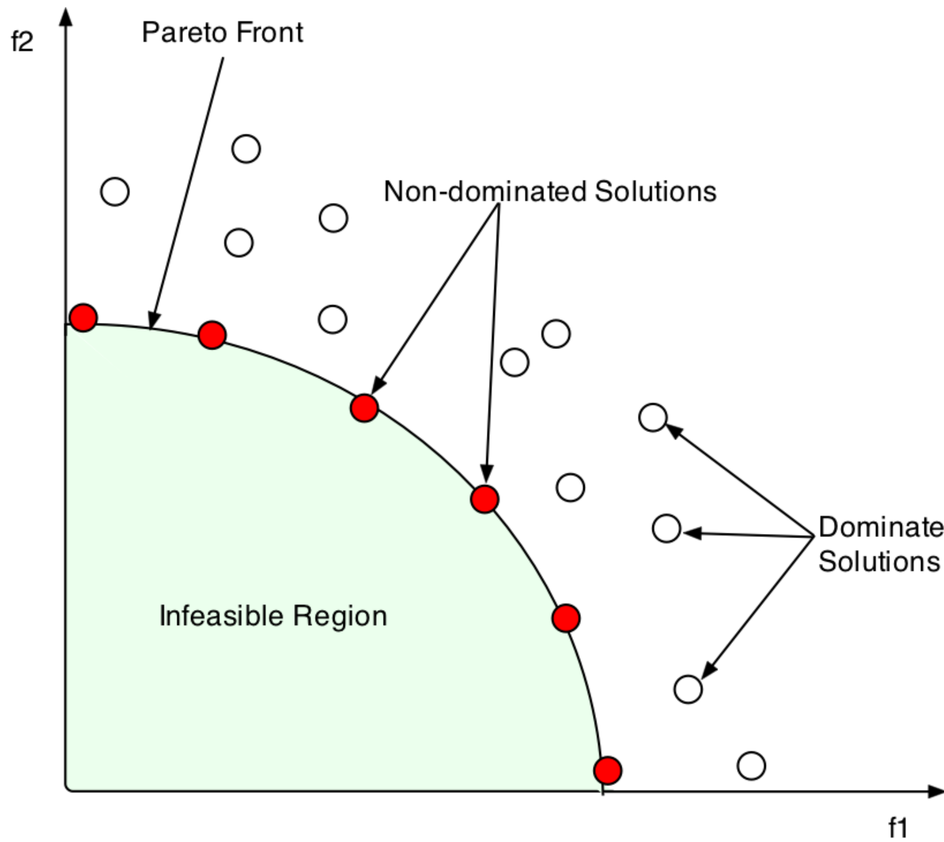


Figure 2.1: Illustration of dominance and Pareto optimality.

efficiently used to capture multiple optimal solutions in one single simulation run for multiobjective optimization problems.

2.3.1 Principles of Evolutionary Algorithms

EAs are stochastic search methods that mimic the metaphor of natural biological evolution. EAs operate on a population of potential solutions. By applying the principle of survival of the fitness to produce better approximations, the population of potential solutions is improved after each iteration. Specifically, at each generation, a new set of approximations is created by the process of selecting individuals (i.e., based on their fitness) and breeding them together using genetic operators (e.g., crossover, mutation) borrowed from natural genetic evolution. This process leads to the evolution of population that are better suited to their environment as in the natural adaptation. EAs model natural processes such as selection, recombination, mutation, and migration. The structure of a simple EA is shown in Fig. 2.2.

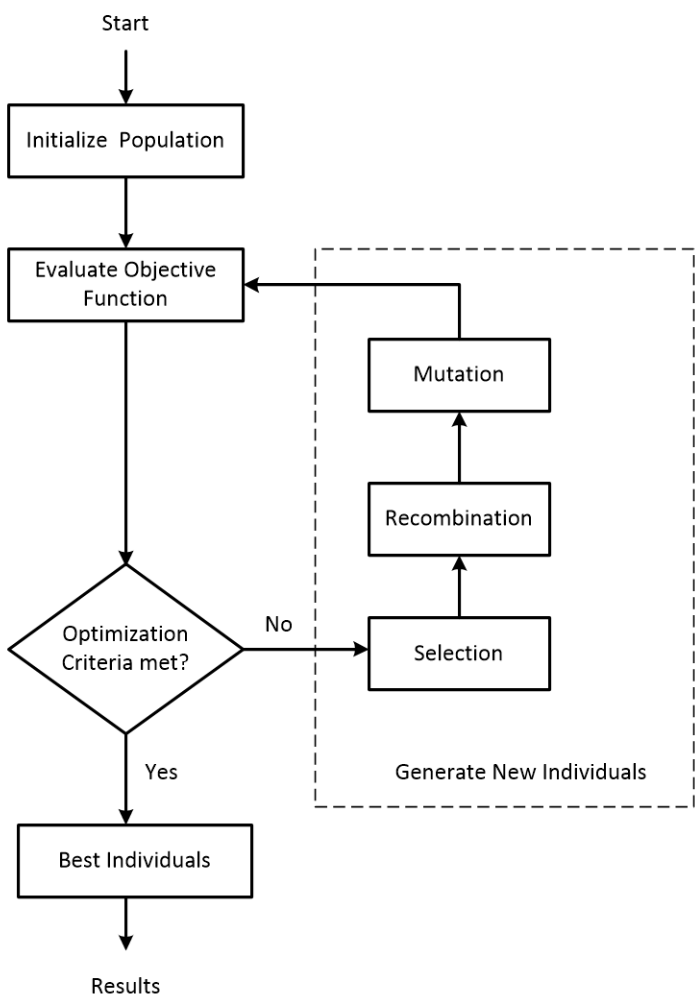


Figure 2.2: Structure of a simple EA.

At the beginning of an EA, a population of individuals is initialized randomly. The objective function is then evaluated for these individuals. The evolutionary process starts with the first generation if the optimization criteria are not met. Based on their fitnesses, individuals are selected as parents for generating the production of offsprings. Parents are recombined by using a genetic operator like crossover to produce offsprings. All offspring individuals are mutated with a mutation probability. The offspring individuals are then evaluated by the objective evaluation function to compute their own fitnesses. The offspring individuals are then inserted into the population to compete the parents to produce a new and improved population. The cycle continues until the optimization criteria are met.

It can be seen that EAs are different from classical search and optimization methods by the following points.

- EAs do not require derivative information or other auxiliary approximation knowledge, instead of only the objective function and the corresponding fitness levels for the evolutionary search.
- EAs search for a population of solutions in parallel, not a single individual in a simulation run.
- EAs use stochastic rules, not deterministic rules.
- EAs provide a number of potential solutions to a given problem. The final choice is left for the decision marker.

EAs are stochastic optimization methods. The differences between stochastic optimization and deterministic optimization can be summarized as follows.

- In stochastic optimization, there is a random choice made for the search direction in each step of the algorithm. In contrast, classical deterministic optimization assumes that perfect information is available about the objective function (e.g., derivatives of the objective function) and that information is used to determine the search direction in a deterministic manner at every step of the algorithm.

- Stochastic optimization methods offer only a guarantee of probabilistic optimal solution, while deterministic optimization methods provide a theoretical guarantee of locating the global optimum, or at least a local optimum whose objective function value differs by $\epsilon > 0$ from the global one.

- Stochastic optimization methods adapt better to black-box problems and extremely ill-behaved functions, whereas deterministic optimization methods usually rely on at least some theoretical assumptions about the problem formulation and its analytical properties.

The basic components of an EA are discussed in the following sections.

Selection

The selection process is used to select or identify good individuals for mating (recombination) pool. There are different selection methods currently used in the literature. Some common approaches are proportionate selection, ranking selection, and tournament selection [Gold89, Deb01, Mess15].

In the proportionate selection technique, solutions are assigned copies with the number of copies proportional to their fitness values. Specifically, if the average fitness of all population members is F, a solution with a fitness f_i gets an expected f_i/F number of copies. This technique is analogous to a roulette wheel with each slice proportional in size to the fitness. This technique is also called the roulette-wheel selection (also called stochastic sampling with replacement). This is perhaps the simplest selection scheme. This technique is implemented as follows. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. A random number is generated and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals is obtained for the mating pool. An example of the proportionate selection with the population size of

6 individuals with their fitnesses and corresponding selection probabilities are described in Table 2.1. Suppose we need to select 3 individuals from the population. The random number generator generates the random selection probabilities as $\{0.065, 0.5, 0.75\}$. The selected individuals are then individual # 1, 2, and 4. The implementation of this selection technique is shown in Fig. 2.3.

Table 2.1: Fitness values and selection probabilities

Individuals	1	2	3	4	5	6
Fitness Values	20	40	5	15	12	8
Selectrion Probability	0.2	0.4	0.05	0.15	0.12	0.08

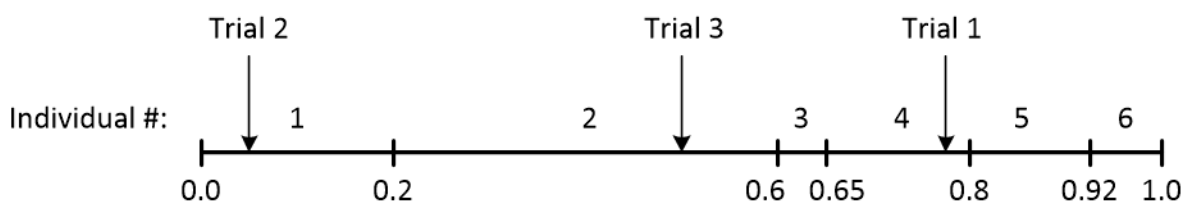


Figure 2.3: An example of proportionate selection.

The proportionate selection has a scaling problem that the outcome of the selection operator depends on the true value of the fitness. If one solution in the population has a large fitness value compared to the rest of the population members, the probability of choosing this solution would be close to 1, thereby dominating the mating pool with its copies.

The ranking selection method is an improvement of the proportionate selection to avoid a scaling problem. In ranking selection method, the solutions are firstly sorted according to their fitness. Each member in the sorted list is assigned a relative fitness equal to the rank of the solution in the list. The proportionate selection operator is then applied with the ranked fitness values to select N solutions for the mating pool.

In a tournament selection, two solutions are played a tournament and the better solution is chosen and placed in the mating pool. Two other solutions are placed in a tournament again and the better is chosen to fill the mating pool. The process continues until the mating pool is filled fully. Each solution can be made to participate in exactly two tournaments. The best individual wins both times, thereby making two copies of it in the new population. The worst individual loses in both tournaments and is eliminated from the population. It is easily seen that the tournament selection does not have any scaling problems. By changing the comparison operator, the minimization and maximization problems can be handled easily with the tournament selection method.

Recombination

Recombination (crossover operator in genetics) is used to produce new individuals by combining the information contained in the parental individuals in the mating pool. Depending on the representation of the variables of the individuals, recombination can be categorized into real-valued recombination or binary-valued recombination. In the very first beginning of the development of EAs, binary crossover operators are commonly used. It can be a single point crossover or multiple point crossover. However, most optimization problems in practice are with real-valued variables. To use binary crossover operators, these variables have to convert to binary-valued variables first before applying a binary crossover. Thus, there are some difficult problems such as (i) the Hamming cliffs associated with certain strings; (ii) the inability to achieve any arbitrary precision in the optimal solution; and (iii) the increase of the computational complexity. In this section, we briefly introduce real-valued recombination operators.

In real-valued recombination, the operators are applied directly to real parameter values. Thus, solving real-valued optimization problems using real-valued recombination is easier when compared to the techniques of using binary-coded operators. There exists a number of real-valued recombination implementations in the literature, such as linear crossover, Naive crossover, Blend crossover, simplex crossover, fuzzy operators, simulated binary crossover [DeAg95, Gold89, Deb01]. The issue of comparisons between crossovers is context-dependent. Evaluation of crossover operators depends on the chosen selection operator for the balance between exploitation and exploration for one successful EA's simulation run. Most of these crossovers have two common characteristics: (*i*) try to maintain the mean of population; and (*ii*) try to increase the population diversity. The simulated binary crossover (SBX) is used widely in successful EAs in the literature. In this section, we briefly introduce SBX as follows.

SBX operator simulates the principle of the single-point binary crossover on binary strings. The procedure of computing the offspring $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ from the parents $x_i^{(1,t)}$ and $x_i^{(2,t)}$ is described as follows. First, a random number $u_i \in [0, 1)$ is generated. The ordinate parameter β_{qi} is calculated by

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}}, & \text{if } u_i \leq 0.5; \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise.} \end{cases} \tag{2.2}$$

The offsprings are then calculated as

$$x_i^{(1,t+1)} = 0.5\left((1+\beta_{qi})x_i^{(1,t)} + (1-\beta_{qi})x_i^{(2,t)}\right) \tag{2.3}$$

$$x_i^{(2,t+1)} = 0.5\left((1-\beta_{qi})x_i^{(1,t)} + (1+\beta_{qi})x_i^{(2,t)}\right) \tag{2.4}$$

Where the parameter η_c is a non-negative distribution index. A large value of η_c gives a higher probability for creating solutions near to the parents. A small value of η_c allows distant solutions to be selected as offsprings. For a fixed η_c the offspring have a spread which is proportional to $\left(x_i^{(2,t+1)} - x_i^{(1,t+1)}\right) = \beta_{qi}\left(x_i^{(2,t)} - x_i^{(1,t)}\right)$. The SBX has an important property that the difference (distance) between the offsprings is proportional to the difference (distance) between the parent individuals.

Mutation

After recombination, every offspring solutions undergo mutation. The need for mutation is to keep diversity in the population. Offspring individuals are mutated by the addition of small random values (size of the mutation step), with low probability. The probability of mutating a variable is usually set to be inversely proportional to the number of variables (dimensions). The larger dimension, an individual has, the smaller mutation probability is applied to that individual. Mutation operators are also classified into binary-valued mutation and real-valued mutation, based on the data representation of individuals in the population.

In binary-valued mutation, the bit-wise procedure is randomly applied to one or more bits in the bit string variable of the individual with a small probability. In real-valued EAs, real-valued mutation is used to do a local perturbation for real parameter EAs. There exists some common real-valued mutation operators such as random mutation, non-uniform mutation, normally distributed mutation, and polynomial mutation [Deb01].

In the polynomial mutation [DeGo96], the probability distribution is considered as a polynomial function. The mutation is performed by

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^U - x_i^L)\bar{\delta}_i \tag{2.5}$$

where x_i^U is the upper bound of variable x_i, x_i^L is the lower bound of variable x_i. $\bar{\delta}_i$ is calculated from a polynomial distribution by

$$\bar{\delta}_i = \begin{cases} (2u_i)^{\frac{1}{\eta_m+1}} - 1, & \text{if } u_i < 0.5 \\ 1 - (2(1-u_i))^{\frac{1}{\eta_m+1}}, & \text{if } u_i \geq 0.5 \end{cases} \tag{2.6}$$

where u_i is a random number $\in [0, 1)$.

2.3.2 Multiobjective Optimization Using EAs

EAs have the important advantage of being able to sample multiple solutions simultaneously. This feature makes EAs common-used in multiobjective optimization (called multiobjective optimization using EAs - MOEA). An MOEA has the capability of finding a set of Pareto solutions in a single run. Genetic operators in EAs help MOEA create candidate solutions and exchange information between them to increase the overall quality of individuals in the population. In this sub-section, the principle and framework of MOEAs are described.

MOEA Framework

Many MOEAs have been proposed in the literature. Most of them are based on the models of *genetic algorithms* (GAs) [Deb01]. Recently, biologically inspired models, such as *partical swarm* (PS), *differential evolution* (DE) have been introduced for multiobjective optimization. The main difference between these approaches is in the method of generating new candidate solutions. In this section, we provide a framework of MOEA that can cover these approaches for multiobjective optimization. The framework of MOEA is described in Algorithm 1.

The MOEA starts with the initialization of the candidate population. The initial population is then evaluated by the objective evaluation function "Eval" for the objectives of each individuals in the population. In the main loop, parent population $\boldsymbol{P}_{par}$ is selected from the current population $\boldsymbol{P}$ by the procedure "Selection". The offspring population $\boldsymbol{P}_{offs}$ is generated from the parent population through the procedure "GenerateOffs". This new offspring population is then evaluated by the objective evaluation function, and ranking and diversity assessments. Updating population is then performed to select the better population from the parent population and the new offspring population by the "Updating" function. In each iteration, the offspring individuals compete with their

Algorithm 1 MOEA Framework

```
procedure MOEA()
    Generate Random Population P size N
    Objectives Evaluation P ← Eval(P)
    repeat
        Select Parent Population P_par ← Selection(P)
        Generate Offspring Population P_offs ← GenerateOffs(P)
        Objectives Evaluation P_offs ← Eval(P_offs)
        Ranking Assessment P_offs ← Ranking(P_offs)
        Diversity Assessment P_offs ← Diversity(P_offs)
        Update Population P ← Updating(P ∪ P_offs)
    until Terminated Conditions
    return Improved Population P
end procedure
```

parents for a better population. This circle continues until the terminated conditions, such as a number of iteration are met. The main components of MOEA framework are described as follows.

1. Ranking Assessment

Ranking assessment function (Ranking) is used to rank each individuals in the population based on their evaluated objectives. How to assign an effective ranking value for an individual (solution) in the population based on their multiple objectives is a challenging step in multiobjective optimization. This procedure is also called fitness assignment. There exist some different fitness assignment methods in the literature. We can categorize them into two main classes of fitness assignment: (i) Aggregation-based assignment; and (ii) Pareto-based assignment.

In aggregation-based assignment, multiple objectives of each individual are aggregated into a scalar by applying a weight vector [IsMu98, IsYM03, Jasz02, Jasz03, JiBr05, Hugh05]. The multiobjective genetic local search (MOGLS) [IsMu98, IsYM03, Jasz02, Jasz03] is a well-known example of applying aggregation-based fitness assignment for multiobjective optimization. In this approach, a different random weight vector is used to aggregate the multiple objectives of each individual to a fitness value in each simulation run. The simulations showed that this aggregation-based MOEA is capable of evolving uniformly distributed and diverse to converge to a Pareto front. The big difficulty of this approach is that it requires a pre-defined weights or a weight generating function for every simulation run. The performance is also too sensitive to the selected weight vectors. However, with the optimization problem having more than three objectives (many objectives or high-dimension multiobjective optimization problem), this approach seems more effective than the Pareto-based assignment approach [Hugh05, GoTa09].

In Pareto-based assignment, the fitness (rank) for each individual is formulated based on the property of non-dominated solutions and Pareto-optimality principle. The Pareto-based assignment is widely used for MOEA in the literature [Deb01, GoTa09, DPAM02, KnCo00, ZiLT01]. State-of-the-art MOEAs such as *non-dominated sorting genetic algorithm II* (NSGA-II) [DPAM02], *Pareto archived evolution strategy* (PAES) [KnCo00], and *strength Pareto evolutionary algorithm 2* (SPEA2) [ZiLT01] use Pareto-based assignment approach for ranking assessment. For instance, in NSGA-II, the Pareto fronts obtained after each simulation run are numbered ascendingly. The front index of each individual is then assigned as the ranking (or fitness) of that individual in the population. However, Pareto-based assignment MOEAS do not scale well with respect to increasing the number of objectives (e.g., more than three objectives) [Hugh05, FaPC10]. In the reports [Hugh05, FaPC10], Pareto-based assignment methods are more effective in low-dimensional multiobjective optimization problems, while aggregation-based assignment approaches scale well with increasing number of objectives.

2. Diversity Assessment

In MOEAs, diversity refers to the diversity of the obtained solutions in the objective space. In the final updated population, we desire to have a set of solutions that spread equally-well on the optimal Pareto front of the multiobjective optimization problem.

Density assessment is commonly used for diversity preservation in MOEAs. Density assessment evaluates the density at different sub-divisions in the parameter or objective spaces. Since we want to obtain a well-distributed and diverse Pareto front, density assessment in objective space is most used for diversity preservation in MOEAs. There exist different density assessment methods in the literature. Based on the metric used, they can be categorized into two main categories as: (i) distance-based assessment; and (ii) distribution-based assessment.

In distance-based assessment, the relative distance between individuals in the objective space is adopted. This method is perhaps the most common used in the literature. Some well-known techniques include niche sharing [HoNG94], crowding distance [DPAM02], and clustering [ZiBa11, Padm13, SiMD13]. These methods are examples of distance based density assessment schemes that are not influenced any external parameters of MOEAs. However, these techniques are susceptible to scaling issues and their effectiveness are limited by the presence of non-commensurable objectives [GoTa09].

In distribution-based density assessment, the probability density of the individuals is used to assess the diversity of population. In [LiYu12], the probability density is used to compute the entropy as a mean to quantify the information contributed by each individual to a Pareto front. Different from distance-based assessment, distribution-based assessment is not affected by non-commensurable objectives. However, the tradeoff is the computational complexity and the uncertainty in estimation of information using Parzen window.

3. Update Population

The process of updating population uses an elitist strategy to preserve the best individuals and also to prevent the loss of good individuals in each evolution of MOEAs. The elitism is an important procedure to improve convergence of MOEAs.

2.3.3 Multiobjective Non-dominated Sorting Genetic Algorithm (NSGA-II)

NSGA-II [DPAM02] is a famous example of elitist MOEAs. In NSGA-II, both an elite-preservation strategy and a diversity-preserving mechanism are adopted. NSGA-II algorithm is described in the Algorithm 2.

In NSGA-II, in the evolutionary circle, the parent population $\boldsymbol{P}_{par}$ is selected from the current population $\boldsymbol{P}$ based on the Pareto ranks and crowding distance values. The offspring population $\boldsymbol{P}_{offs}$ of size N is generated from the parent population $\boldsymbol{P}$ by using genetic operators. Next, the two populations are combined together to form the combined population $\boldsymbol{P}_c$ of size $2N$. Then, a non-dominated sorting procedure NDSort() is applied to classify the individuals in the combined population $\boldsymbol{P}_c$ into different fronts. When the sorting finishes, the new combined population $\boldsymbol{P}_c$ consists of sorted solutions of different ascending non-dominated fronts. In each front i of the sorted population, the crowding distance metric is evaluated for each individual for the diversity assessment. The individuals in the front i is then sorted in the descending order of the obtained crowding distance values. The elitism of NSGA-II is performed by the procedure of selection with replacement to select the best population $\boldsymbol{P}$ of size N from the combined population $\boldsymbol{P}_c$ of size $2N$. The non-dominated sorting, crowding distance evaluation, and selection with replacement procedures are described in details in [DPAM02].

Algorithm 2 NSGA-II

```
procedure NSGA-II()
    Select Parent Population P_par ← Tour-Selection(P)
    Generate Random Population P size N
    Objectives Evaluation P ← Eval(P)
    Non-Dominated Sorting P ← NDSort(P)
    Crowding Distance Evaluation P ← CrowDist(P)
    itrs = 0
    repeat
        Generate Offspring Population P_offs size N
        Objectives Evaluation P_offs ← Eval(P_offs)
        Combination P_c = P ∪ P_offs
        Non-Dominated Sorting P_c ← NDSort(P_c)
        Crowding Distance Evaluation P_c ← CrowDist(P_C)
        Selection with Replacement P ← Replacement(P_c)
        itrs = itrs + 1
    until itrs ≥ MaxItrs
    return Non-dominated Population P
end procedure
```

2.4 Multiobjective Memetic Algorithms

The term "meme" was first introduced and defined by Rechard Dawkins as the basic unit of cultural transmission, or imitation [Dawk89]. Inspired by Darwinian's evolutionary theory and Dawkin's theory of memes, the term Memetic Algorithm (MA) was first introduced by Moscato in 1989 [Mosc89]. In this work, Moscato viewed MAs as extensions of EA that adopt the hybridization between EA and an individual learning procedure performing local refinements. The definition of MA is usually based on its implementation features. For example,"Memetic algorithms are population-based metaheuristics composed of an evolutionary framework and a set of local search algorithms which are activated within the generation cycle of the external framework" [NeCo12].

The use of MA for multiobjective optimization (*Mutiobjective optimization based on memetic algorithms* - MOMA) has attracted much attention and effort in recent years. In the literature, MOMAs have been demonstrated to be much more effective and efficient than the EAs and the traditional optimization searches for some specific optimization problem domains [KrSm05, IHTY09, NeCo12, COLT11, BTMA12]. However, the reports on the applications of MOMAs to real engineering problems are still limited in the literature.

The performance of MOMAs not only relies on the evolutionary framework, but also depends on the local search. The best tradeoff between a local search and the global search in each evolution is an important issue of an MOMA [KrSm05]. There are different MOMAs introduced in the literature for domain-specific applications [MLMH10, LSCS10]. Ishibuchi et al. [IHTY09] introduced an MOMA for combinatorial optimization problems. This work adopts a hybridization of the multiobjective genetic algorithm NSGA-II introduced by Deb and coworkers [DPAM02] and a local search to produce an MOMA for the Knapsack combinatorial optimization problem. In this work, a local search is employed to refine the offsprings with a weighted sum-based scheme. The selection criterion are based on Pareto ranking and crowding distance sorting used in NSGA-II. This MOMA is described in the Algorithm 3.

Algorithm 3 is a hybrid between NSGA-II and a local search. The procedures "Fast Non-Dominated Sort", and "Crowding Distance Assignment" are parts of the NSGA-II algorithm described in details in [DPAM02, TaKL10]. The procedure "Generate Offspring Population" is genetic operation procedure consisting of crossover and mutation operations. The offsprings are refined by the local search with probability of p_{ls}. In the

Algorithm 3 Multiobjective Optimization based on Memetic Algorithm (MOMA)

procedure MOMA(N, p_{ls})
    Generate Random Population $\boldsymbol{P}$ size N
    Objectives Evaluation
    Fast Non-Dominated Sort
    Crowding Distance Assignment
    repeat
        Select Parent Population $\boldsymbol{P}_{par} \leftarrow$ Selection($\boldsymbol{P}$)
        Generate Offspring Population $\boldsymbol{P}_{offs}$
        $\boldsymbol{P}_{impr} \leftarrow$Local-Search($\boldsymbol{P}_{offs}, p_{ls}$)
        $\boldsymbol{P}_{inter} \leftarrow \boldsymbol{P} \cup \boldsymbol{P}_{offs} \cup \boldsymbol{P}_{impr}$
        Fast Non-Dominated Sort
        Crowding Distance Assignment
        Update Population $\boldsymbol{P} \leftarrow$ Updating($\boldsymbol{P}_{inter}$)
    until Terminated Conditions
    return Non-Dominated Population $\boldsymbol{P}$
end procedure

local search, a weighted-sum fitness is used as a fitness assignment. [IHTY09]. The k objectives $(f_1, f_2, ...f_k)$ are weighted to be a single objective by

$$f(x) = \sum_{i=1}^{k} \lambda_i f_i(x) \tag{2.7}$$

where $(\lambda_1, \lambda_2, ..., \lambda_k)$ are random normalized weights generated according to [Jasz02]

$$\begin{cases} \lambda_1 = 1 - \sqrt[k-1]{rand()} \\ \dots \\ \lambda_j = (1 - \sum_{l=1}^{j-1} \lambda_l)(1 - \sqrt[k-1-j]{rand()}) \\ \dots \\ \lambda_k = 1 - \sum_{l=1}^{k-1} \lambda_l \end{cases} \tag{2.8}$$

The local search procedure is performed only on the best individuals of a given offspring generation. Firstly, a random weight vector is generated by Eq. (2.8). Based on the generated random weights, the initial solution for local search is selected from offspring population using tournament selection with replacement. The same random weights are then used for the local search to produce improved population $\boldsymbol{P}_{impr}$ from selected initial individual. The intermediate population $\boldsymbol{P}_{inter}$ is produced by combining the current population $\boldsymbol{P}$, the offspring population $\boldsymbol{P}_{offs}$, and the improved population $\boldsymbol{P}_{impr}$. The non-dominated population $\boldsymbol{P}$ is finally updated by the selection with replacement based on the Pareto ranks and crowding distances in the Updating procedure. The generation updating is described in Fig. 2.4. The algorithm finishes when it meets a predefined maximum number of iterations.

2.5 Evaluation Measures

2.5.1 Benchmark Problems

Benchmark problems are test functions used to verify the important characteristics of the algorithm design, and to explore the capabilities and also possible pitfalls of the algorithm for further improvement. In multiobjective optimization, the test functions must have the characteristics that pose sufficient difficulties to impede the search algorithm for

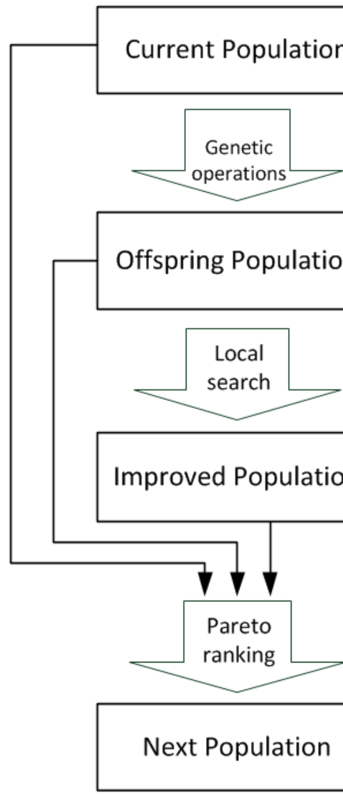


Figure 2.4: Generation update in MOMA's evolutionary circle.

Pareto optimal solutions. Multi-modality referred to the presence of multiple local Pareto fronts is one of the characteristics that hinders convergence in multiobjective optimization [Deb01, GoTa09].

There exists different test problems (test functions) for multiobjective optimization in the literature. In this report, the problems of ZDT1, ZDT2, ZDT3, and DTLZ2 are selected to validate the effectiveness of our proposed multiobjective optimization approaches for convergent speed and the diversity of the Pareto solutions. The Problems ZDT1-3 and DTLZ2 are well-known test problems used widely for the evaluation of MOEAs [DTLZ02, ZiDT00]. These problems are described as follows.

- ZDT1:

$$
\begin{aligned}
f_1(\mathbf{x}) &= x_1, \\
g(\mathbf{x}) &= 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \\
f_2(\mathbf{x}) &= 1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}}
\end{aligned}
\tag{2.9}
$$

where n is the number of variables; $x_i \in [0, 1]$

- ZDT2:

$$
\begin{aligned}
f_1(\mathbf{x}) &= x_1, \\
g(\mathbf{x}) &= 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \\
f_2(\mathbf{x}) &= g(\mathbf{x})\left(1 - \left(\frac{f_1(\mathbf{x})}{g(\mathbf{x})}\right)^2\right)
\end{aligned}
\tag{2.10}
$$

where n is the number of variables; $x_i \in [0, 1]$

- ZDT3:

$$
\begin{aligned}
f_1(\mathbf{x}) &= x_1, \\
g(\mathbf{x}) &= 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \\
f_2(\mathbf{x}) &= g(\mathbf{x})\left(1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}} - \frac{f_1(\mathbf{x})}{g(\mathbf{x})}\sin(10\pi f_1(\mathbf{x}))\right)
\end{aligned}
\tag{2.11}
$$

where n is the number of variables; $x_i \in [0, 1]$

- DTLZ2:

$$\begin{aligned} f_1(\mathbf{x}) &= (1+g(\mathbf{x}_M)).\cos(0.5\pi x_1)...\cos(0.5\pi x_{M-1}) \\ f_2(\mathbf{x}) &= (1+g(\mathbf{x}_M)).\cos(0.5\pi x_1)...\sin(0.5\pi x_{M-1}) \\ &... \\ f_M(\mathbf{x}) &= (1+g(\mathbf{x}_M)).sin(0.5\pi x_1) \\ g(\mathbf{x}_M) &= \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2 \end{aligned} \tag{2.12}$$

where $x_i \in [0,1]$, $i = 1,2,...,n$; M is the number of objectives; n is the number of variables. The last $k = (n - M + 1)$ variables are represented as $\mathbf{x}_M$.

The optimal Pareto fronts of these benchmark test problems are described in Appendix A.

2.5.2 Performance Evaluation Metrics

Different from the single objective optimization, where the performance metric is just to evaluate the one scalar-objective function, in multiobjective optimization the performance metric must assess a number of solutions each having a vector of objectives. Once again, the design of multiobjective optimization is to obtain the optimal population that converges to the optimal Pareto front and maintain the diversity of the set of solutions. Thus, the performance metrics must be able to evaluate the convergence to the optimal Pareto front and the diversity of the obtained optimal solutions.

In this report, the *Inverted Generational Distance* (IGD) [CoCo05, HHBW06, SiMD13] is used as a performance metric. IGD provides a measure for both the convergent proximity and the diversity of the non-dominated population with regards to the true Pareto optimal front. The IGD metric is calculated as follows.

Let $\boldsymbol{P}^*$ be the set of uniformly distributed Pareto optimal solutions in the objective space, and $\boldsymbol{P}$ be the obtained set of non-dominated solutions. The IGD value for the set

$\boldsymbol{P}$ is defined as

$$IGD(\boldsymbol{P}, \boldsymbol{P}^*) = \frac{\sum_{v \in \boldsymbol{P}^*} d(v, \boldsymbol{P})}{|\boldsymbol{P}^*|} \tag{2.13}$$

where $d(v, \boldsymbol{P})$ is the minimum Euclidean distance between $v \in \boldsymbol{P}^*$ and the points in $\boldsymbol{P}$. When the number of elements in $\boldsymbol{P}^*$ is large enough, $IGD(\boldsymbol{P}, \boldsymbol{P}^*)$ could measure both the diversity and the convergence of $\boldsymbol{P}$. The value of $IGD(\boldsymbol{P}, \boldsymbol{P}^*)$ is very small if $\boldsymbol{P}$ closely spread on the true optimal front $\boldsymbol{P}^*$.

2.6 Summary of Chapter 2

This chapter provides a brief background on the multiobjective optimization. The background covers from definitions, theories, algorithms, and implementation, to performance evaluations. The terms defined in this chapter are seen frequently in the next chapters of the thesis. This background chapter is mainly focused on the area of multiobjective optimization to support the next chapter which is about a framework of multiobjective memetic optimization.

However, these backgrounds presented in this chapter do not cover for all methods used for the applications we provide in Chapter 4 and Chapter 5. The backgrounds on methods used for the applications are presented in those chapters for the self-contained presentations.

Chapter 3

Adaptive Multiobjective Memetic Optimization

Multiobjective optimization based on memetic algorithms (MOMA) are recently applied to solve nonlinear optimization with conflicting objectives. An important issue on MOMAs is how to identify the relative best solutions to guide the adaptive processes. Pareto dominance has been used extensively to find the relative relations between solutions for the fitness assessment in multiobjective optimization based on evolutionary algorithms (MOEA). However, the approach based on the Pareto dominance criterion decreases its convergence when the number of objectives increases. The Pareto-dominance based criterion is not sufficient for guiding other adaptive processes in MOMAs. In this chapter, we propose a framework of adaptive multiobjective optimization algorithms (AMMOA) with an effective information-theoretic criterion. The effective information-theoretic criterion is designed based on the multiscale relative Rényi entropy. This criterion is used to guide the adaptive selection, clustering, and local learning processes in our framework of AMMOA. The implementation is applied on several benchmark test problems with remarkable results.

3.1 Introduction

Multiobjective optimization deals with the function of more than two objectives. In most practical decision making problems, there are multiple conflicting objectives or multiple criteria. Unfortulately, these real world problems are often difficult, if not possible, to be solved without advanced and efficient optimization techniques. This is because these problems are characterized by multiple objectives that are much more complex as compared to the single-objective problems. Because of that, a multiobjective optimization problem has been mostly combined and solved as a single-objective optimization problem. However, this method is just looking for one solution instead of a set of optimal solutions.

Evolutionary algorithms (EAs) mimic the nature's evolutionary principles to constitute search and optimization procedures. EAs are metaheuristic search techniques that are different from heuristic search methods (e.g., greedy algorithms) as follows. Heuristic methods are problem dependent, as such they try to take full advantage of the particularities of the problem. The heuristic methods are often too greedy that they are usually trapped in a local optimum. Whereas, metaheuristic methods (e.g., EAs) do not take advantage of any specificity of the problem, and therefore, they can be used as black boxes. They are not greedy, and might even accept a temporary deterioration of the solution, which allows them to explore more thoroughly the search space to search for a better solution.

Evolutionary algorithm is a population based metaheuristic algorithm that uses a population of solutions in each iteration, instead of a single solution in classical methods. The outcome of an EA is also a population of solutions. Thus, an EA can be efficiently used to capture multiple optimal solutions in its final population for multi-objective optimization problems. EAs have the important advantage of being able to sample multiple solutions simultaneously. This feature makes EAs common-used in multiobjective optimization (called multiobjective optimization using EAs - MOEA). Many MOEAs have been proposed in the literature. Most of them are based on the models of *genetic algorithms* (GAs) [Deb01]. Recently, biologically inspired models, such as *partical swarm* (PS), *differential evolution* (DE), and *memetic algorithms* (MAs) have been introduced for multiobjective optimization [LeKi13, WaCa12, IHTY09]. The main difference between these approaches is in the method of generating new candidate solutions.

The term "meme" was first introduced and defined by Rechard Dawkins as the basic unit of cultural transmission, or imitation [Dawk89]. Inspired by Darwinian's evolutionary theory and Dawkin's theory of memes, the term memetic algorithm was first introduced by Moscato in 1989 [Mosc89]. Moscato viewed MAs as extensions of EAs that adopt the hybridization between EA and an individual learning procedure performing local refinements. Different from EAs, the performance of MAs relies on both the global evolutionary search and the local search. The definition of MA is usually based on its implementation features. For example,"Memetic algorithms are population-based metaheuristics composed of an evolutionary framework and a set of local search algorithms which are activated within the generation cycle of the external framework" [NeCo12]. With the integrated local refinements within each evolutionary iteration, MAs are shown to be more superior to GAs in convergent for different optimization problems [KrSm05]. The use of MA for multiobjective optimization (*Mutiobjective optimization based on memetic algorithms* - MOMA) has attracted much attention and effort in recent years. In the literature, MOMAs have been demonstrated to be much more effective and efficient than MOEAs and the traditional optimization searches for some specific optimization problem domains [KrSm05, IHTY09, NeCo12, COLT11, BTMA12, DaKi14a, DaKi14b]. The performance of an MOMA not only relies on the evolutionary framework, but also depends on the local searches. The performance of an MOMA is illustrated through the effectiveness and efficiency evaluations. The effectiveness of an MOMA is evaluated by the convergent speed and the diversity of the obtained optimal Pareto solutions. The efficiency of an MOMA is given by its applicability and complexity. There are different MOMAs introduced in the literature for domain-specific applications [MLMH10, LSCS10].

Several studies have shown that MOEAs scale poorly with regard to increasing the number of objectives [KhYD03, Hugh05, FaPC10, FaPC09]. The main reason is that the principle of Pareto dominance, which is mostly used as a ranking criterion in MOEAs, is less effective when the number of objectives in MOEAs increases. Therefore, only Pareto ranking based MOEAs is not sufficient for solving multiobjective optimization problems. The current MOMAs in the literature use Pareto-dominance ranking as a convergence measure to guide learning processes. On the other hand, diversity preservation is another critical issue of *multiobjective optimization problem* (MOP). The diversity preservation is performed based on diversity assessment criteria (e.g., density, distance, or distribution based criteria). When the number of objectives in an MOP increases, the diversity criterion plays the key role in selecting the solutions. In this context, an effective diversity assessment criterion can help a multiobjective optimization algorithm (e.g., MOEA, MOMA) converges well. Thus, the need for an effective mechanism of diversity preservation is also critical. Besides, in MOMA local searches are adopted for individual learning. Individual refinements can help improve the convergence; however, it can destroy the diversity of the population if we do not have a wise criteria for guiding the search and a well-designed learning mechanism as well [DaKi15a].

In this work, we propose an effective information-theoretic criterion based on the multiscale relative Rényi entropy. This information-theoretic criterion is used to guide the adaptive selection, clustering, and local learning processes in our *adaptive multiobjective memetic optimization algorithms* (AMMOA). The AMMOA framework is proposed based on the observation from the real human cultural evolution that the individuals are gone through a hierarchical social learning structure. They first learn from their small communities to grow to compete with others in their local communities. The best individuals from small communities then contribute to the bigger community. They learn and compete each other to improve the community. The AMMOA framework adopts two layers of local learning with adaptive factors. This framework uses the proposed information-theoretic criterion to guide the adaptive selection, clustering, and local learning processes for improving the convergence and the diversity of the obtained population. The main contribution of this work are as follows.

1. An effective information-theoretic criterion is proposed to guide the adaptive processes such as the selection, clustering, and local learning processes in adaptive multiobjective optimization techniques.

2. A framework of adaptive multiobjective memetic optimization algorithms (AMMOA) based on the proposed information-theoretic criterion is introduced.

3. An implementation of the AMMOA framework with the adaptive tournament selection, fuzzy-clustering, Tabu local searches, and an online stopping criterion, all guided by the proposed information-theoretic criterion, is introduced with remarkable results.

4. An robust online stopping criterion is introduced for AMMOA.

3.2 Information Theoretic Criterion

3.2.1 Relative Probability

In this section, we define the relative probability of each individual in the population for computing the information measures. A relative probability is considered as a dominating probability which shows the probability of an individual dominating the other individuals in a selected pool.

Let recall a simplified multiobjective optimization problem (MOP) from the problem formulation in Eq. (2.1) as

$$\begin{aligned} \text{minimize } \boldsymbol{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), ..., f_M(\mathbf{x})\} \\ \text{subject to } \mathbf{x} \in \mathcal{X} \end{aligned} \tag{3.1}$$

where $\mathbf{x}$ is a decision vector containing decision variables, $\boldsymbol{f}(\mathbf{x})$ is the M-dimensional objective vector ($M \geq 2$), $f_m(\mathbf{x})$ is the m^{th} objective function ($m = 1, 2, ..., M$), and $\mathcal{X}$ is the feasible variable region defined by the user constraints.

Let $\boldsymbol{P}$ is the population including N_P feasible solutions for the problem in Eq. (3.1). Let $\boldsymbol{P'} \subset \boldsymbol{P}$ is the subset of $\boldsymbol{P}$, and $N_{P'} \leq N_P$. We define the relative probability of the m^{th} objective function of the solution $\mathbf{x}_i$ in the sub population $\boldsymbol{P'}$ as

$$p_m(\mathbf{x}_i) = \frac{\sum_{j \# i; \mathbf{x}_j, \mathbf{x}_i \in \boldsymbol{P'}} \mathbb{1}_{(\max(f_m(\mathbf{x}_j) - f_m(\mathbf{x}_i), 0))}}{N_{P'}} \tag{3.2}$$

where

$$\mathbb{1}_{(\max(f_m(\mathbf{x}_j) - f_m(\mathbf{x}_i), 0))} = \begin{cases} 1, & \text{if } \max\left(f_m(\mathbf{x}_j) - f_m(\mathbf{x}_i), 0\right) > 0; \\ 0, & \text{if } \max\left(f_m(\mathbf{x}_j) - f_m(\mathbf{x}_i), 0\right) = 0; \end{cases} \tag{3.3}$$

Lemma 3.1. Let $\mathbf{x}_t$ and $\mathbf{x}_q$ ($t \# q$, $1 \leq t, q \leq N_P$) are the feasible solutions of the population $\boldsymbol{P}$. The solution $\mathbf{x}_t$ is said to dominate the solution $\mathbf{x}_q$ if the relative probabilities of the objective functions satisfy $p_m(\mathbf{x}_t) \geq p_m(\mathbf{x}_q)$, $\forall m = \{1, 2, ..., M\}$, and $\exists n \in \{1, 2, ..., M\} : p_n(\mathbf{x}_t) > p_n(\mathbf{x}_q)$.

Lemma 3.2. solution $\mathbf{x}^* \in \boldsymbol{P}$ is said to be non-dominated solution in the population $\boldsymbol{P}$ if $\forall \mathbf{x} \in \boldsymbol{P}$, $p_m(\mathbf{x}^*) \geq p_m(\mathbf{x})$, $\forall m = \{1, 2, ..., M\}$, and $\exists n \in \{1, 2, ..., M\} : p_n(\mathbf{x}^*) > p_n(\mathbf{x})$.

3.2.2 An Information Theoretic Criterion

From Lemma 3.1 and Lemma 3.2, it can be inferred that an optimal solution is the solution that maximizes the relative probabilities as defined in Eq. (3.2). It can also state that the solution that maximizes the relative probabilities is the solution minimizes the distance between the vector of M relative probabilities, $\langle p_1(\mathbf{x}), p_2(\mathbf{x}), ...p_M(\mathbf{x})\rangle$ and the reference ideally optimal vector, $\langle 1_1, 1_2, ..., 1_M\rangle$. There are different distance measures that can be used to measure a distance between two vectors. They can be 1-norm distance, 2-norm distance, p-norm distance, and information distance. While the p-norm distances are usually used for measuring distances between two absolute real vectors, the information distances are used for measuring the distances between two distributions or probabilistic vectors.

Let us investigate a p-norm distance in this context. First, let's redefine the relative probability vector of the solution $\mathbf{x}$, and the reference vector as $\boldsymbol{p}(\mathbf{x}) = \langle p_1(\mathbf{x}), p_2(\mathbf{x}), ...p_M(\mathbf{x})\rangle$, and $\boldsymbol{r} = \langle r_1, r_2, ..., r_M\rangle = \langle 1_1, 1_2, ..., 1_M\rangle$, respectively. The p-norm distance between two vector $\boldsymbol{p}$ and $\boldsymbol{r}$ is given by

$$\begin{aligned} f^p(\mathbf{x}) = f^p(\boldsymbol{p}, \boldsymbol{r}) &= \left(\sum_{i=1}^{M} |r_i - p_i(\mathbf{x})|^p\right)^{\frac{1}{p}} \\ &= \left(\sum_{i=1}^{M} |1 - p_i(\mathbf{x})|^p\right)^{\frac{1}{p}} \end{aligned} \tag{3.4}$$

where p is a real number.

According to this p-norm distance, the solution $\mathbf{x}_1$ dominates the solution $\mathbf{x}_2$ if $f^p(\mathbf{x}_1) < f^p(\mathbf{x}_2)$. If the value $p = 1$ we have the 1-norm distance, and 2-norm distance if $p = 2$, and so on. With this distance criterion, the best solution we want to get is the one having all relative probabilities as close as possible to the hypothetical reference vector $\boldsymbol{r}$. However, in practice there exists tradeoffs between the M objective functions. That means there are tradeoffs between the relative probabilities in the relative probability vector. Thus, the p-norm distances are not sufficient for characterizing the distance between a solution to a reference as a criterion for guiding other procedures in an MOMA.

In this work, we adopt the generalized Rényi divergent or Rényi relative entropy as an information distance measure. The Rényi divergent or Rényi relative entropy between the distribution vector $\boldsymbol{r}$ and $\boldsymbol{p}(\mathbf{x})$ is defined as [KiDa06, Reny70]

$$\begin{aligned} f^{Rq}(\mathbf{x}) = H_{Rq}(\boldsymbol{r}||\boldsymbol{p}(\mathbf{x})) &= \frac{1}{q-1}\log\frac{\sum_{j=1}^{M} r_j\left(\frac{r_j}{p_j(\mathbf{x})}\right)^q}{\sum_{j=1}^{M} r_j} \\ &= \frac{1}{q-1}\log\frac{\sum_{j=1}^{M} 1\left(\frac{1}{p_j(\mathbf{x})}\right)^q}{M} \\ &= \frac{1}{q-1}\log\frac{\sum_{j=1}^{M} (p_j(\mathbf{x}))^{-q}}{M} \end{aligned} \tag{3.5}$$

where $0 \leq q \leq \infty$.

The Rényi relative entropy, $f^{Rq}(\mathbf{x})$ in Eq. (3.5) has many interesting and good properties for an effective distance measure. These properties are described in the following lemmas.

Lemma 3.3. The distance measure between $\boldsymbol{r}$ and $\boldsymbol{p}(\mathbf{x})$, $f^{Rq}(\mathbf{x})$, is a monotonic non-increasing function in $q \# 0, 1$.

Lemma 3.4. The measure $f^{Rq}(\mathbf{x}) = 0$ when $q = 0$.

Lemma 3.5. If the distribution $\boldsymbol{p}(\mathbf{x})$ has reached the reference vector $\boldsymbol{r}$, then the distance $f^{Rq}(\mathbf{x}) = 0$.

Lemma 3.6. For $q \leftarrow 1$, $f^{Rq}(\mathbf{x})$ becomes Kullback-Leibler divergence:

$$\begin{aligned} \underset{q\leftarrow 1}{f^{Rq}(\mathbf{x})} &= \sum_{j=1}^{M} r_j \log\frac{r_j}{p_j(\mathbf{x})} \\ &= \sum_{j=1}^{M} -\log p_j(\mathbf{x}) \end{aligned} \tag{3.6}$$

The Rényi relative entropy spectrum is a multiscale measure that is good for tuning and comparing two distributions. However, it is too complex to be a tuning metric for a simulation run in multiobjective optimization algorithms. To make it applicable, we use a scalar metric, named Rényi's relative summation entropy (R_{RSE}), defined as

$$R_{RSE}(\boldsymbol{r}||\boldsymbol{p}(\mathbf{x})) = \sum_{q=1}^{20} f^{Rq}(\mathbf{x}) \tag{3.7}$$

In Eq. 3.7, when $q = 1$ (it is the case of $q \to 1$), $f^{Rq}(\mathbf{x})$ becomes Kullback-Leibler divergence as provided in Lemma 3.6.

Lemma 3.7. Let $R_{RSE}(\mathbf{x}_1)$ and $R_{RSE}(\mathbf{x}_2)$ are the R_{RSE} of the solution $\mathbf{x}_1$, and the R_{RSE} of the solution $\mathbf{x}_2$ in the feasible population $\mathcal{P}$, respectively. If $R_{RSE}(\mathbf{x}_1) < R_{RSE}(\mathbf{x}_2)$, then solution $\mathbf{x}_1$ dominates the solution $\mathbf{x}_2$.

The proofs of Lemmas 3.1–3.7 are described in Appendix B. The metric R_{RSE} is used as a criterion for guiding our adaptive tournament selection, clustering, and local searches in our AMMOA framework.

3.3 AMMOA Framework

In this section, a framework of adaptive multiobjective memetic optimization algorithms (AMMOA) is presented. The skeleton of the framework is graphically described in Fig. 3.1.

The framework consists of eight following modules.

(*i*) *Initialize Population*: This module is to create the initial population $\boldsymbol{P}_{init}$. The initial population can be produced by a random number generator (e.g., pseudo random generation, Monte Carlo methods, Fibonacci sequences) or chaotic sequences generated from chaotic maps (e.g., logistic map, Gauss map, Hẽnon map).

(*ii*) *Selection*: In this module, the parent population is selected from the current population to form an intermediate mating pool for generating an offspring population

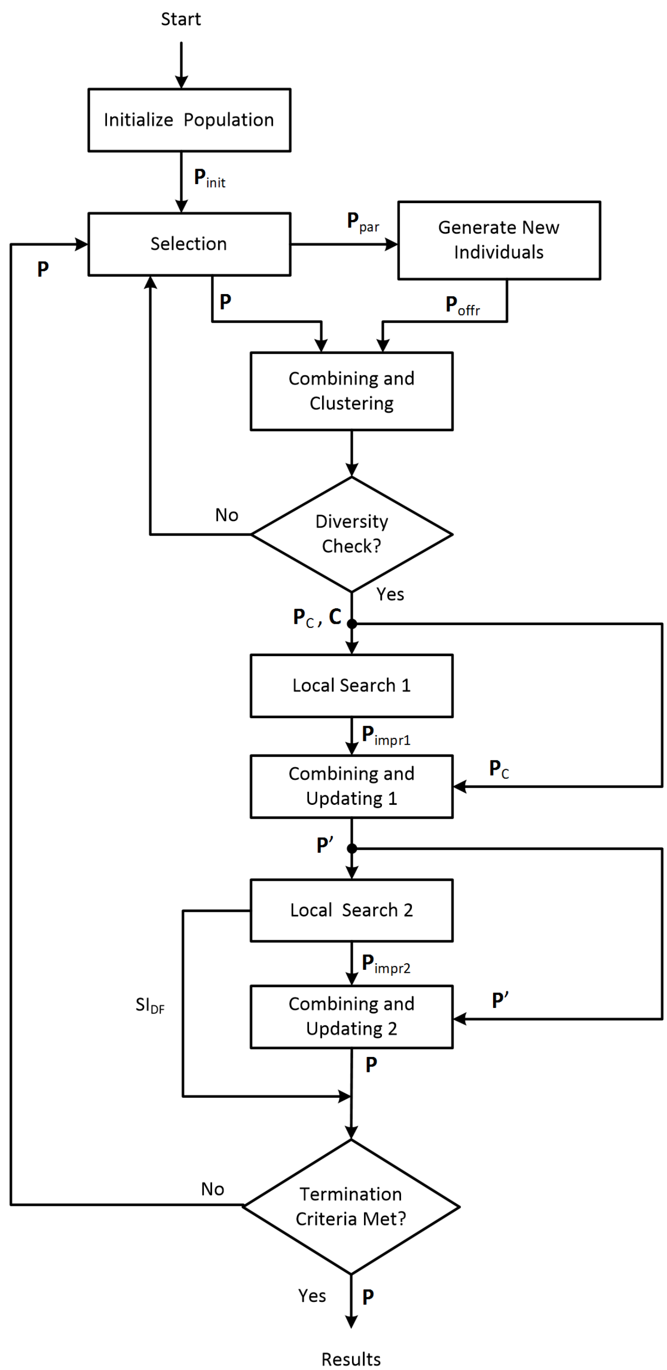


Figure 3.1: The AMMOA framework.

in the next procedure. The selection operator is based on the proposed R_{RSE} criterion in Eq. (3.7). There are several selection operators proposed in the literature, such as proportional selection, ranking selection, and tournament selection [Gold89, Deb01]. The brief introductions of these selection methods are described in Chapter 2, Sec. 2.3. In this work, the tournament selection based on the R_{RSE} criterion is used for the selection of parent population in the AMMOA framework. The tournament selection with the R_{RSE} criterion is described in Algorithm 4.

Algorithm 4 Selection Operator Using R_{RSE}

```
procedure SELECTION(P, N_tour, N_par, p_t)
    Calculate the metric R_RSE for each individual in the solution P.
    repeat
        Pick N_tour (tournament size) individuals from the population P at random.
        Select the best individual having the least R_RSE value from the tournament
            with the probability p_t.
        Select the second best individual having the second least R_RSE from the
            tournament with the probability p_t(1 − p_t).
        Select the third best individual having the third least R_RSE from the
            tournament with the probability p_t(1 − p_t)^2, and so on.
    until Terminated Conditions (Have N_par best individuals)
    return N_par parent individuals
end procedure
```

In our tournament selection algorithm, to increase the population diversity, not only the best solution is picked at each generation but also solutions that are a bit worse is able to contribute to the evolution process. If the probability $p_t = 1$, only the best individual is picked to be the parent in each tournament. Thus, the probability p_t can be adjusted to increase the diversity of the parent population to produce a diverse offspring population.

(*iii*) *Generate New Individuals*: This is an important module of the framework. It produces the new offspring population $\boldsymbol{P}_{offr}$ by mating or learning from the parent population $\boldsymbol{P}_{par}$. In evolutionary framework, this module consists of crossover and mutation operators. In this work, we use the simulated binary crossover (SBX) and the polynomial mutation to implement this module. These operators are described in Chapter 2, Sec. 2.3.

(*iv*) *Combining and Clustering*: In this module, the parent population $\boldsymbol{P}_{par}$ and the new offspring population $\boldsymbol{P}_{offr}$ are firstly combined into an intermediate population $\boldsymbol{P}_C$. Then the metric R_{RSE} of each individual in the new intermediate population $\boldsymbol{P}_C$ is calculated. A clustering algorithm is then applied to cluster the population $\boldsymbol{P}_C$ into C clusters. The R_{RSE} measure is used as the feature for the clustering algorithm. The diversity measure is also computed in this step to evaluate the diversity of the intermediate population $\boldsymbol{P}_C$. The main processing steps of this module are described in Algorithm 5.

It can be seen that the created diversity of the combined population $\boldsymbol{P}_C$ after performing genetic operators (e.g., crossover and mutation) is not destroyed by the clustering algorithm. The diversity of the population $\boldsymbol{P}_C$ is maintained to contribute to the "Combining and Updating 1" module. The clustering algorithm is only used for checking the diversity of the population $\boldsymbol{P}_C$ and helping Local Search 1 search for a diverse and better neighborhood population, if the population $\boldsymbol{P}_C$ is checked to be well-diverse.

In this module, any fast clustering methods can be used to cluster the intermediate population $\boldsymbol{P}_C$ into N_c clusters. The proposed metric R_{RSE} is used as the feature for clustering and computing the clustering quality index I_Q. The clustering quality

Algorithm 5 Combining and Clustering (COM_CLUS)

1: **procedure** COM_CLUS($\boldsymbol{P}_{par}, \boldsymbol{P}_{offr}, N_c$)
2: Combine the parent population and the offspring population into the intermediate population: $\boldsymbol{P}_C \leftarrow \boldsymbol{P}_{par} \cup \boldsymbol{P}_{offr}$.
3: Calculate the metric R_{RSE} for each individual in the population $\boldsymbol{P}_C$.
4: Cluster the population $\boldsymbol{P}_C$ into N_c clusters denoted by CL_i, $i = 1, 2, ..., N_c$, based on the metric R_{RSE}.
5: Calculate the clustering quality index I_Q in Eq. (3.10).
6: **return** The population $\boldsymbol{P}_C$, clusters CL_i, $i = 1, 2, ..., N_c$, the quality index I_Q.
7: **end procedure**

index Q is computed based on both the homogeneity and separation of clusters. While homogeneity is calculated as the average distance between each individual and the centroid of the cluster it belongs to, separation is calculated as the weighted average distance between cluster centroids. The expressions of homogeneity index and separation index are given in Eq. (3.8), and (3.9), respectively.

$$I_H = \sum_{i=1}^{N_c} \frac{1}{|CL_i|} \sum_{S_j \in CL_i} D(S_j, c_i) \tag{3.8}$$

$$I_S = \frac{1}{N_c^2} \sum_{i \# j}^{N_c} D(c_i, c_j) \tag{3.9}$$

where the distance D is Euclidean distance, c_i is the centroid of the cluster CL_i, S_j is the individual j of the cluster CL_i, $|CL_i|$ is the number of individuals in the cluster CL_i.

It can be seen that the index I_H reflects the compactness of the clusters, and the index I_S reflects the overall distance between clusters. Thus, increasing I_H or increasing I_S suggest an improvement of the diversity of the population. Therefore we propose a clustering quality index I_Q used to evaluate the diversity is the average of the homogeneity index and the separation index as given by

$$I_Q = I_H + I_S \tag{3.10}$$

If the index I_Q has a small value, the individuals in the population $\boldsymbol{P}_C$ are too close to each other. The population $\boldsymbol{P}_C$ is said to be diverse if the index I_Q has a large value or be greater than a bound value I_{Q0}.

(v) *Diversity Check*: The clustering quality index I_Q is used to check the diversity of the population $\boldsymbol{P}_C$. The index I_Q of each evolution is stored in a memory. Denote I_Q^t is the clustering quality index at the evolution t, the bound of the index I_Q^t at the evolution t is given by

$$I_{Q0}^t = \frac{1}{t_0} \sum_{i=t-t_0}^{t-1} \frac{1}{2(t-i)} I_Q^i \tag{3.11}$$

The diversity check compares the current clustering quality index I_Q^t with its bound I_{Q0}^t. If $I_Q^t > I_{Q0}^t$, the population maintains the diversity well. To make sure the system has a right decision, we desire to check the diversities of two sequential evolutions before making the final decision as follows. The population is said to be not diverse if in two sequential evolution t and $t+1$, their clustering quality indexes are smaller than their bounds I_{Q0}^t and I_{Q0}^{t+1}, respectively.

When the module decides the population $\boldsymbol{P}_C$ is not diverse at the current evolution, it sends a control signal to reduce the picking probability p_t at the "Selection" process to increase the diversity of the parent population $\boldsymbol{P}_{par}$. Each time of controlling, the adjustment step is set to 0.05.

When the population $\boldsymbol{P}_C$ is evaluated to be diverse enough, Local Search 1 is applied to N_c clusters CL_i, $i = 1, 2, ..., N_c$ with the probability of p_{local1}, to refine individuals in each cluster. If the clusters are too compact and close to each other, the local search produces overlap improved individuals. This reduces the diversity of the population very much. Thus, maintaining the diversity of the population before any local refinement is very important in each evolution of the framework.

(*vi*) *Local Search 1*: This module produces improved individuals for each cluster CL_i. The main steps involved in Local Search 1 are shown in Algorithm 6.

Algorithm 6 Local Search 1

1: **procedure** LOCAL_SEARCH1($\boldsymbol{CL}, p_{local1}$)
2: For each cluster CL_i, with the size N_{CL_i}, $i = 1, 2, ..., N_c$, select the best individual having the least R_{RSE} value.
3: For each cluster CL_i, apply a local search algorithm to the selected individual with the probability P_{local1} to find the N_{CL_i} refined individuals.
4: Combine all the refined individuals into the improved population P_{impr1}.
5: **return** The improved population $\boldsymbol{P}_{impr1}$.
6: **end procedure**

Local Search 1 is only applied when the population $\boldsymbol{P}_C$ is declared to have a good diversity in the "Combining and Clustering" module. It can be seen that all the clusters CL_i, $i = 1, ..., N_c$ are locally refined with the same probability p_{local1}. In each cluster, only the best individual having the least R_{RSE} value is subjected to a local search in any generation of the local search algorithm. This best individual is also called the initial individual for the local search in each cluster. Local Search 1 tries to find N_{CL_i} better individuals for each cluster CL_i after a predefined number of local iterations. In each local simulation run, the objectives of each individual is combined based on the random weights defined in Eq. (2.8) for evaluation and comparison.

(*vii*) *Combining and Updating 1*: This is the first elitism module in our framework. First, the intermediate population $\boldsymbol{P}_C$ and the improved population $\boldsymbol{P}_{impr1}$ are combined to become the population $\boldsymbol{P}_{I1}$. The elitism is then applied to the combined population $\boldsymbol{P}_{I1}$ to preserve the N_p best individuals. These main steps involved in this module are shown in Algorithm 7.

Algorithm 7 Combining and Updating 1

procedure COMBINE_UPDATE1($\boldsymbol{P}_{impr1}, \boldsymbol{P}_C$)
    Combine the two populations: $\boldsymbol{P}_{I1} \leftarrow \boldsymbol{P}_C \cup \boldsymbol{P}_{impr1}$.
    Apply an elitist strategy to the population $\boldsymbol{P}_{I1}$ to preserve the N_p best individuals. These N_p best individual form the population $\boldsymbol{P'}$.
    return The population $\boldsymbol{P'}$ with N_p individuals.
end procedure

The process of updating uses an elitist strategy to preserve the N_p best individuals. The elitism is an important procedure in this module. It should both improve the convergence greatly and preserve the diversity of the population.

(*viii*) *Local Search 2*: This second local search is applied to refine the population $\boldsymbol{P'}$. Different from Local Search 1 that is applied to each cluster for local refinement, in Local Search 2, the search is applied to the whole population for a global refinement. While Local Search 1 is both for improving diversity and convergence, Local Search 2 is mainly for speeding up the convergence. The main steps in Local Search 2 are shown in Algorithm 8.

It can be seen that Algorithm 8 has two functionalities including population refinement and the calculation of stopping indicator values. For each individual $\mathbf{x}$ in the population $\boldsymbol{P'}$, the algorithm generates N_{nb} neighbors surrounding $\mathbf{x}$ within a radius. These generated neighborhood individuals are then evaluated to calculate the

Algorithm 8 Local Search 2

1: **procedure** LOCAL_SEARCH2($\boldsymbol{P'}$)
2: For each individual in the population $\boldsymbol{P'}$, generate N_{nb} neighborhood individuals.
3: Evaluate the objective functions and the information measure R_{RSE} of the generated neighborhood individuals.
4: Calculate the stopping indicator function based on dominance feature, SI_{DF}.
5: Update the best dominating individuals to form the improved population P_{impr2}.
6: **return** The improved population $\boldsymbol{P}_{impr2}$, and the stopping indicator value SI_{DF}.
7: **end procedure**

information theoretic measure R_{RSE}. Based on the information criterion R_{RSE}, the algorithm can evaluate the dominating quality of each individuals. The stopping indicator feature is then calculated based on the quality and quantity of dominating individuals which dominates the solution $\mathbf{x}$. The articulation and procedures of designing this stopping criterion are provided in the section of "Termination Criteria".

After Local Search 2, the improved population $\boldsymbol{P}_{impr2}$ is competed with the population $\boldsymbol{P'}$ to select the N_p best individuals. This updating process is implemented in the "Combining and Updating 2" module.

(*ix*) *Combining and Updating 2*: In this module, the individuals of the population $\boldsymbol{P}_{impr2}$ and the population $\boldsymbol{P'}$ are competed each other to preserve the N_p best individuals. To do that, the population $\boldsymbol{P'}$ and the population $\boldsymbol{P}_{impr2}$ are combined into the population $\boldsymbol{P}_{I2}$. An elitist strategy is then applied to the population $\boldsymbol{P}_{I2}$ to preserve the N_p best individuals forming the population $\boldsymbol{P}$. These main steps are shown in Algorithm 9.

(*x*) *Termination Criteria*: Most evolutionary algorithms are terminated after a pre-

Algorithm 9 Combining and Updating 2

1: **procedure** COMBINE_UPDATE2($\boldsymbol{P}_{impr2}, \boldsymbol{P'}$)

2: Combine the two populations: $\boldsymbol{P}_{I2} \leftarrow \boldsymbol{P'} \cup \boldsymbol{P}_{impr2}$.

3: Apply an elitist strategy to the population $\boldsymbol{P}_{I2}$ to preserve the N_p best individuals. These N_p best individuals form the population $\boldsymbol{P}$.

4: **return** The population $\boldsymbol{P}$ with N_p individuals.

5: **end procedure**

defined number of generations. In our framework, the algorithm can be stopped according to a predefined number of iterations, or based on an online stopping criterion. Specifically, we proposed an online stopping criterion calculated in Local Search 2, which based on the dominance quality of the generated neighborhood population. The dominance quality is calculated based on the information theoretic criterion R_{RSE}.

The idea of using a dominance quality as an online criterion to evaluate the performance of a multiobjective evolutionary algorithm has recently studied in [GMBG10, TWNP09, BWBA09]. The motivation is that if a solution $\mathbf{x}^*$ is an optimal solution in the Pareto optimal set, there are no neighborhood solutions that dominate $\mathbf{x}^*$. Ideally, the number of generated neighborhood individuals that dominate a solution $\mathbf{x}$ reduces in each iteration until 0 (optimal). This articulation is showed in Fig. 3.2.

Based on the observation in Fig. 3.2, we define the dominance quality indicator of each neighborhood area (Ω) of the considering solution $\mathbf{x}$ is the number of individuals dominating the solution $\mathbf{x}$ over the number of generated individuals in Ω, calculated from

$$dq_{\Omega} = \frac{\sum_{\mathbf{y}\in\Omega} \mathbb{1}(\mathbf{x}, \mathbf{y})}{N_{nb}} \tag{3.12}$$

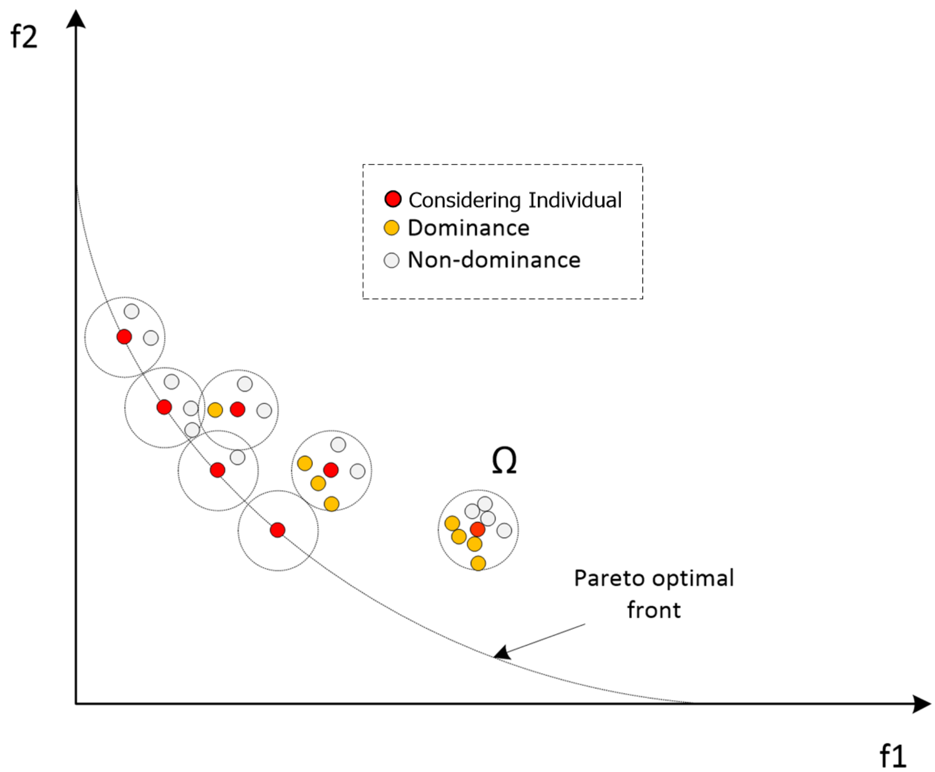


Figure 3.2: Neighborhood demonstration in the objective space.

where

$$\mathbb{1}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1, & \text{if } R^{\mathbf{y}}_{RSE} < R^{\mathbf{x}}_{RSE}; \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

The dominance quality of the whole population is the average of the dominance quality indicators of neighborhood areas, which is used as a stopping indicator calculated from

$$SI_{dq} = \frac{\sum_{i=0}^{N_{nb}} dq_{\Omega}(i)}{N_{pop}} \tag{3.14}$$

The indicator values SI_{dq} calculated in each evolutionary iteration from the local search 2 is used as feature to predict the convergence of AMMOA. Specifically, by analyzing SI_{dq}, we can formulate a robust stopping criterion for AMMOA. The procedure is simply described as follows. After a number of iterations (e.g., 50 iterations), SI_{dq}, calculated from Local Search 2, is analyzed by using a time window of size 30. The new time window is setup to overlap with the previous window in 10 elements. In each window, the mean μ and the standard deviation σ are calculated. The mean and the standard deviation are then compared with a predefined stopping threshold ϵ. If $\mu \leq \epsilon$ and $\sigma \leq \epsilon$, the algorithm indicates that AMMOA converges to the Pareto optimal front and stops. Ideally, the threshold is set to 0; however, in practice the threshold is a small number (e.g., $\epsilon = 10^{-3}$) and can be varied with different multiobjective optimization problems. The procedure of determining the stopping criterion based on the calculated indicator values SI_{dq} is described in Fig. 3.3.

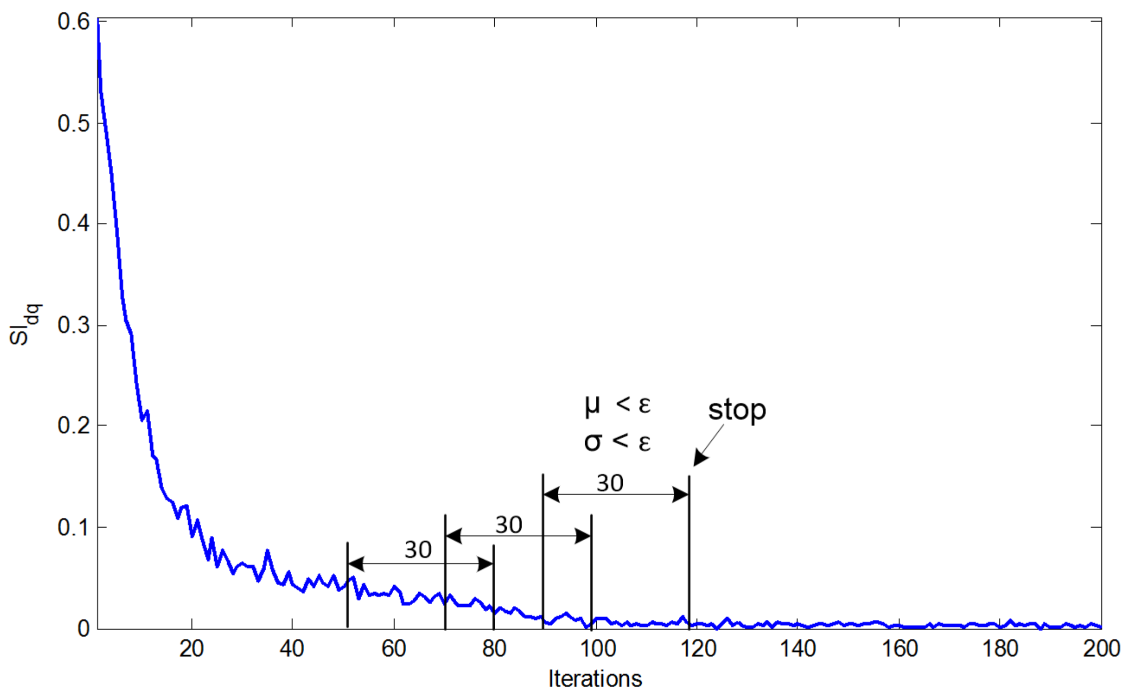


Figure 3.3: Concept of calculating the stopping criterion.

3.4 An Implementation of The AMMOA Framework

In this section, we present an implementation of the proposed AMMOA framework. The new components of the implementation are as follows: (i) the function "generate offspring population" is implemented by simulated binary crossover (SBX) and polynomial mutation; (ii) the clustering is implemented by fuzzy C-mean clustering; (iii) the local searches are implemented based on the principles of Tabu local search; (iv) The updating procedure adopts the elitist strategy of NSGA-2 with non-dominated sorting based on Pareto ranks and crowding distance. The main implementation is described in Algorithm 10.

1) *Generating the offspring population*: Algorithm 10 strictly follows the proposed AMMOA framework. The offspring population is generated by the simulated binary crossover (SBX) with the crossover probability of $p_x = 0.8$, and the polynomial mutation with the mutation probability $p_m = 1/N_p$. The details of these two genetic operators are described in Chapter 2, Sec. 2.3.

2) *Clustering*: In AMOMA, we propose to use *fuzzy c-mean* (FCM) clustering method to cluster the combined population $\boldsymbol{P}_C$ into N_c clusters $\boldsymbol{CL}_i$, $i = 1, 2, ..., N_c$ based on the relative metric $\boldsymbol{R}_{RSE}^{\boldsymbol{P}_C}$.

FCM clustering algorithm was introduced by Dunn [Dunn74] and later generalized by Bezdek [Bezd81]. FCM is an iterative clustering method that produces an optimal N_c partitions by minimizing the following error objective function.

$$J_m = \sum_{i=1}^{N} \sum_{k=1}^{N_c} u_{ki}^m d^2(x_i, v_k) \tag{3.15}$$

where x_i is the i^{th} data item of the data set $\boldsymbol{X} = \{x_i\}_{i=1}^N$, N is the number of data items, $\{v_k\}_{k=1}^L$ are the centers of the clusters; m is fuzzy exponent; u_{ki} is the membership function of x_i in the k^{th} cluster, $d^2(x_i, v_k)$ is a distance measure between object x_i and cluster center v_k. The objective function can get the minimum by updating u_{ki} and v_k

Algorithm 10 Adaptive Multiobjective Memetic Algorithm (AMMOA)

1: **procedure** AMOMA(N_p)
2: Generate random population $\boldsymbol{P}$ having (N_p) individuals
3: Evaluate objectives: $\boldsymbol{P} \leftarrow$ OBJ-EVAL($\boldsymbol{P}$)
4: Calculate the metric vector: $\boldsymbol{R}_{RSE}^{\boldsymbol{P}} \leftarrow$ RRSE($\boldsymbol{P}$)
5: **repeat**
6: Select the parent population: $\boldsymbol{P}_{par} \leftarrow$ SELECTION($\boldsymbol{P}, \boldsymbol{R}_{RSE}^{\boldsymbol{P}}, p_{sel}$)
7: Generate the offspring population $\boldsymbol{P}_{offs}$ by simulated binary crossover (SBX) with probability p_x, and polynomial mutation with probability p_m
8: Combine populations: $\boldsymbol{P}_C \leftarrow \boldsymbol{P} \cup \boldsymbol{P}_{offs}$
9: Calculate the metric vector: $\boldsymbol{R}_{RSE}^{\boldsymbol{P}_C} \leftarrow$ RRSE($\boldsymbol{P}_C$)
10: Fuzzy C-mean clustering: $\boldsymbol{CL} \leftarrow$ FCM-CLUSTER($\boldsymbol{P}_C, \boldsymbol{R}_{RSE}^{\boldsymbol{P}_C}, N_c$)
11: Check the diversity: if $\boldsymbol{P}_C$ is diverse, go to step 12; otherwise, reduce the selection probability $p_{sel} \leftarrow p_{sel} - 0.05$ and go back to step 6.
12: Local search 1: $\boldsymbol{P}_{impr1} \leftarrow$LOCAL-SEARCH-1($\boldsymbol{CL}, \boldsymbol{R}_{RSE}^{\boldsymbol{P}_C}, p_{ls1}$)
13: Combine population: $\boldsymbol{P}_{I1} \leftarrow \boldsymbol{P}_C \cup \boldsymbol{P}_{impr1}$
14: Fast Non-Dominated Sort
15: Crowding Distance Assignment
16: Update Population: $\boldsymbol{P'} \leftarrow$ UPDATING($\boldsymbol{P}_{I1}$)
17: Local Search 2: $\boldsymbol{P}_{impr2}, SI_{dq} \leftarrow$LOCAL-SEARCH-2($\boldsymbol{P'}$)
18: Combine population: $\boldsymbol{P}_{I2} \leftarrow \boldsymbol{P'} \cup \boldsymbol{P}_{impr2}$
19: Fast Non-Dominated Sort
20: Crowding Distance Assignment
21: Update Population: $\boldsymbol{P} \leftarrow$ UPDATING($\boldsymbol{P}_{I2}$)
22: Evaluate Stopping Criterion: $stop\_flag \leftarrow$ STOP-EVAL($\boldsymbol{SI}_{dq}$)
23: **until** Terminated Conditions: $stop\_flag = 1$ or reaching the max iteration
24: **return** Non-Dominated Population $\boldsymbol{P}$
25: **end procedure**

as follows.

$$u_{ki} = \frac{1}{\sum_{j=1}^{N_c} \left(\frac{d^2(x_i,v_k)}{d^2(x_i,v_j)} \right)^{\frac{1}{m-1}}} \tag{3.16}$$

$$v_k = \frac{\sum_{i=1}^{N} u_{ki}^m x_i}{\sum_{i=1}^{N} u_{ki}^m} \tag{3.17}$$

A defuzzification process takes place when the algorithm converges (ie., $max|V^t - V^{t+1}| < \epsilon$, where $V = [v1, v2, ...vl]$ are the vector of the cluster's centroids) in order to convert the fuzzy partition matrix U to a crisp partition. This procedure assigns the data item i^{th} to the class CL_k with the highest membership $CL_k = arg\{max\{u_{ki}\}\}$, $k = 1, 2, ..., N_c$.

The FCM based clustering algorithm used in AMMOA is described in Algorithm 11.

3) *Local Searches*: Various local searches used in optimization have a tendency to become stuck in suboptimal regions where many solutions are equally fit. In AMMOA, we use a modified Tabu search for the local searches (Local Search 1 and Local Search 2) because it enhances the performance of local searches by introducing the rule of prohibitions to discourage the search from coming back to previous-visited solutions. In addition, at each step, worsening moves can be accepted if no improving move is available. These rules are implemented by maintaining a Tabu list.

Local Search 1 is described in Algorithm 12. It can be seen in Algorithm 12 that the Tabu search is applied to every cluster with the probability p_{ls1}. In each cluster the individual having the least R_{RSE} value is selected as the initial solution for the Tabu local search.

Local Search 2 is very different from the local search 1 even though it uses the principle of Tabu list to update the improved individuals. The details of Local Search 2 are described in Algorithm 13.

4) *Elitist strategy (Updating)*: In AMMOA, the procedure "Non-dominated Sort" and "Crowding Distance Assignment" are the components of the elitist strategy proposed in NSGA-2 [DPAM02]. This elitism is introduced shortly in Chapter 2, Sec. 2.3. The

---

Algorithm 11 FCM Based Clustering

1: **procedure** FCM-CLUSTER($\boldsymbol{P}_C, \boldsymbol{R}_{RSE}^{P_C}, N_c, \epsilon$)

2: $N \leftarrow \text{size}(\boldsymbol{P}_C, 1)$

3: Initialize the fuzzy partition matrix $U^{(0)}$, the fuzzy exponent $m \leftarrow 2$

4: Set the loop counter $l \leftarrow 0$

5: Calculate the N_c cluster centers $v_k^{(l)}$ with partition matrix $U^{(l)}$ by

$$v_k^{(l)} = \frac{\sum_{i=1}^{N} u_{ki}^m R_{RSE}^{P_C}(i)}{\sum_{i=1}^{N} u_{ki}^m} \tag{3.18}$$

6: Calculate the Euclidean error distances by

$$d^2(R_{RSE}^{P_C}(i), v_k(l)) = ||R_{RSE}^{P_C}(i) - v_k(l)||^2 \tag{3.19}$$

7: Update the membership matrix $U^{(l+1)}$ by

$$u_{ki}^{(l+1)} = \frac{1}{\sum_{j=1}^{N_c} \left(\frac{d^2\left(R_{RSE}^{P_C}(i), v_k\right)}{d^2\left(R_{RSE}^{P_C}(i), v_j\right)} \right)^{\frac{1}{m-1}}} \tag{3.20}$$

8: If $\max\{U^{(l)} - U^{(l+1)}\} < \epsilon$ then stop, otherwise, set $l \leftarrow l + 1$ and go to step 5.

9: Defuzifize and assign individuals of $\boldsymbol{P}_C$ in to N_c clusters $\{\boldsymbol{CL}_i\}_{i=1}^{N_c}$

10: **return** The N_c clusters $\{\boldsymbol{CL}_i\}_{i=1}^{N_c}$

11: **end procedure**

Updating procedure can be summarized as follows. The non-dominated sorting uses the Pareto ranks and crowding distance to sort the population. First each individual in the population is evaluated the rank and sorted ascendingly based on their Pareto ranks. The crowding distance are then evaluated and assigned for each individual. The individuals in each rank are sorted descendingly based on their crowding distance values. The N_p best individuals are selected based on the smallest ranks first, and follow the max crowding distance values.

Algorithm 12 Tabu LocaL Search 1

```
procedure LOCAL-SEARCH-1(CL, p_ls1)
    for k ← 1, N_c do                                          ▷ Do Tabu search for each cluster
        POP ← CL{k}; N_LS ← size(POP, 1);
        TL ← ∅                                                  ▷ Initialize the Tabu List
        if rand() ≥ p_ls1 then
            rnd_weight ← Lambda(M)                              ▷ Random weights for M objectives
            InitSolution ← Least R_RSE(POP)                     ▷ Select initial solution for the search
            x ← InitSolution ;
            fx ← rnd_weight. ∗ InitObjs                         ▷ InitObjs: objectives of InitSolution
            itrs ← 0; BestObj ← fx
            repeat
                itrs ← itrs + 1
                x_ns ← Neighbor-Generate(x, N_LS)               ▷ Generate N_LS neighbors
                for i ← 1, N_LS do
                    x_nsObjs ← OBJ-EVAL(x_ns(i, :))             ▷ Objective evaluation
                    f_x_ns(i) ← x_nsObjs. ∗ rnd_weight
                    if f_x_ns(i) < BestObj and x_ns(i, :) ∉ TL then
                        xnew ← x_ns(i, :); fnew ← f_x_ns(i);
                    else
                        xnew ← x; fnew ← BestObj;
                    end if
                    Update the Tabu list TL
                end for
                x ← xnew; BestObj ← fnew;
            until itrs ≥ MaxItrs
            TL ← Sort(TL, rnd_weight)                           ▷ Sorting the Tabu list
        end if
        P_temp{k} ← TL(1 : N_LS, :)
    end for
    P_impr1 ← P_temp{1} ∪ P_temp{2} ∪ ... ∪ P_temp{N_c}         ▷ Combining populations
    return P_impr1
end procedure
```

Algorithm 13 Tabu LocaL Search 2

procedure LOCAL-SEARCH-2($\boldsymbol{P'}$)
    $N_{pop} \leftarrow$ size($\boldsymbol{P'}$,1)
    $\boldsymbol{TL} \leftarrow \emptyset$ ▷ Initialize the Tabu List
    for $i \leftarrow 1, N_{pop}$ **do**
        $\boldsymbol{x}\_nb \leftarrow$ NB-GENERATE($\boldsymbol{P'}(i,:), N_{nb}, r$) ▷ Generate N_{nb} neighbors within radius r
        for $j \leftarrow 1, N_{nb}$ **do**
            $\boldsymbol{x}\_nbObjs \leftarrow$ OBJ-EVAL($\boldsymbol{x}\_nb(j,:)$) ▷ Objective evaluation
        end for
        $\boldsymbol{P}_{nb} \leftarrow \boldsymbol{P'}(i,:) \cup \boldsymbol{x}\_nb$ ▷ Combinel $\boldsymbol{P'}(i,:)$ and $\boldsymbol{x}_{nb}$
        $\boldsymbol{x}\_nbRRSE \leftarrow$ RRSE($\boldsymbol{P}_{nb}$) ▷ Calculate R_{RSE}
        Calculating the stopping indicator value SI_{dq} by Eq. (3.14) and Eq. (3.12).
        $best\_nb \leftarrow \min(\boldsymbol{x}\_nbRRSE < x^{R_{RSE}})$ ▷ The best dominating individual
        Update $best\_nb$ in the Tabu list $\boldsymbol{TL}$
    end for
    $\boldsymbol{P}_{impr2} \leftarrow \boldsymbol{TL}$
    return $\boldsymbol{P}_{impr2}$
end procedure

3.5 Experimental Results and Discussion

In the experiments, we use the continuous test problems introduced in Chapter 2. They are bi-objective ZDT test functions (ZDT1, ZDT2, ZDT3), and tri-objective test function DTLZ2. All the test problems are setup with the number of variables $n = 5$. All the experimental results are compared with the results obtained from the well-known MOEA algorithm, NSGA-II, with the same settings.

The size of the population in this experiment for both AMMOA and NSGA-II is setup to $N_p = 100$ for bi-objective functions (ZDT1-3), and $N_p = 200$ for the tri-objective function DTLZ2. In AMMOA, the number of cluster N_c is setup to 10 for bi-objective problems and 20 for tri-objective problems. The probability of Local Search 1 is set to 0.5. To reduce the computation time of the relative probabilities, we setup the size of the pool of random selected individuals is the $N_p/4$. In the diversity check module, we set up the autoregressive value $t_0 = 5$ that means the lower bound of the clustering quality index is calculated based on the 5 past values of I_Q.

The convergence and diversity of the population after each simulation run are evaluated by IGD metric which is introduced in Chapter 2. To calculate the IGD value after each simulation run, we need to know the actual Pareto optimal front of the test problem. If we do not know the actual Pareto fronts of the test problems, we can produce a very good upper approximation to the Pareto fronts by solving the linear programming (relaxation) of the Tchebycheff's approximation equation with a number of uniformly distributed weights $\boldsymbol{\lambda}$ for simple test problems [Jasz02].

To visualize the behavior of the stopping indicator SI_{dq} values, we first run the algorithm with the stopping criterion is the max iterations of 500 and the algorithms. We next implement the online stopping criterion to detect the convergence of the algorithm.

First we do the experiments with bi-objective problems (ZDT1, ZDT2, ZDT3). In these experiments, the population size is 100, and the max iterations is 500. The evolution of the IGD metrics for each test problems in the first 300 iterations are described in Fig. 3.4, 3.5, and 3.6.

It can be seen from Fig. 3.4, 3.5, and 3.6 that the proposed AMMOA performs very well on the bi-objective test problems. It is more superior than the NSGA-II in convergence and diversity. With the test problem ZDT1, AMMOA almost reaches the minimum IGD value after 20 simulation runs, while it is 180 simulation runs for NSGA-II. With the test problem ZDT2, AMMOA reaches the minimum IGD after 90 iteration runs, while NSGA-II almost reaches that value after 200 iteration runs. With the test

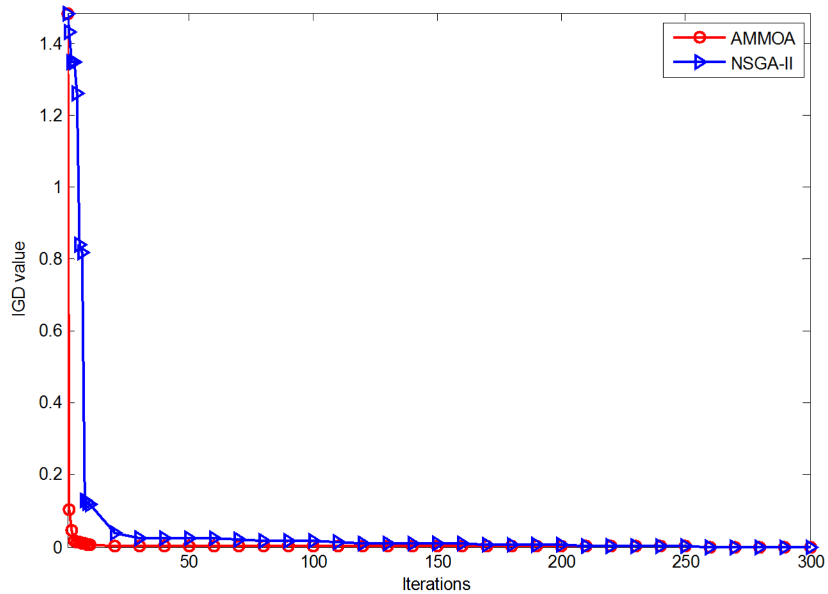


Figure 3.4: IGD metric in AMMOA and NSGA-II for the test problem ZDT1.

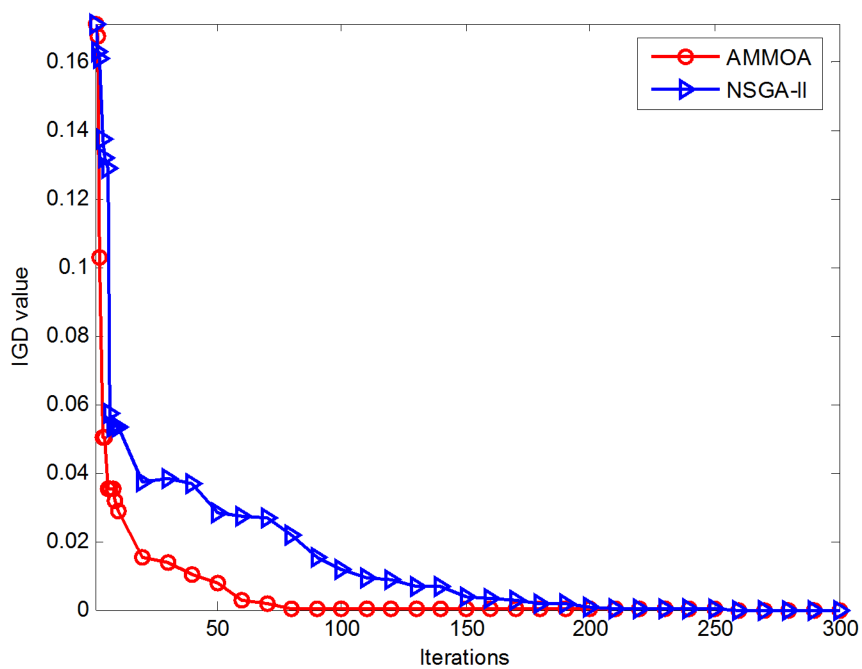


Figure 3.5: IGD metric in AMMOA and NSGA-II for the test problem ZDT2.

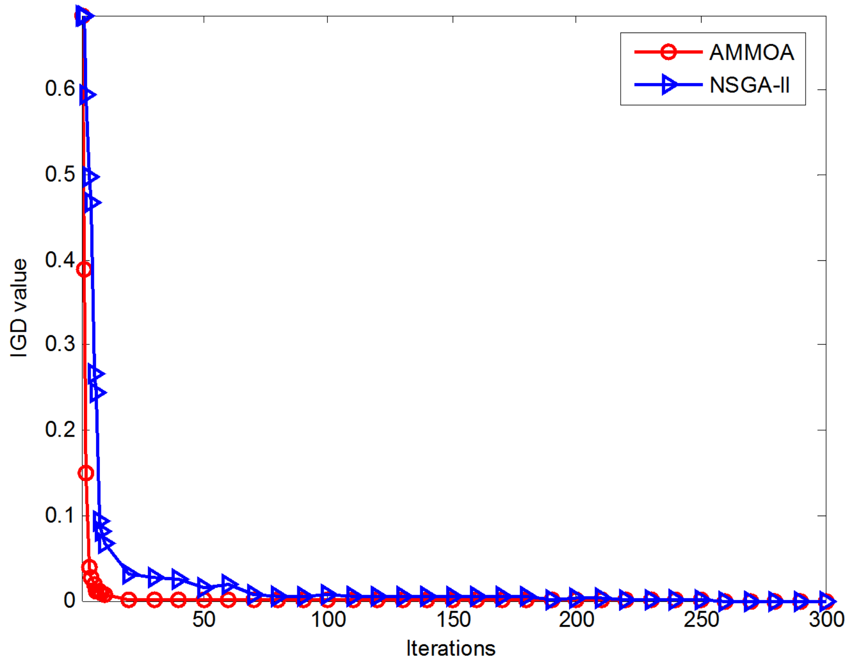


Figure 3.6: IGD metric in AMMOA and NSGA-II for the test problem ZDT3.

problem ZDT3, AMMOA obtains the minimum IGD after around 25 simulation runs, while it is 190 for NSGA-II. These results indicate that AMMOA converges much faster than NSGA-II in minimizing the IGD metric value for the bi-objective test problems.

To show the effective of the proposed AMMOA, we visualize the Pareto fronts of the obtained population after 30 simulation runs. These Pareto fronts of AMMOA are also compared with the Pareto fronts obtained by NSGA-II in the same settings. The Pareto fronts obtained by AMMOA and NSGA-II after 30 simulation runs for the bi-objective test problems ZDT1, ZDT2, and ZDT3 are described in Fig. 3.7, Fig. 3.8, and Fig. 3.9, respectively. The results show that AMMOA obtains very good convergence and diversity just in few simulation runs. It is much superior to the NSGA-II approach.

The online stopping indicators SI_{dq} calculated for 500 iterations for the test problems ZDT1, ZDT2, and ZDT3 are described in Fig. 3.10, Fig. 3.11, and Fig. 3.12, respectively. The stopping threshold is setup by $\epsilon = 10^{-3}$. With the test problem ZDT1, the algorithm

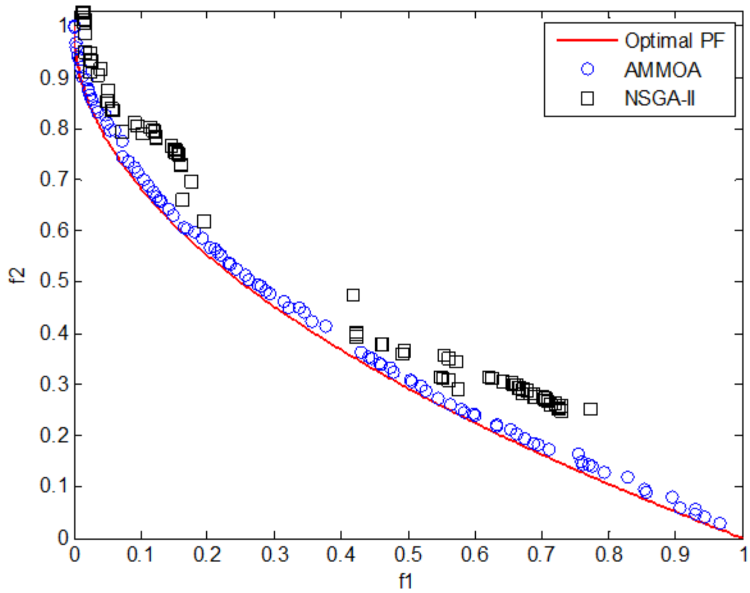


Figure 3.7: Pareto front of the non-dominated solutions after 30 iterations for the test problem ZDT1.

stops at the iteration 110 (the detected stopping criterion = 110). The algorithm stops at the iteration 170 for the test problem ZDT2, and the algorithm stops at the iteration 105 for the test problem ZDT3.

The obtained Pareto fronts of non-dominated solutions after stopping because of detecting the convergence (the online stopping criterion) for the bi-objective test problems ZDT1, ZDT2, and ZDT3 are illustrated in Fig. 3.13, Fig. 3.14, and Fig. 3.15, respectively. The detected Pareto optimal fronts are also compared with the fronts obtained by the algorithm NSGA-II with the prefixed 200 iterations. It can be seen that the solutions obtained by AMMOA are spread extremely well on the whole optimal Pareto fronts. Both AMMOA and NSGA-II obtained the optimal Pareto fronts. However, AMMOA converges very fast to the optimal front, and obtains very good diversities for the non-dominated populations.

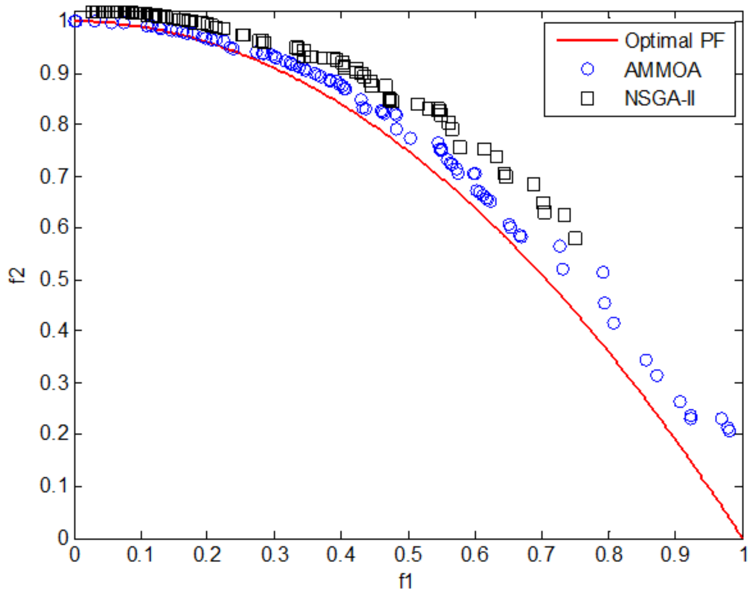


Figure 3.8: Pareto front of the non-dominated solutions after 30 iterations for the test problem ZDT2.

We next do the experiments with the three-objective test problem DTLZ2. In these experiments, the population size N_p is setup to 200, and the maximum iterations is also setup to 500. The online stopping indicators SI_{dq} calculated for 500 iterations for DTLZ2 is shown in Fig. 3.16. The stopping threshold is also setup by $\epsilon = 10^{-3}$. With this stopping threshold value, the algorithm stops at the iteration 252.

To show the performance of our proposed AMMOA, we plot the true Pareto optimal front of the test problem DTLZ2 for comparisons. The true Pareto optimal front and its 3D surface of the test problem DTLZ2 are shown in Fig. 3.17 and Fig. 3.18.

The evolution of the IGD metric for test problem DTLZ2 is described in Fig. 3.19. It can be also seen that that the proposed AMMOA performs well on the three-objective test problem DTLZ2. It is more superior than the NSGA-II algorithm in both convergence and diversity. However, both AMMOA and NSGA-II are not able to get the optimal

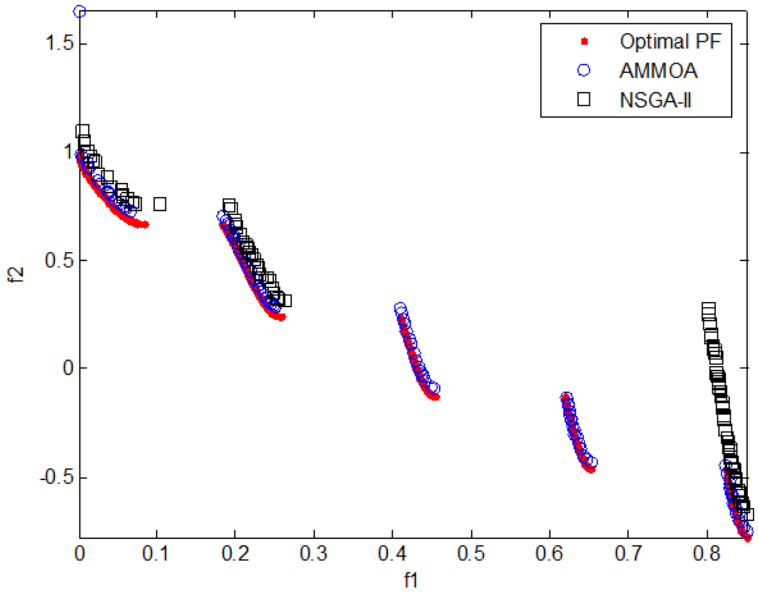


Figure 3.9: Pareto front of the non-dominated solutions after 30 iterations for the test problem ZDT3.

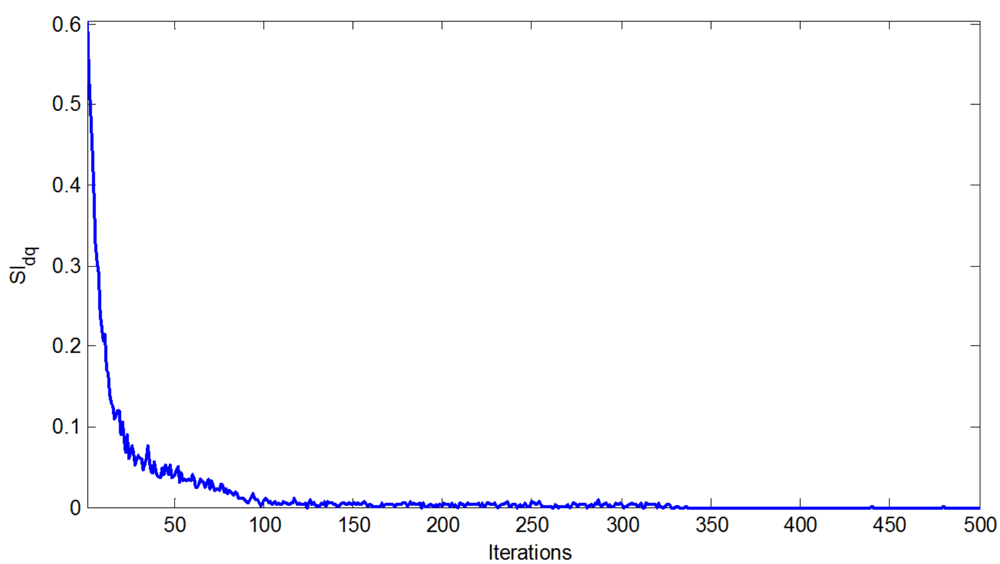


Figure 3.10: The online stopping indicator SI_{dq} of the test problem ZDT1.

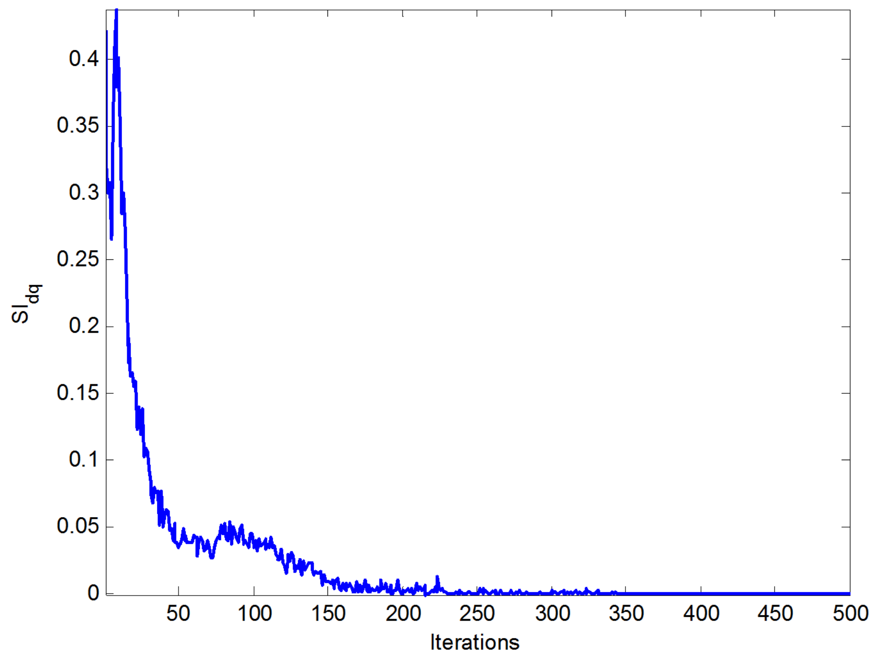


Figure 3.11: The online stopping indicator SI_{dq} of the test problem ZDT2.

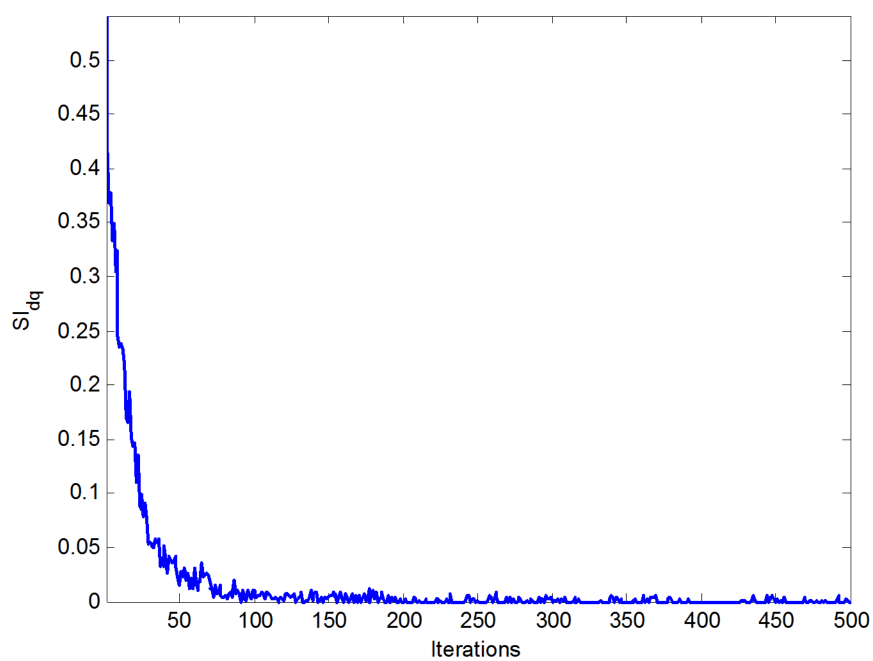


Figure 3.12: The online stopping indicator SI_{dq} of the test problem ZDT3.

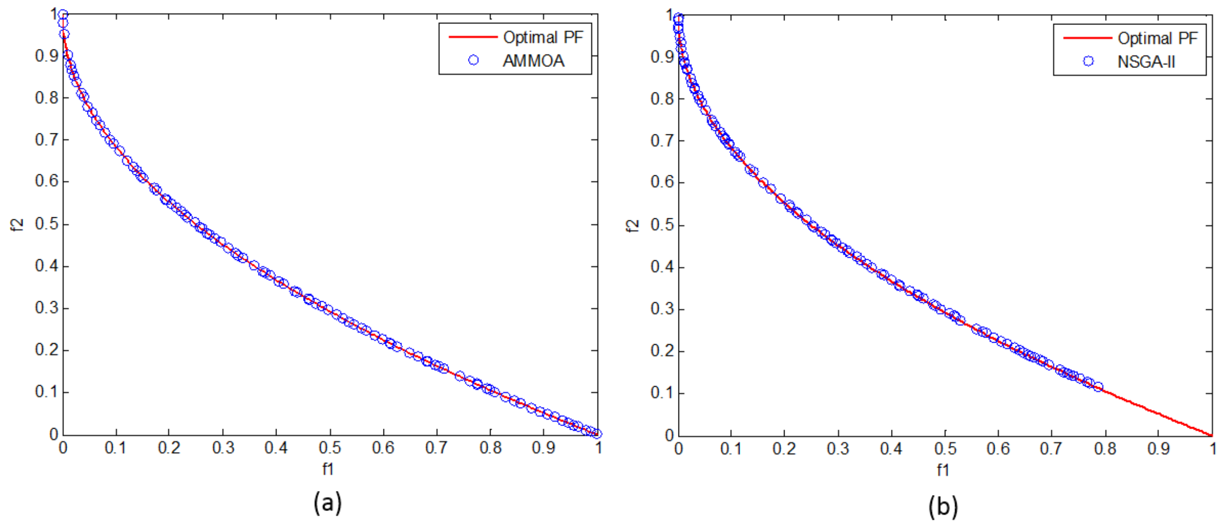


Figure 3.13: Pareto front of the non-dominated solutions for the test problem ZDT1: (a) AMMOA stopping at 110 iterations; (b) NSGA-II with the prefixed 200 iterations.

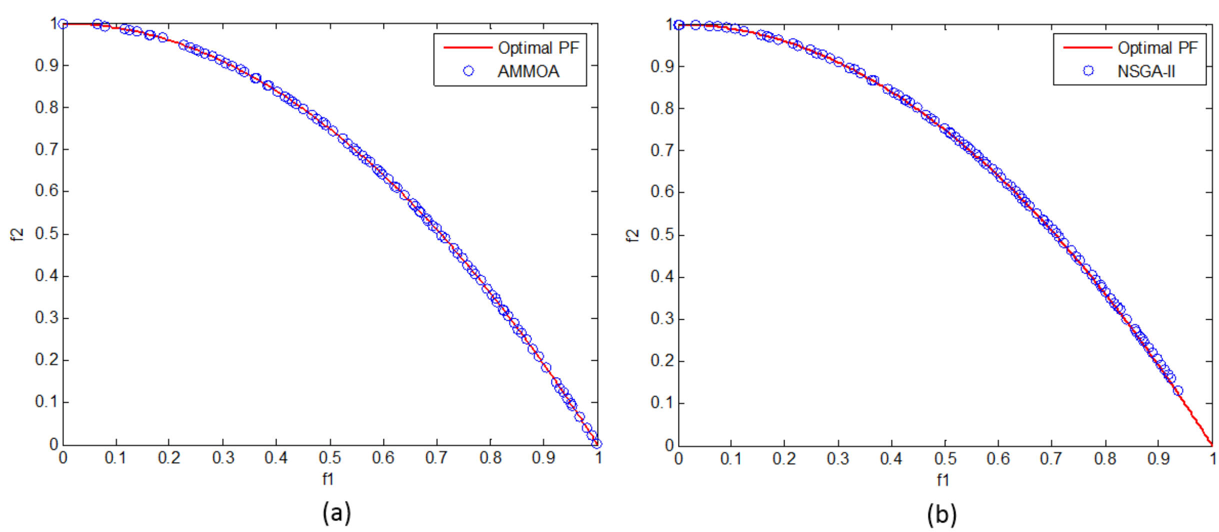


Figure 3.14: Pareto front of the non-dominated solutions for the test problem ZDT2: (a) AMMOA stopping at 170 iterations; (b) NSGA-II with the prefixed 200 iterations.

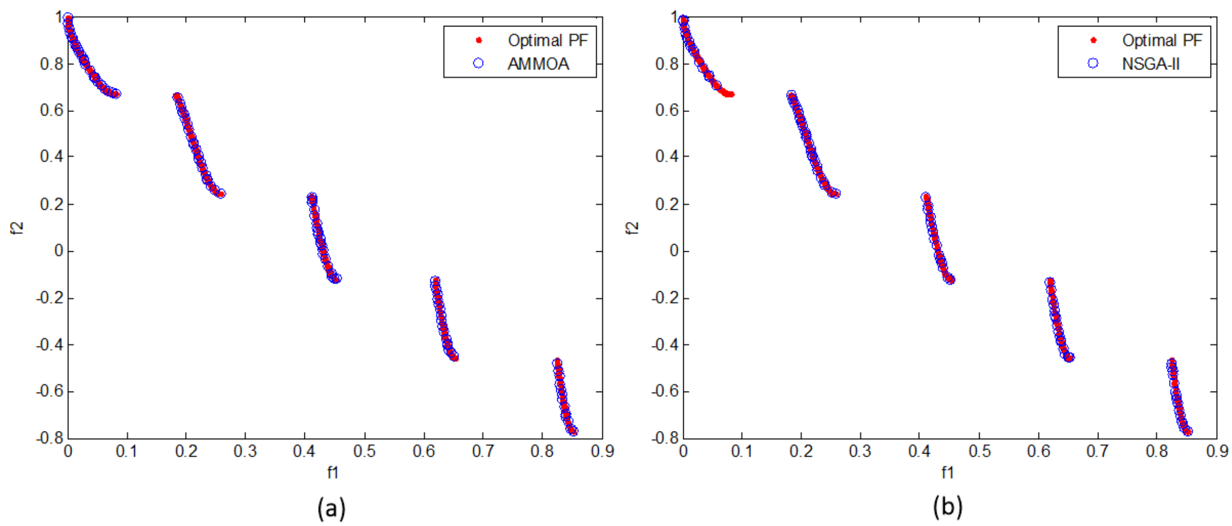


Figure 3.15: Pareto front of the non-dominated solutions for the test problem ZDT3: (a) AMMOA stopping at 105 iterations; (b) NSGA-II with the prefixed 200 iterations.

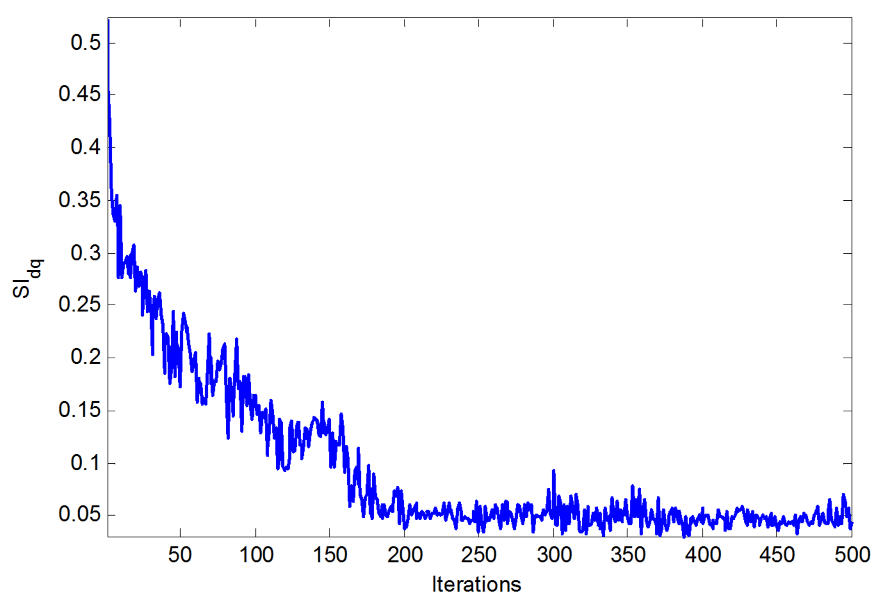


Figure 3.16: The online stopping indicator SI_{dq} of the test problem DTLZ2.

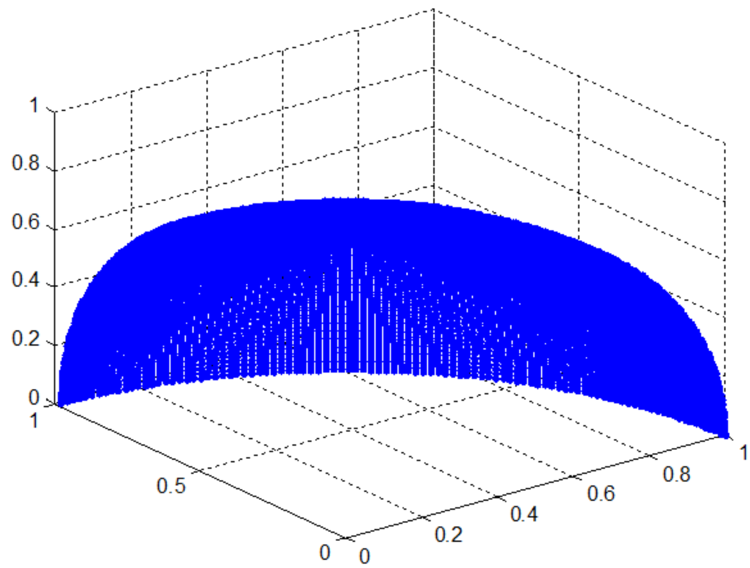


Figure 3.17: The true Pareto optimal front of the test problems DTLZ2.

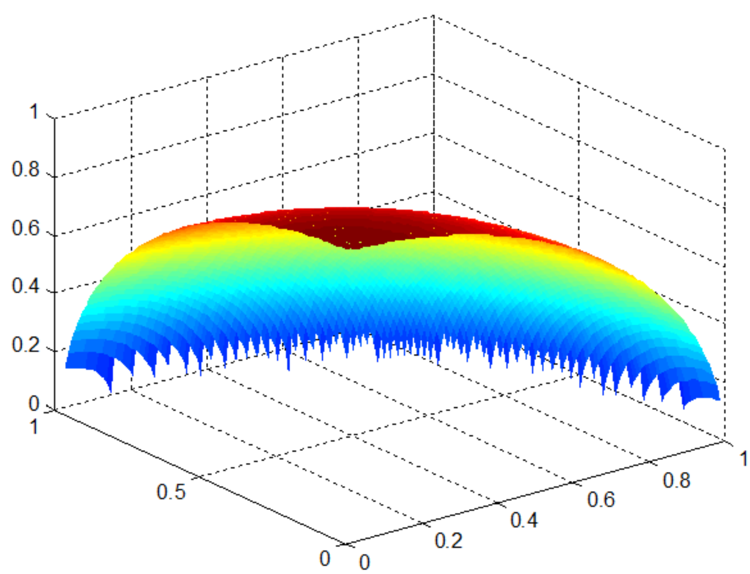


Figure 3.18: The 3D visualization of the true Pareto optimal front's surface of DTLZ2.

value of IGD metrics as in the two-objective test problems.

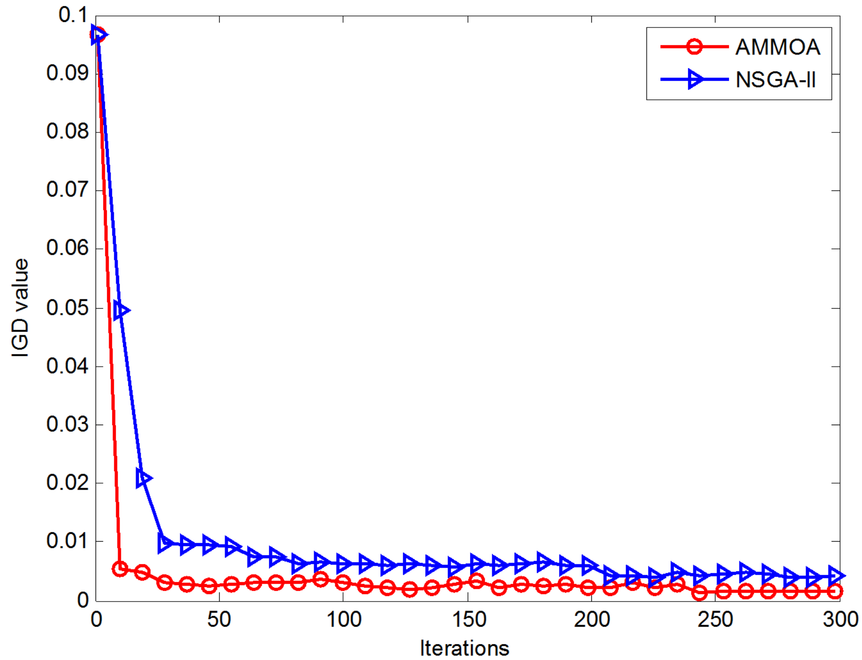
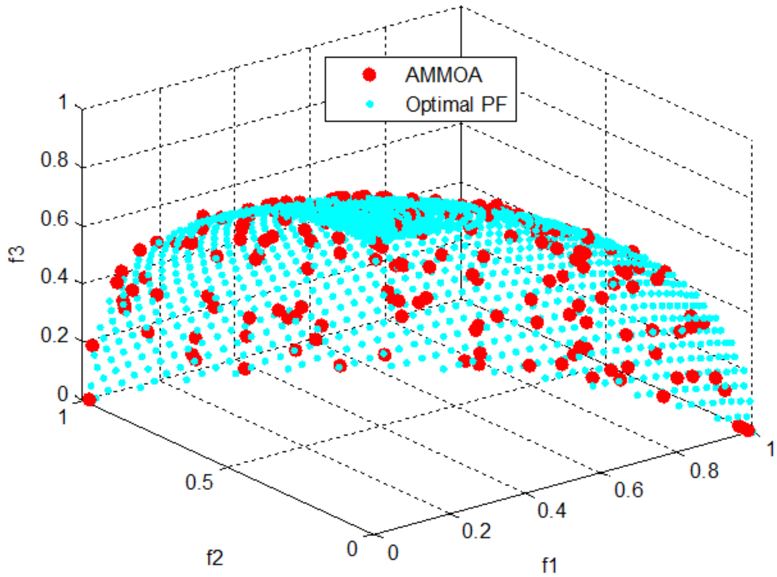
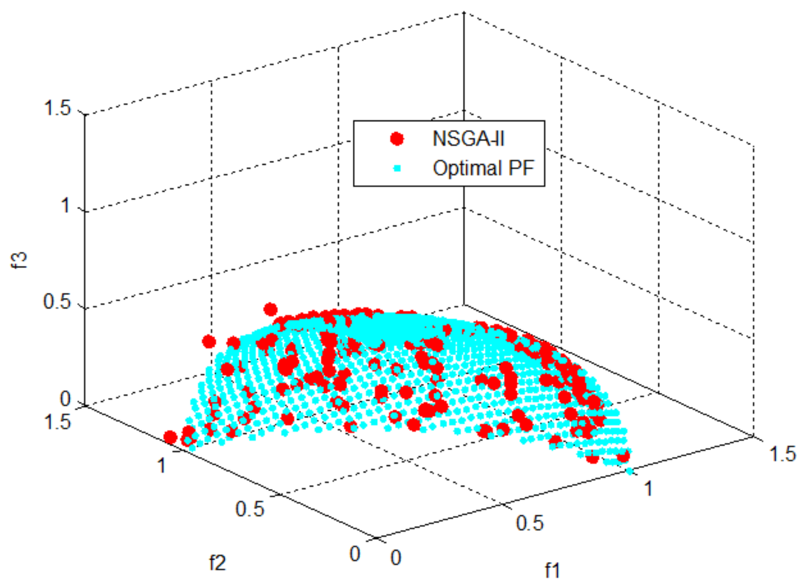


Figure 3.19: IGD metric in AMMOA and NSGA-II for the test problem DTLZ2.

The three dimensional obtained Pareto fronts of the non-dominated solutions after stopping at the detected convergence at the iteration 252 for the three-objective test problem DTLZ2 are shown in Fig. 3.20. The surfaces of these obtained Pareto fronts are depicted in Fig. 3.21. The results are compared with the fronts obtained by NSGA-II after the fixed 300 iterations. It can be seen from Fig. 3.20 that the solutions obtained by AMMOA are converged and spread well on the true Pareto optimal front. We can also see from Fig. 3.21 that while the surface of the front obtained by AMMOA is closely similar to the surface of the true Pareto optimal front. Therefore, the results obtained by AMMOA is much better than the results obtained by NSGA-II.



(a)



(b)

Figure 3.20: Pareto fronts of the non-dominated solutions for the three-objective test problem DTLZ2: (a) AMMOA stopping at 252 iterations; (b) NSGA-II with the prefixed 300 iterations.

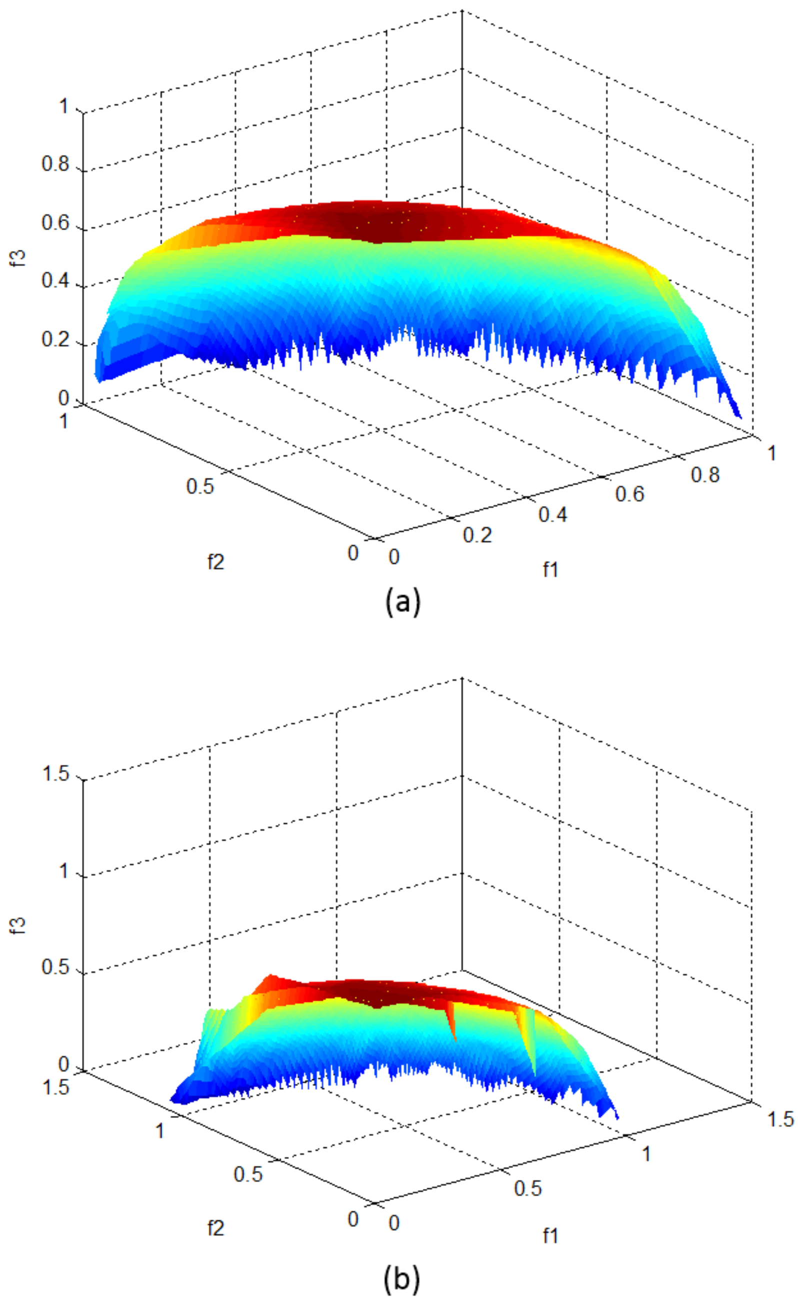


Figure 3.21: 3-D surfaces of the Pareto fronts of the non-dominated solutions for the three-objective test problem DTLZ2: (a) AMMOA stopping at 252 iterations; (b) NSGA-II with the prefixed 300 iterations.

3.6 Summary of Chapter 3

This chapter has presented a framework for adaptive multiobjective memetic optimization algorithm (AMMOA). We have introduced a new information theoretic based criterion used in AMOMA for guiding the selection, clustering, and local learning processes. The experimental results of the implementation of AMOMA have shown that the framework performs well on both two-objective and three-objective test problems, and it outperforms the well-known multiobjective optimization NSGA-II.

However, this implementation of AMMOA still has some disadvantages that need to improve in the future research. These disadvantages can be listed as follows.

- The proposed implementation of AMMOA adopts the elitist strategy of NSGA-II that use non-dominated sorting algorithm based on Pareto ranks and crowding-distance. This elitism works well on the problems having few objectives (2 or 3 objectives). However, it scales poorly with the problems having many objectives (more than 4 objectives). Thus, the performance of AMMOA is also poor with many-objective problems.

- The strategy of local refinements are still decided by observations from experimental results. More specifically, the probability of Local Search 1 is selected based on the experimental observations.

Chapter 4

Multiobjective Image Data Hiding

This chapter presents a hybridization of neural networks and multiobjective memetic optimization for an adaptive, robust, and perceptual data hiding method for colour images. The multiobjective optimization problem of a robust and perceptual image data hiding is introduced. In particular, trade-off factors in designing an optimal image data hiding to maximize the quality of watermarked images and the robustness of watermark are investigated. With the fixed size of a logo watermark, there is a conflict between these two objectives, thus a multiobjective optimization problem is introduced. We propose to use a hybrid between *general regression neural networks* (GRNN) and the *adaptive multiobjective memetic optimization algorithm* (AMMOA) to solve this challenging problem. Specifically, a GRNN is used for the efficient watermark embedding and extraction in the wavelet domain. Optimal watermark embedding factors and the smooth parameter of GRNN are searched by AMMOA. The experimental results show that the proposed approach achieves adaptation, robustness, and imperceptibility in image data hiding.

4.1 Introduction

Data hiding is the technique of embedding information (watermark) into a carrier signal (video, image, audio, text) such that the watermark can be extracted or detected later for copyright protection, content authentification, identity, fingerpringing, access control, copy control, and broadcast monitoring [WuLi03]. The important requirements for the data hiding systems are robustness, transparency, capacity, and security under different attacks and varying conditions [PaHJ04, MaDD04]. These requirements can vary under different applications. Consequently, a good data hiding technique should be adaptive to the environment. A more advanced approach should involve perception, cognition, and learning [Kins12, WABB12].

In general, data hiding can be categorized into two classes, depending on the domain of embedding the watermark [WuLi03], (i) spatial domain data hiding, and (ii) transformed domain data hiding. Spatial domain data hiding approaches [ChWo01, MuMA04, KaBe06, CoCh07] can be implemented easily and fast; however, they are usually susceptible to signal processing attacks such as compression, adding noise, and filtering. Transformed-domain data hiding techniques such as the watermarking methods based on the *discrete Fourier transform* (DFT) [XiZH06, AhMo06], *discrete cosine transform* (DCT) [SuOb03], and *discrete wavelet transform* (DWT) [HsWu98, ChLi03, WaLi04, GBIB06] typically offer more robustness under most of the casual signal processing attacks when compared with spatial domain watermarking schemes. Digital data hiding techniques are also classified based on the watermark data embedded into the host signal. A logo data hiding (logo watermarking) technique requires a visual watermark like a logo image, while a statistical data hiding (statistical watermarking) technique requires a statistical watermark like a pseudo random sequence. In statistical watermarking approaches (eg., [BaBP01, KuAS05]), watermarks are detected by statistical method to demonstrate that the watermark in the host signal is unchanged. In logo watermarking (eg., [YuTL01, ChLi10]), visual watermarks are extracted from the host signals for visual copyright proofs. These watermarks are not only assessed by machines but also by humans through their ability to recognize visual patterns through *human visual system* (HVS). Thus, the presentation of a visual watermark is much more persuasive than a numerical value of a statistical watermark.

Transparency and robustness are two main challenges in logo watermarking techniques since the logo consists of much information that is not easy to embed perceptually into a host signal. Moreover, the robustness in logo watermarking is so strict because it requires satisfactory recognition from human beings. With a fixed size of a logo watermark, there is a conflict between the transparency and robustness of the watermark. Increasing the transparency of watermark (or the quality of the watermarked image) decreases the robustness of the watermark and vice versa. A good logo watermarking is a robust data hiding with the acceptable quality of watermarked image. Thus, an optimal logo watermarking should be modeled as a multiobjective optimization problem.

Recently, some researchers have applied computational intelligence to design perceptual and robust data hiding systems, such as *back-propagation neural networks* (BPNN) based watermarking [PaHJ04, YuTL01, DaKi12b], *support vector machine* (SVM) based watermarking [WaYC08, TsSu07, ShFL05], and *genetic algorithms* (GA) based watermarking [SHWP04, RaRa11], which can detect or extract the watermark without requiring the original signal for comparison. BPNNs have been recently exploited for intelligent watermarking methods [PaHJ04, YuTL01]. The BPNNs have been used to extract the relationships between selected pixels or selected transformed coefficients and their neighbours for embedding and extracting the watermark bits. Thus, these algorithms are robust to the amplitude scaling and a number of other attacks. However, one key disadvantage of the BPNN is that it can take a large number of iterations to converge to the desired solution [Hayk99, Spec91]. The data hiding problems have been recently considered as single optimization problems. Shieh and coworkers [SHWP04] introduced a watermarking technique that use a GA to find the optimum frequency bands for embedding watermark bits into DCT coefficients that can improve imperceptibility or robustness of the watermark.

In this chapter, an optimal logo watermarking for colour images is formulated as a multiobjective optimization problem. To solve this problem, we propose a novel logo watermarking method based on wavelets, and the hybrid of a *general regression neural network* (GRNN) and the *adaptive multiobjective memetic optimization algorithm* (AMMOA). This new method is different from previous techniques in that it utilizes a GRNN to extract relationships between wavelet coefficients of the Y channel of the corresponding YCrCb image for embedding and extracting the watermark. Embedding factors (watermarking strengths) and GRNN's smooth parameter are searched optimally by the AMMOA to maximize the quality of the watermarked image and the robustness of the watermark. The main contributions of this work are as follows:

1. A novel logo watermarking method for colour images is proposed based on wavelets and GRNN. The optimality of the method is achieved by using AMMOA;
2. Different classes of wavelets are analyzed experimentally to select an appropriate wavelet for robust and perceptual image data hiding based on computational intelligence;
3. A new multiscale perceptual measure, the relative R*é*nyi dimension spectrum, is introduced for measuring the transparency of the watermark with remarkable results; and
4. A multiobjective optimization problem of image data hiding is introduced.

The chapter is organized as follows: In Sec. 4.2, the brief introduction of information hiding, wavelets, and GRNN are provided. The proposed watermark embedding and extraction algorithms are introduced in Sec. 4.3. The optimal data hiding using AMMOA is described in Sec. 4.4. Experimental results and discussions are given in Sec. 4.5.

4.2 Background on Methods Used

4.2.1 Theory of Information Hiding

Communication Model of Information Hiding

Information hiding can be considered as a basic communication theoretical model [DaKi12a, CoMM99, MoKo05, Cach04, MoSu03]. Cox and coworkers [CoMM99] suggested that information hiding closely resembles communications with side information at the transmitter and decoder, a configuration originally described by Shannon. Moulin *et al.* [MoSu03, MoKo05] formulated the information hiding problem as a communication problem where the hiding capacity is considered as the maximum rate of reliable communication through the communication system. A game theory approach was proposed to seek an upper bound of the hiding capacity. In this work, we use the theory of bags, as described by [Yage86, Kins12b], to explain the communication model of information hiding as depicted in Fig. 4.1.

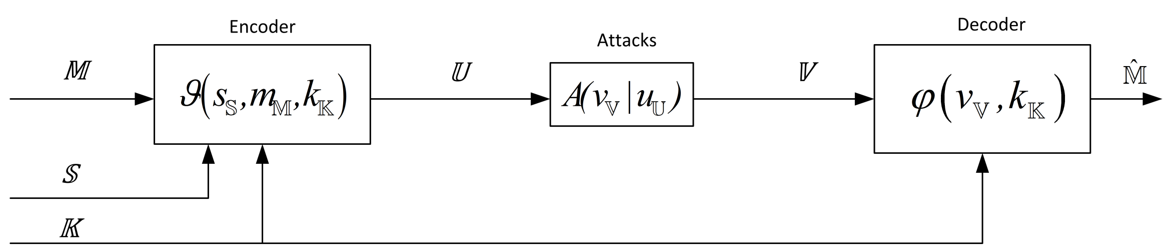


Figure 4.1: Communication theoretic model of information hiding.

In this model, we denote bag $\mathbb{S}$ as the host signal, bag $\mathbb{M}$ as the hidden message (watermark), bag $\mathbb{K}$ as the secret key shared between the encoder and the decoder, bag $\mathbb{U}$ as the embedded signal, bag $\mathbb{V}$ as the received (attacked) signal, and bag $\hat{\mathbb{M}}$ as the extracted hidden message.

At the encoder, there are three inputs (three sub-bags) of $\boldsymbol{s}_{\mathbb{S}} \subset \mathbb{S}$, $\boldsymbol{m}_{\mathbb{M}} \subset \mathbb{M}$, and $\boldsymbol{k}_{\mathbb{K}} \subset \mathbb{K}$. The message $\boldsymbol{m}_{\mathbb{M}}$ is first scrambled with the secret key $\boldsymbol{k}_{\mathbb{K}}$ which is independent of the host signal $\boldsymbol{s}_{\mathbb{S}}$, then embedded into the host signal $\boldsymbol{s}_{\mathbb{S}}$ to produce the embedded signal $\boldsymbol{u}_{\mathbb{U}} \subset \mathbb{U}$ using an embedding function $\boldsymbol{u}_{\mathbb{U}} = \vartheta(\boldsymbol{s}_{\mathbb{S}}, \boldsymbol{m}_{\mathbb{M}}, \boldsymbol{k}_{\mathbb{K}})$.

In transit, the embedded signal $\boldsymbol{u}_{\mathbb{U}}$ is influenced by intended or unintended interference such as noise addition, compression, filtering, amplitude scaling, and block-lost. These interferences are all considered as attacks created by attackers. An attacker takes the embedded signal $\boldsymbol{u}_{\mathbb{U}}$, and creates a modified signal $\boldsymbol{v}_{\mathbb{V}}$ by the function $\boldsymbol{v}_{\mathbb{V}} = A(\boldsymbol{v}_{\mathbb{V}}|\boldsymbol{u}_{\mathbb{U}})$. The attacker usually wants to produce the modified signal $\boldsymbol{v}_{\mathbb{V}}$ that is perceptually close to $\boldsymbol{u}_{\mathbb{U}}$, but destroys the hidden message in $\boldsymbol{u}_{\mathbb{U}}$.

At the decoder, the message $\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}}$ is extracted from the received (attacked) signal $\boldsymbol{v}_{\mathbb{V}}$ and the secret key $\boldsymbol{k}_{\mathbb{K}}$ by using the extracting function $\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}} = \varphi(\boldsymbol{v}_{\mathbb{V}}, \boldsymbol{k}_{\mathbb{K}})$.

Important Technical Issues of Information Hiding

The technical issues presented here are usually considered as requirements for an information hiding technique for a specific application. They include:

a) *Transparency*: In most applications, the embedded signal $\boldsymbol{u}_{\mathbb{U}}$ is required to be received perceptually as the host signal $\boldsymbol{s}_{\mathbb{S}}$. This means that the hidden message $\boldsymbol{m}_{\mathbb{M}}$ should be invisible in the host signal $\boldsymbol{s}_{\mathbb{S}}$. The transparency is measured by comparing the two signal $\boldsymbol{u}_{\mathbb{U}}$ and $\boldsymbol{s}_{\mathbb{S}}$ using the function $t = sim(\boldsymbol{u}_{\mathbb{U}}, \boldsymbol{s}_{\mathbb{S}})$. In practice, embedding a message into a host signal always creates a distortion, d, to the signal $\boldsymbol{s}_{\mathbb{S}}$. A perceptual information hiding should minimize the distortion d regarding the human visual system to obtain the maximum transparency t.

b) *Robustness*: Robustness refers to the ability of the hidden message (watermark), which is embedded into the host signal by an information hiding technique, to survive common attacks such as signal processing operations (compression, filtering, noise addition, desynchronization, cropping, insertions) [PeAK99, MoKo05]. The robustness of an information hiding system is measured by comparing the accuracy of the extracted hidden message $\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}}$ to the hidden message $\boldsymbol{m}_{\mathbb{M}}$ by the function $a = comp(\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}}, \boldsymbol{m}_{\mathbb{M}})$. There is a trade-off between the robustness and the transparency in an information hiding system.

c) *Capacity*: This refers to the number of bits of the hidden message $\boldsymbol{m}_{\mathbb{M}}$ that are able to be perceptually embedded into the signal $\boldsymbol{s}_{\mathbb{S}}$ by an information hiding technique. There is also a trade-off between the capacity and the transparency.

d) *Security*: In the worst case, when a pirate or attacker can extract the hidden message from the embedded signal $\boldsymbol{u}_{\mathbb{U}}$, the security guarantees that the pirate is not able to understand the extracted hidden message. In other words, security is the ability of the hiding algorithm to make the hidden message incomprehensible to the pirates/attackers. To have security, the hidden message $\boldsymbol{m}_{\mathbb{M}}$ is scrambled or encrypted by a scrambling or encryption technique with a secret key $\boldsymbol{k}_{\mathbb{K}}$ before being embedded into the host signal $\boldsymbol{s}_{\mathbb{S}}$.

e) *Detectability*: This refers to the ability of the hiding technique that makes the hidden message transparent to detection techniques given by the third parties. One might confuse the detectability with transparency. While the transparency refers to the transparency of the hidden message to the human perception, the detectability refers to the transparency of hidden message to the detection techniques such as statistical detection techniques. The detectability is an important requirement for steganography applications.

4.2.2 Wavelet Decomposition

Many wavelet-based watermarking methods have been introduced in the literature. Wavelets are widely used for image watermarking because wavelet decomposition is considered to closely mimic the HVS's structure in perception [LeKn92, WoPD99]. Extensive experimental research about the HVS has been conducted by visual psychologists over the years. They discovered that the human eye filters the image into a number of bands, each approximately one octave wide in frequency [LeKn92]. A wavelet transform is very suitable for identifying the disturbed areas where tamperings can be hidden more easily. This property allows one to exploit the HVS frequency masking effect for a perceptual watermarking [DaKi12a]. Each wavelet-based watermarking algorithm usually uses its own specific class of wavelets and decomposition level. More details about wavelet transforms and HVS based on wavelets used for perceptual watermarking presented in Appendix D. The questions of what are the optimal wavelets and what is the sufficient level of decomposition for image watermarking are still open-ended.

In this work, we investigate 36 wavelet functions in 5 wavelet families for image watermarking in connection to computational intelligence-based watermarking algorithm. They are Haar (known as Db1), Daubechies (Db2, Db3, Db4,..., Db10), Symlets (Sym2, Sym3,..., Sym8), Coiflets (Coif1, Coif2,..., Coif5), and Biorthogonal (Bior1.3, Bior1.5, Bior2.2,..., Bior6.8) wavelets. The test algorithm WAT-GRNN for embedding and extracting the watermark in wavelet domain is described in details in Section 4.3.1 and Section 4.3.2. In the first stage, we do simulations and comparisons for wavelet functions in each wavelet family. The experimental benchmark consists of a quality (transparency) test and common attacks such as noise addition, JPEG compression, filtering, cropping, amplitude scaling. We then select the wavelets that produced better results, and compare them together. Table 4.1 shows the *peak signal to noise ratio* (PSNR) of the watermarked images of the Lena image using these better wavelets in the case of using embedding factor of $\eta = 18$. An example of the robustness of watermark against Gaussian noise attack is depicted in Fig. 4.2. All wavelets used in these experiments are decomposed in 4 level. Based on these simulations, Sym2 wavelet offers us a better robustness and an acceptable quality for the watermarked image.

Table 4.1: PSNRs of watermarked image using different wavelets for Lena image

Wavelets	**db2**	**db4**	**sym2**	**bior1.3**	**coif2**
PSNR (dB)	42.25	42.66	42.46	41.97	42.81

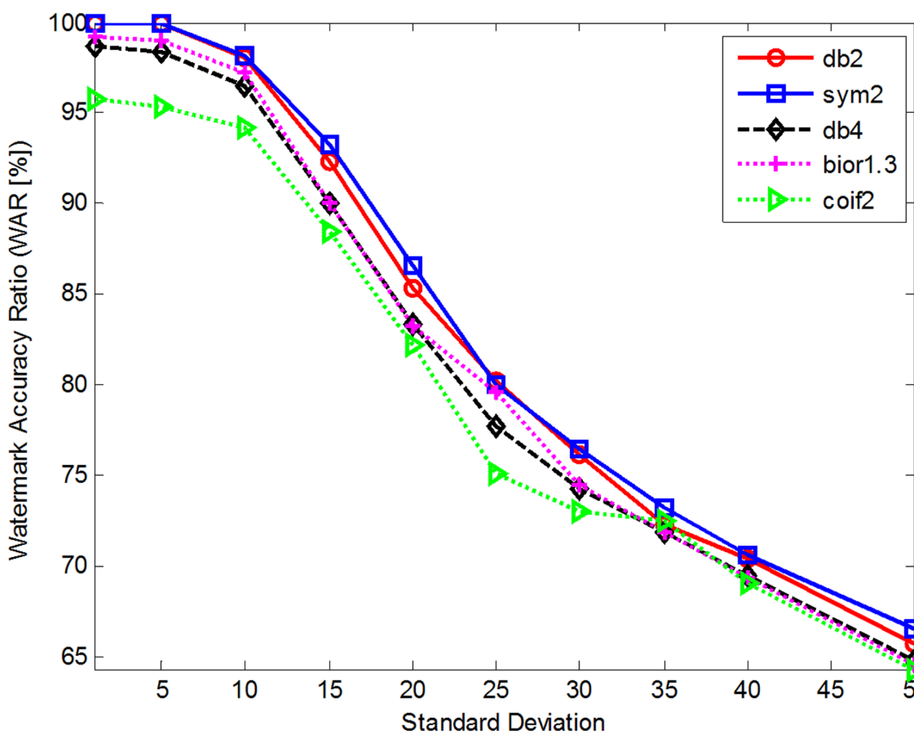


Figure 4.2: Robustness of wavelets against Gaussian noise addition attacks for Lena colour test image

The wavelet Sym2 is then selected to implement the algorithm with three different levels of decomposition. It can be seen from Fig. 4.3 that four-levels of decomposition provides a better robustness against Gaussian-noise attacks when compared to two-levels and three-levels of decomposition. The same behavior to JPEG compression, filtering, amplitude scaling, cropping attacks are also observed. From the above results, we choose Sym2 wavelet and four-levels of decomposition as the appropriate wavelet decomposition tool for our logo watermarking approach in this paper.

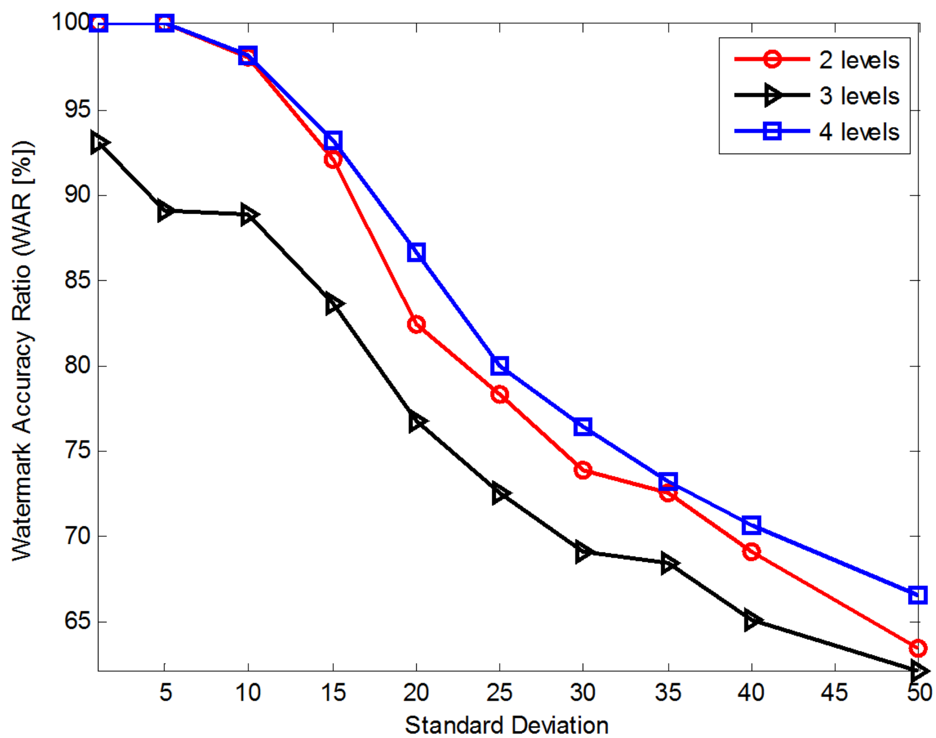


Figure 4.3: Robustness of the Sym2 wavelet against Gaussian noise addition attacks in different decomposition levels for Lena colour test image.

4.2.3 General Regression Neural Networks

Artificial neural networks are models inspired by the working of the human brain. They are set up with some unique attributes such as universal approximation (input-output mapping), the ability to learn from and adapt to their environment, and the ability to invoke weak assumptions about the underlying physical phenomena responsible for the generation of the input data [Hayk99]. A neural network can provide an approximation to any function of the input vector, provided the network a sufficient number of nodes [MaHi93]. Because of those universal features, neural networks are studied extensively for applications in classification, pattern recognition, forecasting, process control, image compression, and others. Various classes of neural networks such as perceptron networks, multilayer perceptron networks, radial-basis function networks, self-organizing map networks, recurrent networks, and probabilistic networks have been proposed. In this section, we provide a brief overview of the GRNN.

The GRNN, proposed by Specht [Spec91], is a special network in the category of probabilistic neural networks (PNN). GRNN is an one-pass learning algorithm with a highly parallel structure. Different from other probabilistic neural networks, GRNNs provide estimates of continuous variables and converges to the underlying (linear or nonlinear) regression surface. This makes GRNN a powerful tool to do predictions, approximation, and comparisons of large data sets. It also allows to have fast training and simple implementation. GRNN is successfully applied for image quality assessment [LiBW11], function approximation [GLZC07], and web-site analysis and categorization [AAKV04].

A diagram of the GRNN is shown in Fig. 4.4. In this diagram, a simple example of an one-dimensional input vector $\boldsymbol{X}[1, Q]$ is used to explain the calculation principle of the network. With the input of multidimensional vectors (i.e., matrices), it is considered as the vectors of one dimensional vector. The network has Q neurons at the input layer, Q neurons at the pattern layer, two neurons at the summation layer, and one neuron at the output layer. The input units are the distribution units. There is no calculation at this layer. It just distributes all of the measurement variable $\boldsymbol{X}$ to all of the neurons in the pattern units layer. The pattern units first calculate the cluster center of the input vector, $\boldsymbol{X}^i$. When a new vector $\boldsymbol{X}$ is entered the network, it is subtracted from the corresponding stored cluster center. The square differences d_i^2 are summed and fed into the activation function $f(x)$, and are given by

$$d_i^2 = (\boldsymbol{X} - \boldsymbol{X}^i)^T * (\boldsymbol{X} - \boldsymbol{X}^i) \tag{4.1}$$

$$f_i(\boldsymbol{X}) = \exp\left(-\frac{d_i^2}{2\sigma^2}\right) \tag{4.2}$$

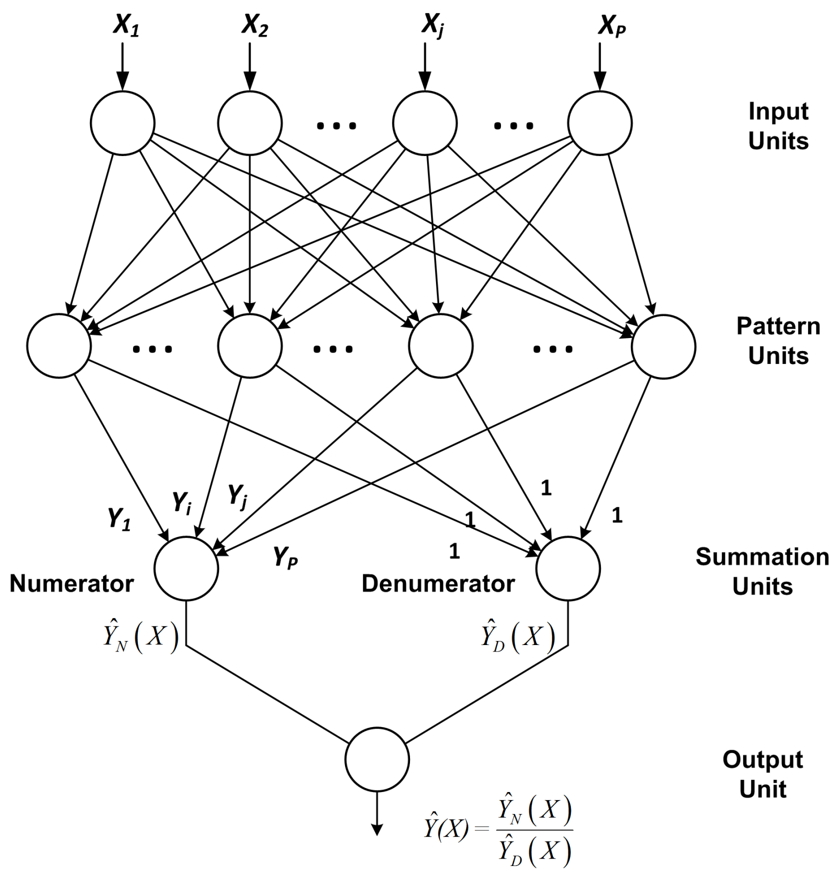


Figure 4.4: GRNN block diagram.

The signal of a pattern neuron i going to the *numerator* neuron is weighted with corresponding values of the observed values (target values), Y_i, to obtain the output value of the numerator neuron, $\hat{Y}_N(\boldsymbol{X})$. The weights of the signals going to the *denumerator* neuron are one, and the output value of the denumerator neuron is $\hat{Y}_D(\boldsymbol{X})$. The output value of the GRNN is the division of $\hat{Y}_N(\boldsymbol{X})$ and $\hat{Y}_D(\boldsymbol{X})$.

$$\hat{Y}_N(\boldsymbol{X}) = \sum_{i=1}^{Q} Y_i f_i(\boldsymbol{X}) \tag{4.3}$$

$$\hat{Y}_D(\boldsymbol{X}) = \sum_{i=1}^{Q} f_i(\boldsymbol{X}) \tag{4.4}$$

The output of GRNN is given by

$$\hat{Y}(\boldsymbol{X}) = \frac{\sum_{i=1}^{Q} Y_i f_i(\boldsymbol{X})}{\sum_{i=1}^{Q} f_i(\boldsymbol{X})} \tag{4.5}$$

In GRNN, only the standard deviation or a smooth parameter, σ, is subject to a search. To select a good value of, σ, Specht recommends the use of the holdout method [Spec91]. In our work, the optimal σ is searched by a multiobjective memetic algorithm for a perceptual and robust logo image watermarking.

4.3 Proposed Methods

4.3.1 Watermark Embedding Algorithm

The proposed watermark embedding scheme is depicted in the Fig. 4.5. In this work, we use an RGB colour image as the host image. The watermark image is a binary logo image. The RGB image is first converted to YCrCb colour image. The luminance component Y is decomposed by wavelet transform. In this paper, we only select the luminance component Y of YCbCr colour image for embedding the watermark because of the following reasons: (i) colour channels Cr and Cb have so much redundant information for HVS so that compression techniques for colour images do most compression work in these colour channels (hence, embedding watermark in CrCb creates more redundancy and make watermark susceptible to compression attacks); (ii) luminance Y is more sensitive to HVS that any tampering is easily detected (this makes watermarking in Y channel more robust than watermarking in color channels CrCb). The wavelet coefficients in each band are grouped into 3-by-3 non-overlapping blocks. Based on the random number sequence generated from the key (i, p), the algorithm selects which blocks for embedding watermark. These coefficients are used to train the GRNN. The watermark bits are embedded into selected coefficients by training the GRNN. Finally, inverse wavelet transform IDWT is applied to reconstruct the watermarked image. One can ask whether all components in the watermark embedding algorithm are necessary; for example, is the GRNN needed? The answer is yes. We propose to use the GRNN for a blind watermarking technique. At the decoder side, the watermark extraction process does not need to reference the original image to extract the watermark. It only needs to have the trained GRNN, which is obtain by the training process for embedding watermark at the encoder side, to extract the watermark from the watermarked image.

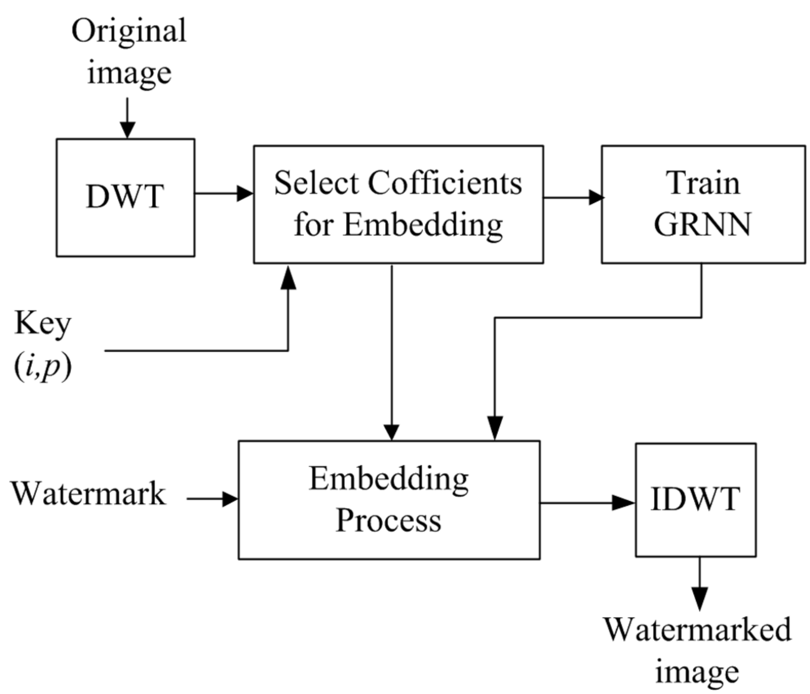


Figure 4.5: Block diagram of the proposed watermark embedding scheme.

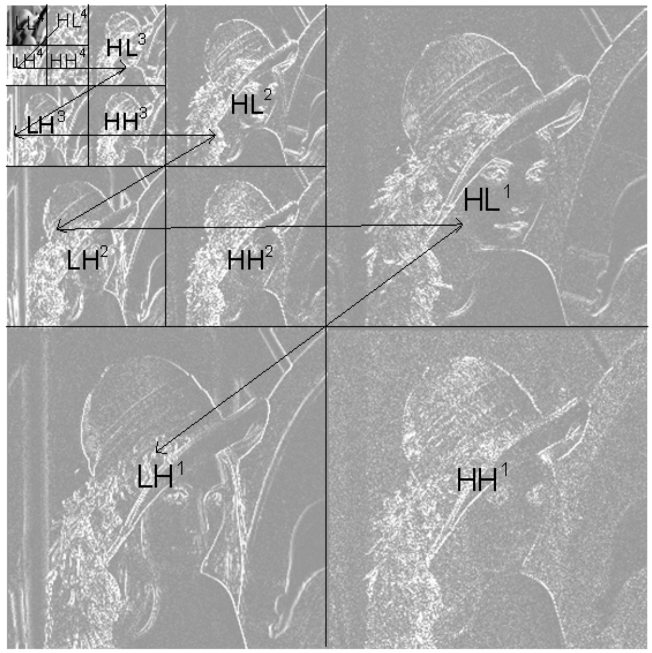


Figure 4.6: Intensity-adjusted display of 4-level wavelet decomposition of Lena colour image (wavelet subbands are rescaled to a gray-intensity range for display), and the scanning order of subbands for watermarking.

The Y component is decomposed by Symlet-2 ($sym2$) DWT in four levels as shown in Fig. 4.6. The watermark bits are embedded only into the following subbands: HL^4, LH^4, HH^4, HL^3, LH^3, HH^3, HL^2, LH^2, HH^2, HL^1, LH^1. In our scheme, scaling coefficients in LL^4 and coefficients in HH^1 are not used for embedding the watermark since embedding in LL^4 degrades the watermarked image while embedding the watermark in subband HH^1 makes the watermark more susceptible. These selected subbands are divided into non-overlapping 3-by-3 blocks and then scanned to arrange into a sequence of blocks with the subband order $HL^4LH^4HH^4HL^3LH^3HL^2LH^2HH^2HL^1LH^1$. The blocks for embedding watermarks are then selected randomly by the sequence of random non-repeated integer numbers generated by the Fibonacci p-code algorithm [ZAJP08] using the key (i, p). The Fibonacci p-code algorithm [ZAJP08] is selected because of the following reasons: (i) it is non-bandwidth expandable scrambling; (ii) the golden rate property of Fibonacci number offers the equally spectrum spread of the numbers; (iii) it offers secure key for security management (key (i, p)); and (iv) it is easy to implement in a complex watermarking algorithm. The relationship between wavelet coefficients and its neighbours in selected 3-by-3 blocks are extracted by a given GRNN for watermark embedding and extracting processes. The Fibonacci p-code sequence is defined by

$$F_p(n) = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{if } n = 1, \\ F(n-1) + F(n-p-1) & \text{if } n > 1, p \in Z^+ \end{cases} \tag{4.6}$$

Then for K sequence (k=1,2,...,K), the sequence of random integer numbers $T_k = T_1, T_2, ..., T_K$ is generated by

$$T_k = k(F_p(n) + i) \bmod F_p(n+1) \tag{4.7}$$

where $k = 1, 2, 3, ..., K$; $i \in [-3, 3]$ and i is an integer such that $F_p(n) + i < F_p(n+1)$. The security key or the key to generate K non repeated random integer numbers are parameters (i, p).

We now have selected blocks for embedding watermark bits. With each block B_i having the center coefficient $I(i, j)$, the input vector X_i and target T_i are set up as in Eq. (4.8) to train the GRNN with 8 input neurons, 8 pattern neurons, 2 summation neurons, and 1 output neuron. Where $i = 1, 2, ..., K$; K is the number of watermark bits.

$$\begin{cases} X_i = \big[I(i-1, j-1), I(i-1, j), I(i-1, j+1), \\ \qquad I(i, j-1), I(i, j+1), I(i+1, j-1), \\ \qquad I(i+1, j), I(i+1, j+1)\big] \\ T_i \; = [I(i, j)] \end{cases} \tag{4.8}$$

With each pair (X_i, T_i), the GRNN produces the ouput $\hat{I}(i, j)$. The watermark bits are embedded into the selected block-center coefficients according to

$$I_w(i, j) = \hat{I}(i, j) + \eta(i)(2W(i) - 1) \tag{4.9}$$

where $\eta(i)$ is the watermarking factor for each embedding watermark bits to selected block-center coefficient $I(i, j)$ of selected block B_i. They can be altered to obtain the imperceptibility and robustness. If η is small, we get the higher quality of watermarked image, but lower level of robustness, and vice versa. This is a trade-off between the quality of the watermarked image with the robustness of watermark. $W(i)$ is the i^{th} watermark bit in the sequential watermark bits. $I_w(i, j)$, the watermarked coefficient, is obtained by replacing the central coefficient $I(i, j)$ by the combination of the output of the GRNN $\hat{I}(i, j)$ and the watermark bit $W(i)$. After embedding, an inverse DWT is performed to get the watermarked luminance Y. By combining the watermarked Y with Cr, Cb and converting to RGB, the colour watermarked image is reconstructed. This embedding algorithm is denoted as WAT-EMB procedure.

4.3.2 Watermark Extraction Algorithm

The watermark extraction scheme is illustrated in Fig. 4.7. The extraction process is the inverse process of the embedding process. The colour watermarked image is first converted to $YCrCb$ colour domain. The luminance Y is then decomposed by 4-level Symlet-2 DWT. The wavelet coefficients are grouped into 3-by-3 blocks and arranged into the ordering sequence as described in Sec. 4.3.1. From the key (i, p) received, the sequence of random integer numbers are generated based on the Fibonacci p-code algorithms to detect the watermarked blocks. Denote I_w is the wavelet decomposition of the component Y of the watermarked image. From the detected blocks, we setup the input vector X_i' as in Eq. (4.8). The trained GRNN obtained in the embedding process

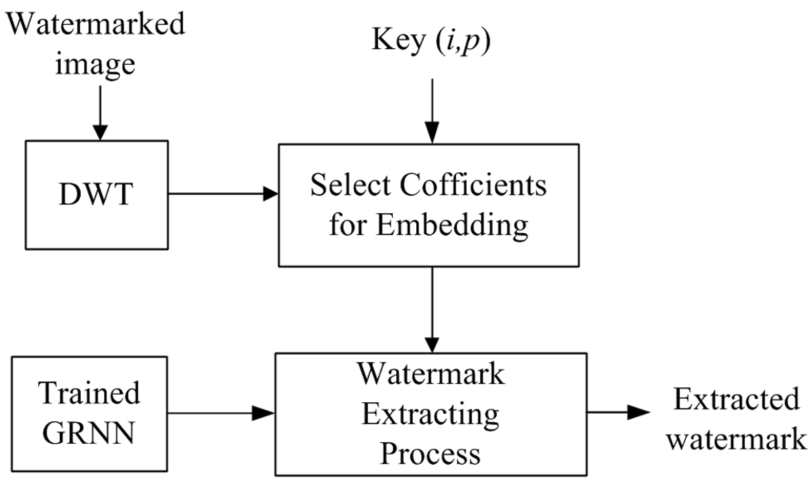


Figure 4.7: Block diagram of the proposed watermark extraction scheme.

is used to extract the watermark bits. With each input vector X'_i, the trained GRNN produce the output $\tilde{I}(i,j)$. The watermark bit extraction is performed by

$$\tilde{W}(i) = \begin{cases} 1 & \text{if } I_w(i,j) \geq \tilde{I}(i,j) \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

where i=1,2,...,K, K is the block number, and also is the number of watermark bits. $\tilde{W}$ is the extracted watermark. The extraction algorithm is denoted as WAT-EXTR procedure.

If the watermarking algorithms described in Secs. 4.3.1 and 4.3.2 use a fixed value η and a predefined fixed value of smooth parameter of GRNN, σ (for example $\eta = 18$, $\sigma = 0.5$), we label it as WAT-GRNN algorithm.

4.3.3 Optimal Image Data Hiding Using AMMOA

In logo watermarking, with the fixed logo watermark, there always exist two conflicting objectives. These are robustness of the watermark and quality of the watermarked image (imperceptibility or transparency of watermark). In this work, we apply AMMOA to search for the optimal parameters. They are the smooth parameter of the GRNN σ, and K embedding factors $\eta(i)$, $i = (1, 2, ..., K)$ to maximize the quality of watermarked image and the averaged robustness of watermark in the cases of noise addition, JPEG compression, amplitude scaling, and filtering attacks. The optimal image data hiding using AMMOA is graphically described in Fig. 4.8.

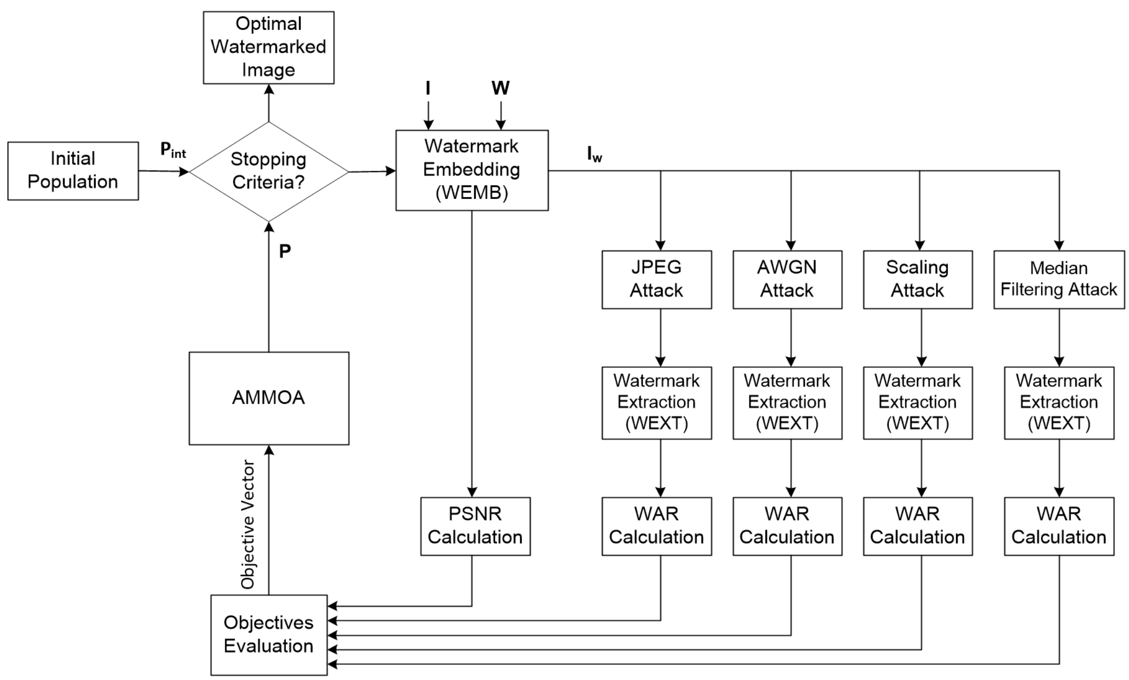


Figure 4.8: Graphical description of WAT-AMMOA method.

The inputs consist of the N number of chromosomes in population P, the colour image I, the watermark W, and key (i, p). From the key (i, p), the algorithm generates a sequence of random numbers RN based on the Fibonacci p_code algorithm from Eqs. (4.6) and (4.7). Each chromosome consists of $(1+K)$ genes. The first genes represents for the smooth parameter σ of the GRNN used for embedding and extracting the watermark. The next K genes represents K embedding factors $\eta(i)$ with $i = 1, 2, ..., K$, where K is the number of watermark bits embedded into the image.

The objectives evaluation procedure OBJ-EVAL is used to evaluate objectives for each chromosome in the given population. In this work, we search for optimal watermarking parameters to maximize the quality of watermarked image, and the averaged robustness of watermark in the case of noise addition attack, JPEG compression attack, amplitude scaling attacks, and filtering attacks.

The algorithm is briefly described as follows. With each chromosome in the given population, the algorithm extracts the parameter of GRNN and the K embedding factors to implement the embedding process WEMB that embeds K watermark bits W to the colour image I. The watermarked image is then tampered by the JPEG compression, Gaussian noise addition, amplitude scaling, and median filtering attacks. After each attack, the algorithm extracts the watermark and measure the robustness of the watermark by WAR measure. The parameters PSNR, and WARs are then fitted to the objective evaluation to setup the objective vectors for AMMOA. After each iteration of AMMOA, a better population $\boldsymbol{P}$ is searched. The population $\boldsymbol{P}$ then replaces the initialized population $\boldsymbol{P}_{int}$ for the next iteration. Thus, the population is updated after each iteration. The process continues until reaching the termination criteria. When the algorithm finishes, the best solution or best chromosome (S_{best}) is selected from the non dominated population $\boldsymbol{P}$. Finally, we obtained the watermarked image I_W by implementing the watermark embedding algorithm presented in Sec. 4.3.1 (WAT-EMB) with smooth parameter $\sigma = S_{best}(1)$, embedding factors $\eta(i) = S_{best}(i+1)$, $i = 1, 2, ..., K$. At the decoder side, the watermark is extracted by the watermark extraction process presented in Sec. 4.3.2 (WAT-EXTR). The initialization and objective evaluation algorithms are discussed as follows.

1) *Initialization*: Each chromosome represents $1 + K$ real nonegative parameters to be searched. The first parameter is the smooth parameter of the GRNN, σ, which is set in the range from 0.01 to 5. The K remaining parameters represents for the K watermarking factors $\eta(i), i = 1, 2, ..., K$. The watermarking factors are searched in a

wide range from 1 to 50.

2) *Objective Function Evaluation*: The objective function uses the *peak signal to noise ratio* (PSNR) as the quality objective, and the averaged *watermark accuracy ratio* (WAR) in the cases of four different attacks as robustness objectives. The PSNR is defined by

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{I_{peak}^2}{\mathrm{MSE}}\right) \tag{4.11}$$

where I_{peak} is the maximum intensity value of the three color channels R, G, B, and the *mean squared error* (MSE) computed for all three color channels R, G, and B is given by

$$\mathrm{MSE} = \frac{1}{KMN}\sum_{k=1}^{3}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j,k) - I_W(i,j,k))^2 \tag{4.12}$$

The watermark accuracy ratio is defined by

$$\mathrm{WAR} = \frac{\sum_{i=1}^{M_w}\sum_{j=1}^{N_w} W(i,j)\bar{\oplus}\tilde{W}(i,j)}{M_w * N_w} \tag{4.13}$$

where W and $\tilde{W}$ are the original and extracted watermarks, and (M_w, N_w) is the size of the watermarks. The logic operator $\bar{\oplus}$ does comparison between W and $\tilde{W}$. $W(i,j)\bar{\oplus}\tilde{W}(i,j) = 1$ if $W(i,j)$ and $\tilde{W}(i,j)$ have the exactly same value of 0 or 1. If WAR $\geq 70\%$, the extracted watermark can be considered as the original watermark. It is close to be perfect if WAR $\geq 85\%$.

Let $K = M_w * N_w$ be the number of watermark bits embedded into the image. We denote $\bar{\boldsymbol{\alpha}} = [\alpha_1, \alpha_2, ..., \alpha_{K+1}]$ as the watermarking parameters to be searched, where $\alpha_1 = \sigma$ (the smooth parameter of the GRNN), $\alpha_{2:K+1} = \eta(1 : K)$ (the embedding factors). The objectives function is then set up as follows

$$\bar{f}(\bar{\boldsymbol{\alpha}}) = [f_1(\bar{\boldsymbol{\alpha}}), f_2(\bar{\boldsymbol{\alpha}})] \tag{4.14}$$

where

$$f_1(\bar{\boldsymbol{\alpha}}) = \mathrm{PSNR}(\bar{\boldsymbol{\alpha}}) = \mathrm{PSNR}(\alpha_1, \alpha_2, ..., \alpha_{K+1})$$

and

$$f_2(\bar{\boldsymbol{\alpha}}) = \frac{W_G(\bar{\boldsymbol{\alpha}}) + W_J(\bar{\boldsymbol{\alpha}}) + W_A(\bar{\boldsymbol{\alpha}}) + W_M(\bar{\boldsymbol{\alpha}})}{4}$$

where W_G is the WAR in the case that the watermarked image is tampered by the Gaussian noise addition attack; W_J is the WAR under JPEG compression attack; W_A is the WAR under the amplitude scaling attack; and W_M is the WAR under the median filtering attack. Our optimal watermarking problem is to search for optimal parameters $\bar{\alpha}$ that can be formed by

$$\max_{\bar{\boldsymbol{\alpha}}} \bar{f}(\bar{\boldsymbol{\alpha}}) = \max_{\bar{\boldsymbol{\alpha}}} [f_1(\bar{\boldsymbol{\alpha}}), f_2(\bar{\boldsymbol{\alpha}})] \tag{4.15}$$

The pseudocode of our objective function evaluation is described in Algorithm 14.

4.4 Experimental Results and Discussion

In this section, experimental results are demonstrated and discussed to show the watermark robustness and transparency of the proposed algorithm. In the embedding process, the memetic algorithm is used to search for optimal watermarking factors and the optimal smooth parameter of the GRNN. In the watermark extraction process, the original image is not required, but the secret key (i, p), the smooth and weight parameters of the trained GRNN from the embedding process are needed. The watermark extraction process is the same as the watermark extraction algorithm described in the Sec. 4.3.2 (WAT-EXTR). The experimental results obtained from the proposed algorithm using multiobjective memetic algorithm (WAT-AMMOA) are compared with results of the WAT-GRNN algorithm, Kutter's method [KuJB98], and Yu's method [YuTL01]. WAT-GRNN is the watermarking algorithm used WAT-EMB in Sec. 4.3.1 and WAT-EXTR in Sec. 4.3.2 with the fixed embedding factor (embedding strength) $\eta = 18$, and the smooth parameter of the GRNN $\sigma = 0.5$. In the Yu's and Kutter's methods, we setup the wa-

Algorithm 14 Objectives Evaluation

procedure OBJ-EVAL(P, I, W, R_N)
  $N \leftarrow size(\boldsymbol{P}, 1)$ ▷ Number of chromosome in population $\boldsymbol{P}$
  for $i \leftarrow 1, N$ **do**
    $\sigma \leftarrow \boldsymbol{P}(i, 1)$ ▷ Smooth parameter of GRNN
    $[I_W, grnn\_weight] \leftarrow$ WAT-EMB($\boldsymbol{P}(i, :), I, W, R_N$)
    $f_1 \leftarrow$ PSNR(I_W, I)
    $I_{WG} \leftarrow I_W + GaussNoise$ ▷ AWGN attack
    $\tilde{W} \leftarrow$ WAT-EXTR($I_{WG}, grnn\_weight, \sigma, R_N$)
    $W_G \leftarrow$ WAR($\tilde{W}, W$)
    $I_{WJ} \leftarrow$ JPEG(I_W) ▷ JPEG compression attack
    $\tilde{W} \leftarrow$ WAT-EXTR($I_{WJ}, grnn\_weight, \sigma, R_N$)
    $W_J \leftarrow$ WAR($\tilde{W}, W$)
    $I_{WA} \leftarrow$ AmplitudeScaling(I_W) ▷ Scaling attack
    $\tilde{W} \leftarrow$ WAT-EXTR($I_{WA}, grnn\_weight, \sigma, R_N$)
    $W_A \leftarrow$ WAR($\tilde{W}, W$)
    $I_{WM} \leftarrow$ MedianFilter(I_W) ▷ Median filtering attack
    $\tilde{W} \leftarrow$ WAT-EXTR($I_{WM}, grnn\_weight, \sigma, R_N$)
    $W_M \leftarrow$ WAR($\tilde{W}, W$)
    $f_2 \leftarrow (W_G + W_J + W_A + W_M)/4$
    $f(i, :) \leftarrow [f_1, f_2]$
  end for
  return f
end procedure

termark strength $\alpha = 0.2$ to have a good robustness to be compared to the proposed algorithm WAT-AMMOA.

To evaluate the performance of our watermarking algorithms, the "Winipeg Jet" logo is embedded into various colour images. The binary watermark of size 64-by-64 is embedded into highly-textual colour images "Lena", "Baboon", "Airplane-F16", and "House" each with size of (512-by-512)-by-3. These test images are provided in Appendix C.

4.4.1 Results of Adaptive Multiobjective Memetic Optimization Algorithm

In the WAT-AMMOA algorithm, which uses the multiobjective memetic optimization to search for optimal watermarking factors and the smooth parameter of GRNN. The initial population consisting of 100 individuals is shown in Fig. 4.9. The algorithm uses the online stopping criterion described in Chapter 4 to detect the convergence. The stopping threshold is setup with $\epsilon_s = 0.02$. The obtained Pareto front is described in Fig. 4.10.

The experimental results in Fig. 4.10 shows that our WAT-AMMOA algorithm is able to obtain a set of efficient Pareto solutions, or the Pareto front. The objective, PSNR, and the objective, averaged WARs, are maximized significantly. We can see that these two objectives are conflicting seriously. There exists a set of efficient solutions for this optimization problem instead of one solution. Obtaining these efficient solutions helps the watermarking system select the best-suited solution in the context of the working environment at every moments. This is an important behavior of an adaptive and cognitive watermarking system. For instance, at the moment $t1$, the mission requires a balance between the robustness of watermarks and the quality of embedded images. In this case, for example with the input image of Lena, at the moment $t1$, the watermarking system selects the optimal solution having the PSNR of 42.82 dB, and the averaged WARs of

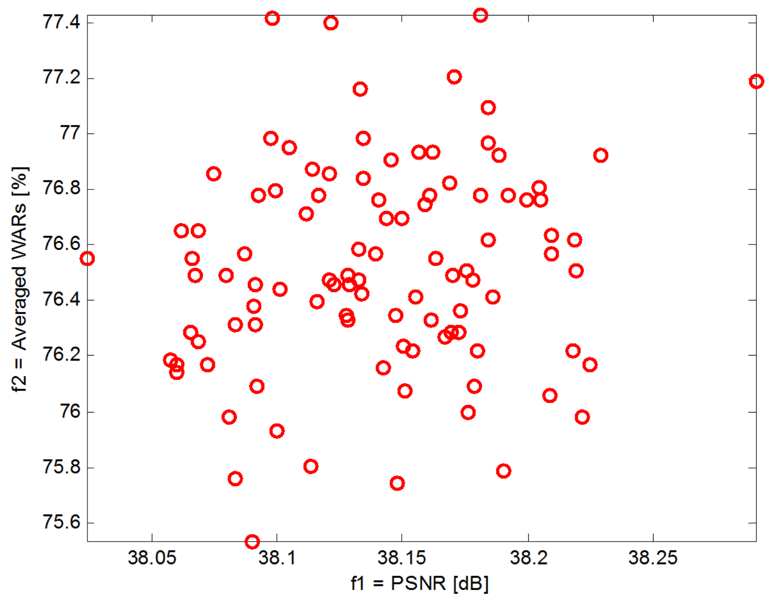


Figure 4.9: The objective space of WAT-AMMOA's initial population for Lena colour image.

80.72 %. This obtained solution for the image of Lena includes the smooth parameter of GRNN and 64x64=4096 embedding factors. The obtained optimal embedding factors are illustrated in Fig. 4.11 corresponding the smooth parameter of GRNN $\sigma = 2.48$.

If at the moment $t2 > t1$, the mission requires to increase the robustness of watermarks, the watermarking system can then select one optimal solution in the Pareto optimal set, with the higher value of the averaged WARs objective. In this situation, the optimal embedding factors increase to improve the robustness of watermarks; however, it reduces the quality of embedded images. This is a trade-off and win-win situation.

4.4.2 Quality Evaluation

To measure the transparency or the similarity of the watermarked image to the original image, watermarking systems mostly employ the PSNR. In Fig. 4.12, the differences

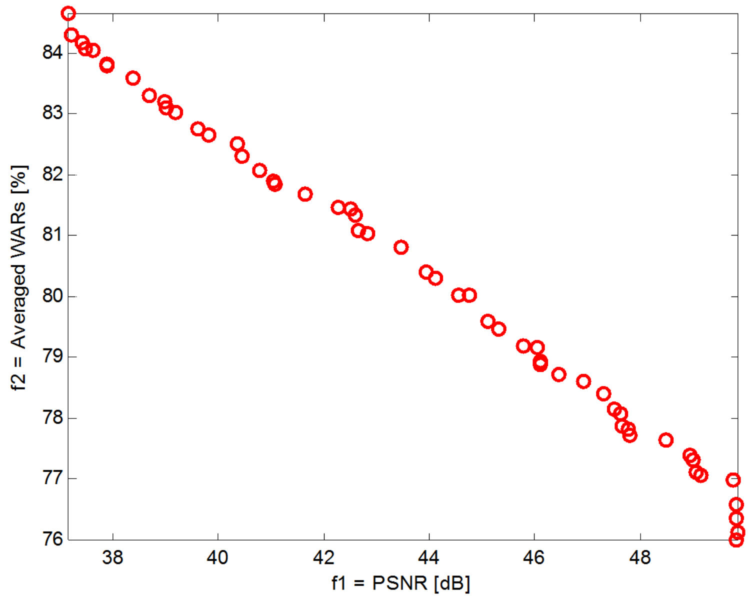


Figure 4.10: The objective space of WAT-AMMOA's obtained Pareto optimal solutions for Lena colour image.

between the original images and the watermarked images are difficult to observe by human eyes. The PSNRs obtained by WAT-AMMOA for all these four colour test images are compared with PSNRs obtained by WAT-GRNN, Yu's method, and Kutter's method. The comparison results are described in Table 4.2.

However, the scalar measure PSNR is not sufficient for human visual perception, since: (i) it considers only energy differences between pixels; (ii) it ignores the perceptual nature of edges and textures of the images. Hence, we also employ the *Relative Rényi Fractal Dimension Spectrums* (RRFDS) introduced by Kinsner and Dansereau [KiDa06], to measure the imperceptibility of the watermark regarding to the HVS. The approach of RRFDS for gray-scale images is described in [KiDa06]. In our application, we generalize

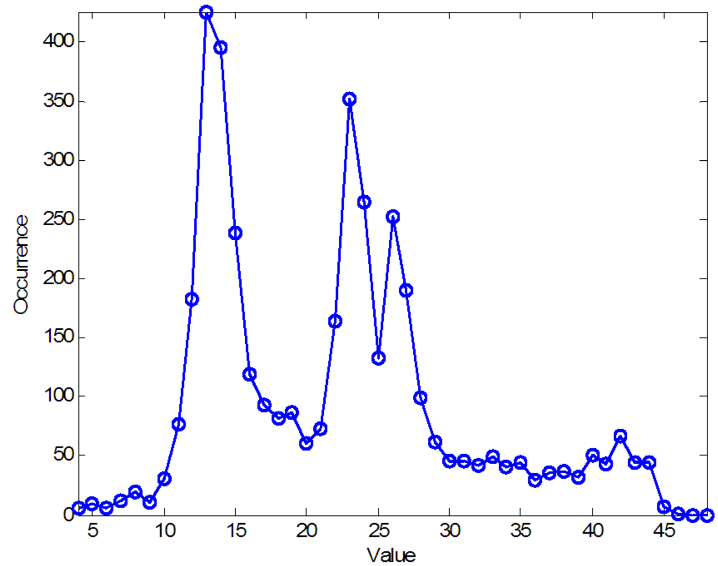


Figure 4.11: The obtained watermarking factors for Lena colour image.

this measure for color images as follows.

The finite probability distribution X_l of the watermarked image (or Y_l of the original image) at the l^{th} scale s is obtained from

$$x_{jl} = \frac{\sum_{\forall i,j \in b_{jl}} \sum_{k=1}^{3} (I(i,j,k)+1)}{\sum_{\forall i,j \in B_{jl}} \sum_{k=1}^{3} (I(i,j,k)+1)} \tag{4.16}$$

where x_{jl} is the probability of a volume element (vel for short) b_{jl}, B_l is the non-overlapping covering (union of all the vels b_{jl}). The *Rényi relative entropy*, H_{Rq}, is given by

$$H_{Rq}(X||Y) = \frac{1}{q-1} \log \frac{\sum_{j=1}^{N} x_j \left(\frac{x_j}{y_j}\right)^q}{\sum_{j=1}^{N} x_j}; -\infty \leq q \leq \infty \tag{4.17}$$

The relative Rényi fractal dimension spectrum, D_{Rq}, at all l^{th} scale of s is then defined as

$$D_{Rq}(X_l||Y_l) = \lim_{s \to 0} \frac{H_{Rq}(X_l||Y_l)}{log(s_l)} \tag{4.18}$$

From Eq. (4.17) and Eq. (4.18), we can see that if $X_l = Y_l$, $D_{Rq} = 0$. This means that the watermarked image is perceived as the original image if the spectrum D_{Rq} of

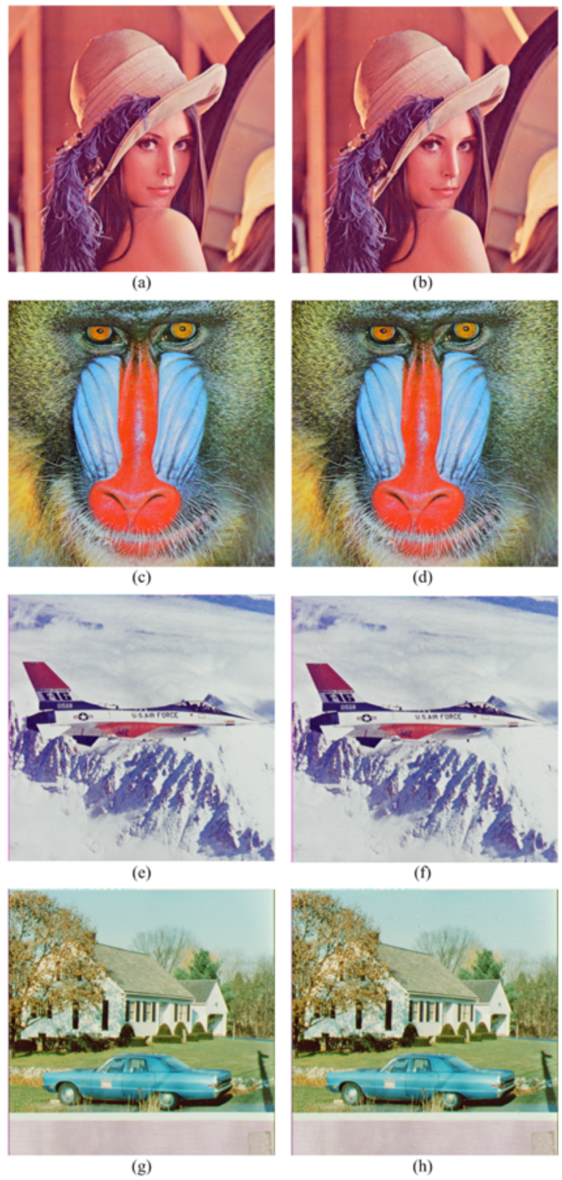


Figure 4.12: The original test images and watermarked test images: (a) original Lena image, (b) watermarked lena image with the obtained PSNR=42.82 dB, (c) original Baboon image, (d) watermarked Baboon image with the obtained PSNR=42.43 dB, (e) original Airplane F16 image, (f) watermarked Airplane F16 image with the obtained PSNR=42.80 dB, (g) original House image, (h) watermarked House image withe the obtained PSNR=42.86 dB.

Table 4.2: PSNR comparison of watermarked images

Images	PSNR [dB]			
	Kutter's	Yu's	WAT-GRNN	WAT-AMMOA
Lena	41.8433	41.6670	42.4590	42.8180
Baboon	41.3612	41.2206	42.5781	42.4320
Airplane	38.6961	38.5295	42.3353	42.8027
House	39.4374	39.2806	42.3143	42.8596

watermarked image is close to zero. With $q = 0$ then $D_{Rq} = 0$, and with $q = 1$ then H_{Rq} is precisely the Kullback-Leibler distance. The spectrum, D_{Rq}, for watermarked images of Lena colourr image obtained by WAT-AMMOA, WAT-GRNN, Yu's method, and Kutter's method are depicted in Fig. 4.13.

In Fig. 4.13, we evaluate the multifractal spectrum in the range of q of $[-20, 20]$. It is seen that the spectrum for the watermarked images of Lena with WAT-GRNN and WAT-AMMOA are both very small (close to zero). This means that the proposed algorithms gain good transparency in embedding the watermarks. The spectra in Fig. 4.13 also tell us that the WAT-AMMOA method is better in gaining the imperceptibility of watermark than the other methods (WAT-GRNN, Yu's, and Kutter's method) that PSNRs in Table 4.2 do not.

4.4.3 Robustness Evaluation

The robustness of the watermark is evaluated by the similarity between the extracted watermark and the original watermark through WAR computed by Eq. (4.13). The watermarks extracted from the watermarked images in Fig. 4.12 are shown in Fig. 4.14. The calculated WARs indicate that our method perfectly extracts watermarks

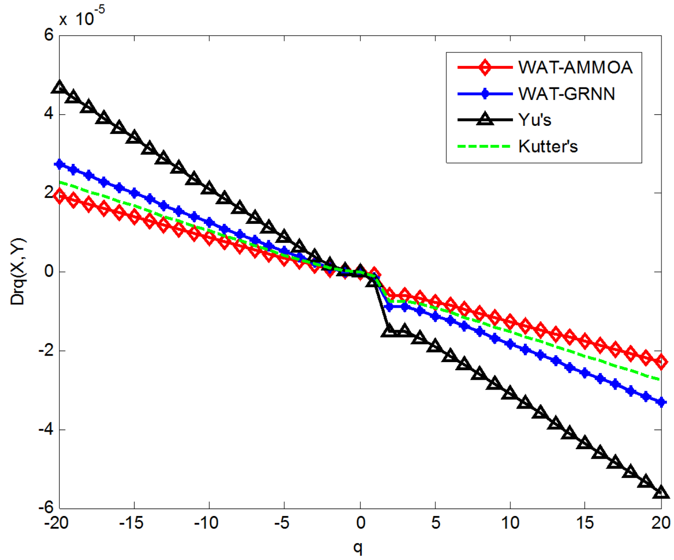


Figure 4.13: Relative Rényi multifractal dimension spectrum for watermarked images of Lena obtained by WAT-AMMOA, WAT-GRNN, Yu's, and Kutter's algorithms.

from watermarked images in the case of without any attacks.

We test the proposed algorithm with five different classes of attacks such as (*i*) compression attacks (JPEG compression), (*ii*) noise addition attacks (AWGN, salt & pepper, and fractional noises), (*iii*) filtering attacks (median filtering), (*iv*) amplitude scaling attacks, (*v*) and geometric manipulation attacks (image cropping, and rotation).

1. *Robustness Against JPEG Compression*: JPEG is a common image compression standard for multimedia application. Hence, watermarking systems should be robust to this attack. Fig. 4.15 shows an example of JPEG compression attack with the quality factor of 40 to the watermarked images of Lena and Baboom, and the proportional extracted watermarks. The robustness comparison with WAT-GRNN, Yu's and Kutter's methods for the watermarked image of Lena in Fig. 4.12 is displayed in Fig. 4.16.

2. *Robustness Against Amplitude Scaling*: The colour values of the watermarked image are divided by a *scaling factor* (SF). The attack is called negative amplitude scaling attack

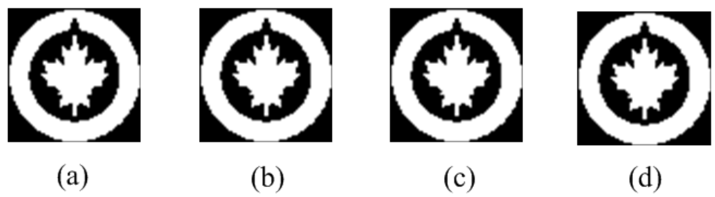


Figure 4.14: Watermarks extracted from watermarked images in Fig. 4.12: (a) extracted from Fig. 4.12(b) with WAR=100 %, (b) extracted from Fig. 4.12(d) with WAR=100 %, (c) extracted from Fig. 4.12(f) with WAR=100 %, (d) extracted from Fig. 4.12(h) with WAR=100 %.

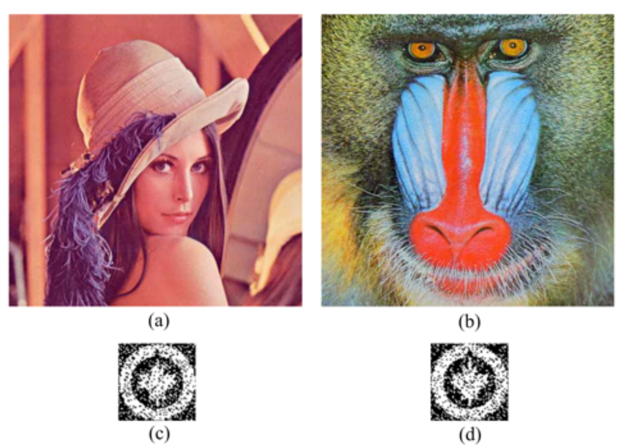


Figure 4.15: An example of JPEG compression attack and watermark extraction with JPEG quality factor of 40: (a) compression of watermarked image of Lena at Fig. 4.12(b) with SNR=26.14 dB, (b) compression of watermarked image of Baboom at Fig. 4.12(d) with SNR=18.98 dB, (c) the extracted watermark from (a) with WAR=82.47 %, (d) the extracted watermark from (b) with WAR=83.42 %.

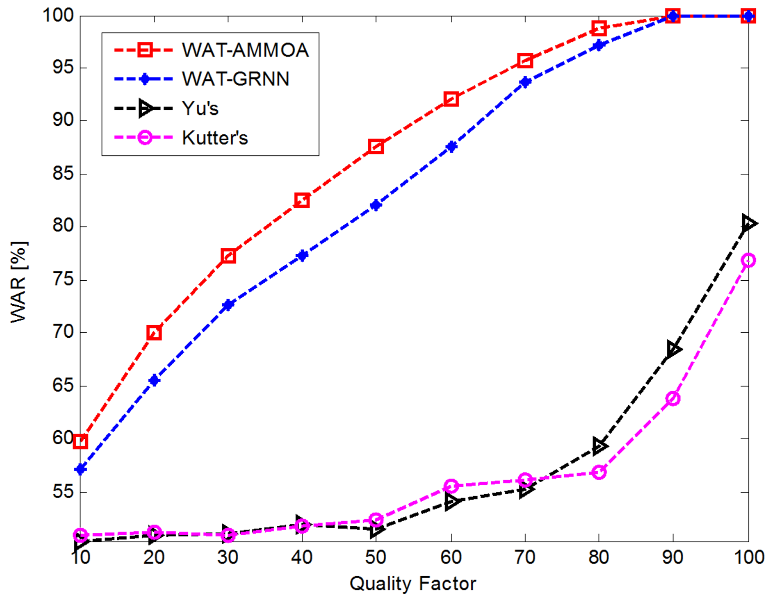


Figure 4.16: The experimental results under the JPEG compression attack for watermarked image of Lena.

if SF is greater than one, and vice versa is the positive amplitude scaling attack. An example of the positive amplitude scaling attack with SF$= 0.3$ for watermarked images of Lena and Baboom in Fig. 4.12 are depicted in Fig. 4.17. The robustness of watermark compared with results from WAT-GNRR, Yu's and Kutter's methods is illustrated in Fig. 4.18.

It can be seen that the WAT-AMMOA algorithm is very robust to amplitude scaling attacks. Even if with the positive attack of SF=0.3 that decreases the SNR of the attacked watermarked image to -7.36 dB, we are still able to recover the watermark excellently.

3. *Robustness Against Additive White Gaussian Noise*: Since the natural features of electronic devices and communications channels, AWGN is perhaps the most common noise in communications systems. Thus, a good watermarking scheme should be robust to AWGN. The robustness fo our scheme against AWGN is shown in Fig. 4.19 and Fig.

Figure 4.17: An example of amplitude scaling attack and watermark extraction with SF=0.3: (a) scaling the watermarked image of Lena at Fig. 4.12(b) with SNR=-7.36 dB, (b) scaling the watermarked image of Baboom at Fig. 4.12(d) with SNR=-7.36 dB, (c) the extracted watermark from (a) with WAR=98.09 %, (d) the extracted watermark from (b) with WAR=90.09 %.

4.20.

The AWGN is added to the watermarked images with different standard deviation σ_n (corresponding SNRs). The Gaussian noise is added to the colour image of watermarked image, I_W, by

$$I_W^N = I_W + \sigma_n N \tag{4.19}$$

where N is the normally distributed random noise, and I_W^N is the watermarked image corrupted by the Gaussian noise. The proposed method works really well, even with a variance of AWGN=40^2 (with the equivalent SNR around 10 dB). This level is a challenge to every watermarking and denoising techniques [Dono95, Kins02].

4. *Robustness Against Salt & Pepper noise*: Salt & Pepper noise is a common type of impulse noises. It is caused by faulty camera sensors or transmission in noisy communications channels with memory. When the image is corrupted by this noise, the

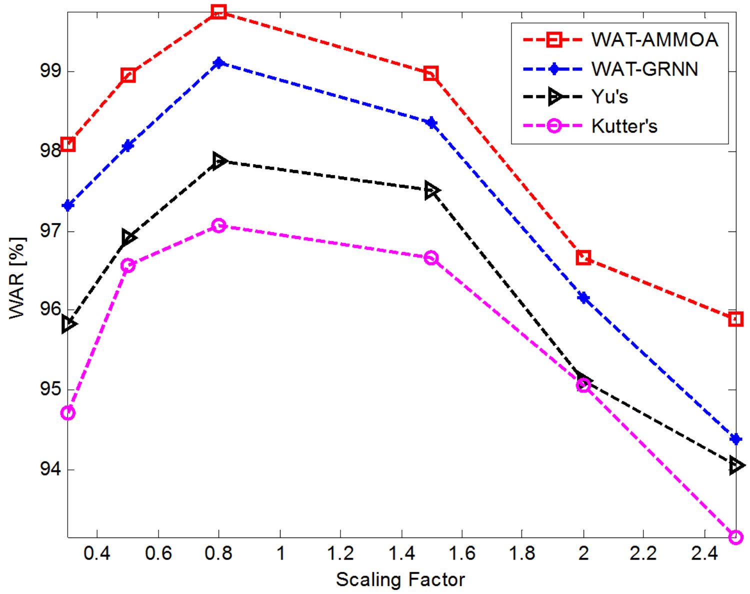


Figure 4.18: The experimental results under the amplitude scaling attack for watermarked image of Lena.

Figure 4.19: An example of AWGN noise attack and watermark extraction with variance of AWGN= 40^2: (a) attacked watermarked image of Lena at Fig. 4.12(b) with SNR= 10.9 dB, (b) attacked watermarked image of Baboom at Fig. 4.12(d) with SNR=10.74 dB, (c) the extracted watermark from (a) with WAR=75.34 %, (d) the extracted watermark from (b) with WAR=71.73 %.

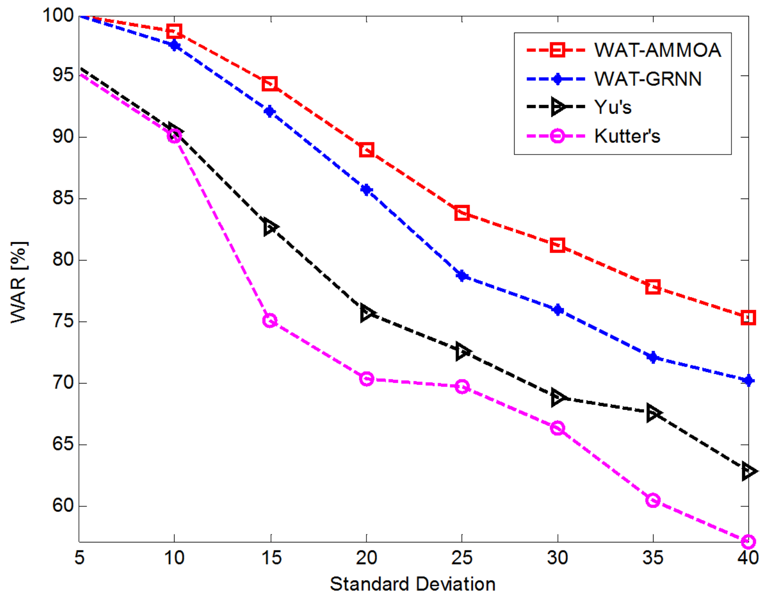


Figure 4.20: The experimental results under the AGWN noise attack for watermarked image of Lena.

noisy pixels have only maximum or minimum values in the color range. Thus, this noise degrades the image considerably. Because of these features, salt & pepper noise is a challenge for denoising and watermarking techniques. The robustness of the proposed method is depicted in Fig. 4.21 and Fig. 4.22.

5. *Robustness Against Fractional Noise*: A fractional noise (coloured noise) has a power spectrum density that decays as $1/f^{\beta}$, where f denotes the frequency and $\beta \geq 0$ [Kins12c, Kasd95, StGB11]. The coloured noise include pink noise ($\beta = 1$), brown noise ($\beta = 2$), and black noise ($\beta = 3$). While white noise ($\beta = 0$) is independent and uncorrelated, the coloured noise may be independent but correlated. Thus, coloured noises are also called correlated noises. Coloured noises have been observed in many different fields such as in electronic devices, musical melodies, astronomy, human cognition system, financial systems, and natural images [Kins12c, StGB11]. In fact, coloured noises exist

Figure 4.21: An example of salt & pepper noise attack and watermark extraction with the density of noise is 0.1: (a) attacked watermarked image of Lena at Fig. 4.12(b) with SNR= 10.04 dB, (b) attacked watermarked image of Baboom at Fig. 4.12(d) with SNR=9.89 dB, (c) the extracted watermark from (a) with WAR=83.11 %, (d) the extracted watermark from (b) with WAR=79.66 %.

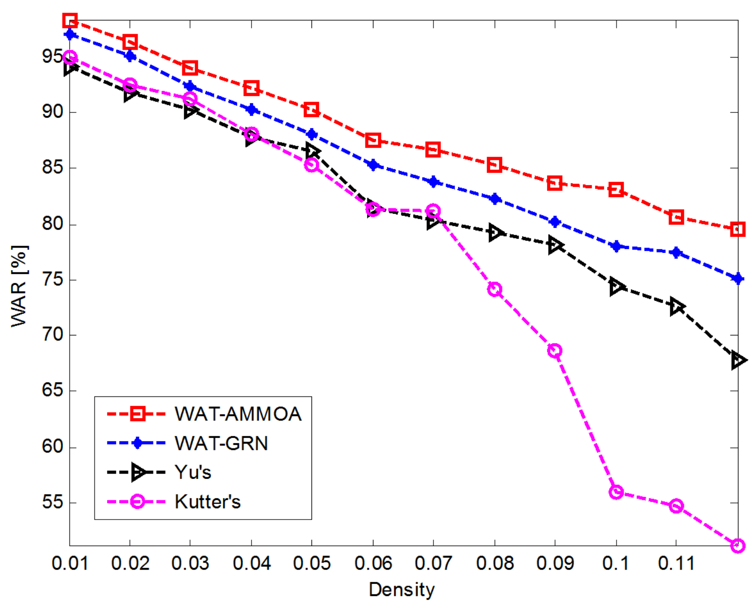


Figure 4.22: The experimental results under the salt & pepper noise attack for watermarked image of Lena.

everywhere throughout the nature. Hence, the study of the effect of coloured noises to watermarking systems is critical.

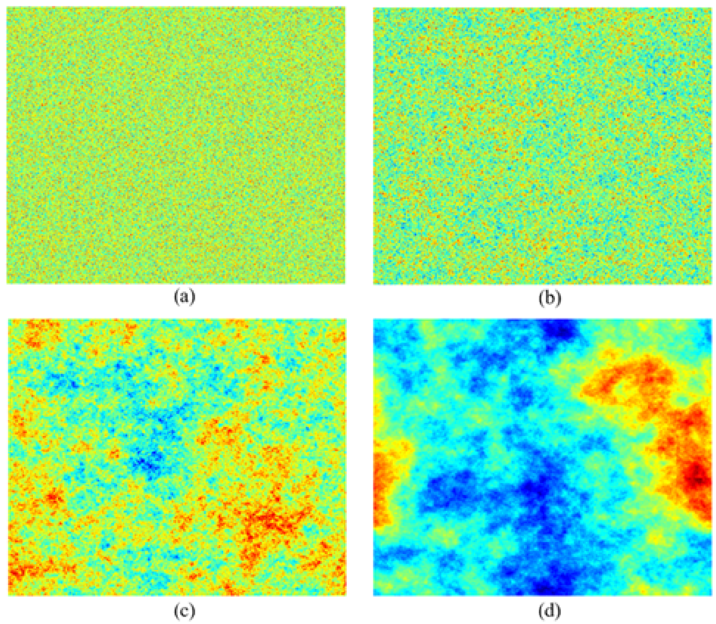


Figure 4.23: Two dimensional coloured noises: (a) white noise; (b) pink noise; (c) brown noise; (d) black noise.

To add the coloured noises to watermarked images, we first synthesize the fractional noise $1/f^{\beta}$. There are five methods to synthesize the fractional noises which are described in [Kins12c]. They are: (*i*) superposition of relaxation processes; (*ii*) fractional integration algorithm; (*iii*) midpoint displacement algorithm; (*iv*) stochastic noise synthesis algorithm; (*v*) spectral filtering algorithm; (*vi*) random cut algorithm; and (*vii*) functional based modeling. In this work, we utilize the spectral filtering method to synthesize coloured noises. The two dimensional synthesized white, pink, brown, and black noises are depicted in Fig. 4.23.

The coloured noises are added to the watermarked images by

$$I_W^{C_N} = I_W + \gamma C_N \tag{4.20}$$

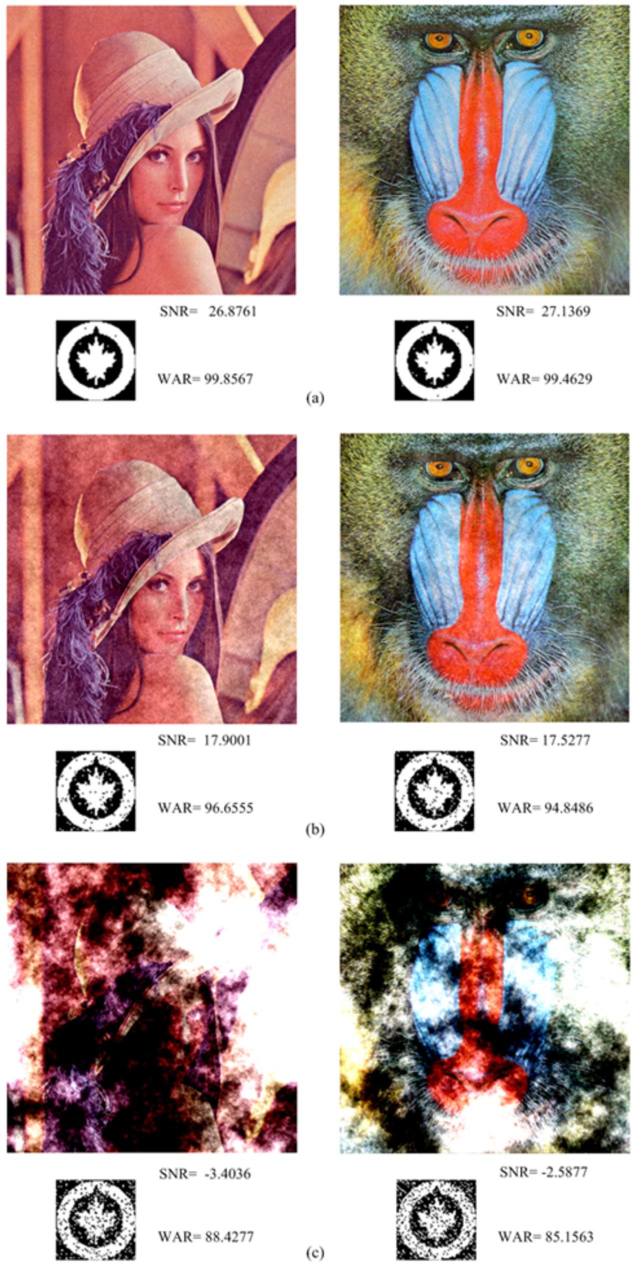


Figure 4.24: The watermarked image of Lena and Baboon in Fig. 4.12 corrupted by coloured noises and the corresponding extracted watermarks: (a) with pink noise; (b) with brown noise; (c) with black noise.

where C_N is the coloured noise, I_W is the watermarked image, $I_W^{C_N}$ is the watermarked image corrupted by the coloured noise, and γ is the noise amplitude scaling factor.

We use two watermarked images of the two most complex structure images Lena and Baboon to do this experiment. The watermarked images corrupted by coloured noises with the same value of γ and the correspoinding extracted watermarks are shown in Fig. 4.24

The images behave differently under the influence of the different coloured noises. With the same amplitude scaling factor $\gamma = 1000$, the pink noise degrades the image the least (with SNR around 27 dB), followed by the brown noise (with SNR $\approx$ 18 dB), and complete alteration by the black noise (with SNR $\approx$ -3 dB). It is seen that the proposed method is extremely robust against the fractional noise. With pink noise, the WARs are around 99.5%. With brown-noise attack, the WARs are around 95%. With the black-noise attack, the algorithm still able to recover the watermark excellently with the WARs around 86%. The comparison with other methods in the case of coloured noise addition attacks is described in Table 4.3.

Table 4.3: Coloured-noise attacks to watermarked images of Lena

Noises	WAR [%]			
	Kutter's	Yu's	WAT-GRNN	WAT-AMMOA
Pink Noise	85.8887	90.1123	98.0957	99.8567
Brown Noise	77.1729	82.8125	88.0615	96.6555
Black Noise	72.8760	60.1807	73.1689	88.4277

6. *Robustness Against Median Filtering*: Median filtering is always a serious challenge to watermarks. This is because a median filter does average pixel values in the window size that eliminates high dynamic values in the image in the spatial domain. Hence,

median filtering can affect the watermark severely. An example of doing median filtering for watermarked images of Lena and Baboon with the filter window size of 5 is displayed in the Fig. 4.25. The robustness comparison of the proposed algorithm with other methods for the watermarked image of Lena is depicted in Fig. 4.26.

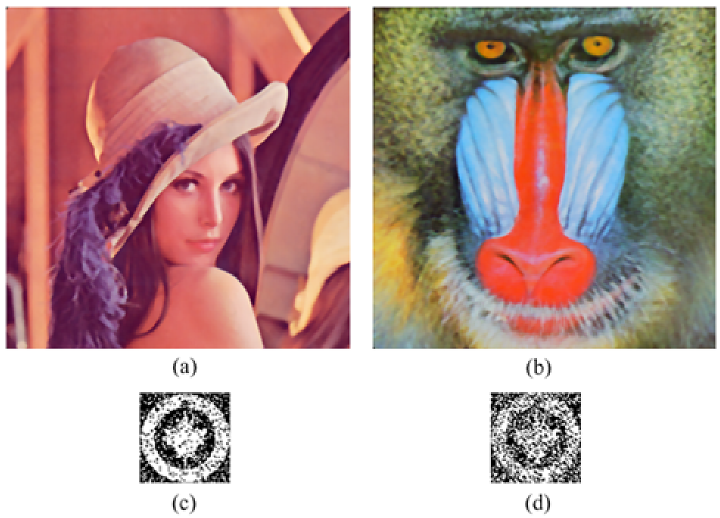


Figure 4.25: Median filtering attack to the watermarked images with filter window = 5: (a) filtered watermarked Lena image with SNR=26.87dB; (b) filtered watermarked Baboon image with SNR=17.43 dB; (c) watermark extracted from (a) with WAR=84.45 %; (d) watermark extracted from (b) with WAR=70.73 %.

7. *Robustness Against Image Cropping*: Image cropping is a class of geometric manipulation attacks which are very common in practice. This attack usually degrades the image severely because it leads to the loss of so much information. If the data hiding algorithm embeds the watermark in only a local area of the image, it becomes very vulnerable to this attack. In this experiment, we remove a part of the watermarked image with different levels (from 10% to 60%). An example of cropping 50% of its surrounding is displayed in Fig. 4.27. The robustness comparison of the proposed algorithm with other methods for the watermarked image of Lena under surrounding cropping attacks is depicted in Fig 4.28.

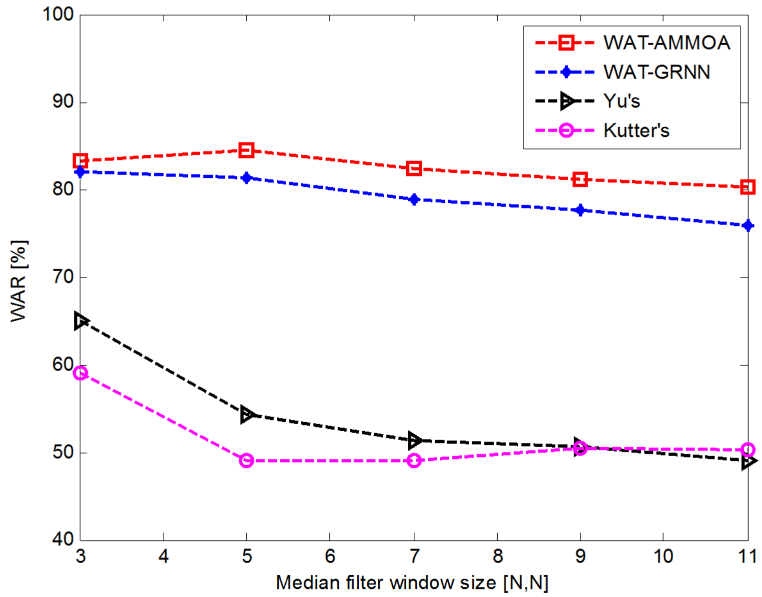


Figure 4.26: Experimental results under median filtering attacks for the watermarked images of Lena.

Figure 4.27: Cropping 50% of the watermarked images of Lena and Baboon and extracted watermarks: (a) cropping the watermarked Lena image with SNR=2.86 dB; (b) cropping the watermarked Baboon image with SNR=3.63 dB; (c) watermark extracted from (a) with WAR=76.35 %; (d) watermark extracted from (b) with WAR=75.29 %.

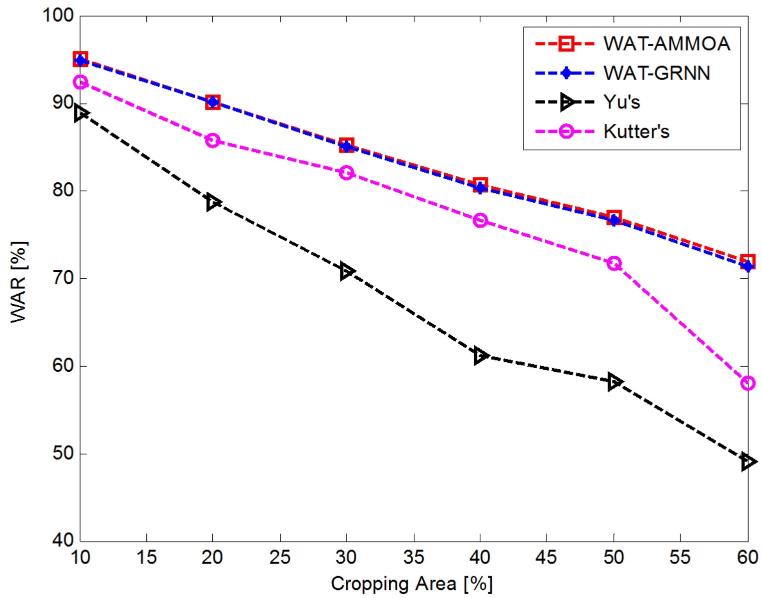


Figure 4.28: Experimental results under surrounding cropping attacks for the watermarked images of Lena.

We also do experiments with block-lost attack, which is another class of cropping geometric manipulation. The results are shown in Fig. 4.29. The results obtained in Fig. 4.29 are very impressive. They show that our proposed algorithm has excellent robustness against block-lost attacks. Even if the image losses 50% information, we are still able to recover the watermark.

8. *Robustness Against Rotation Attacks*: Image rotation is a class of geometric transformation. Since the synchronization between the extracted watermark and the embedded watermark is lost after applying rotation, most watermarking extraction algorithms are unable to detect and extract the watermark. In this experiment, rotation is varying from 0.1^0 to 1^0 clockwise. After rotation, the rotated image is cropped the four conners to keep the size of the rotated image as the same size of the watermarked one. The results in Fig. 4.30 show that the proposed algorithm has little robust to this class of attack.

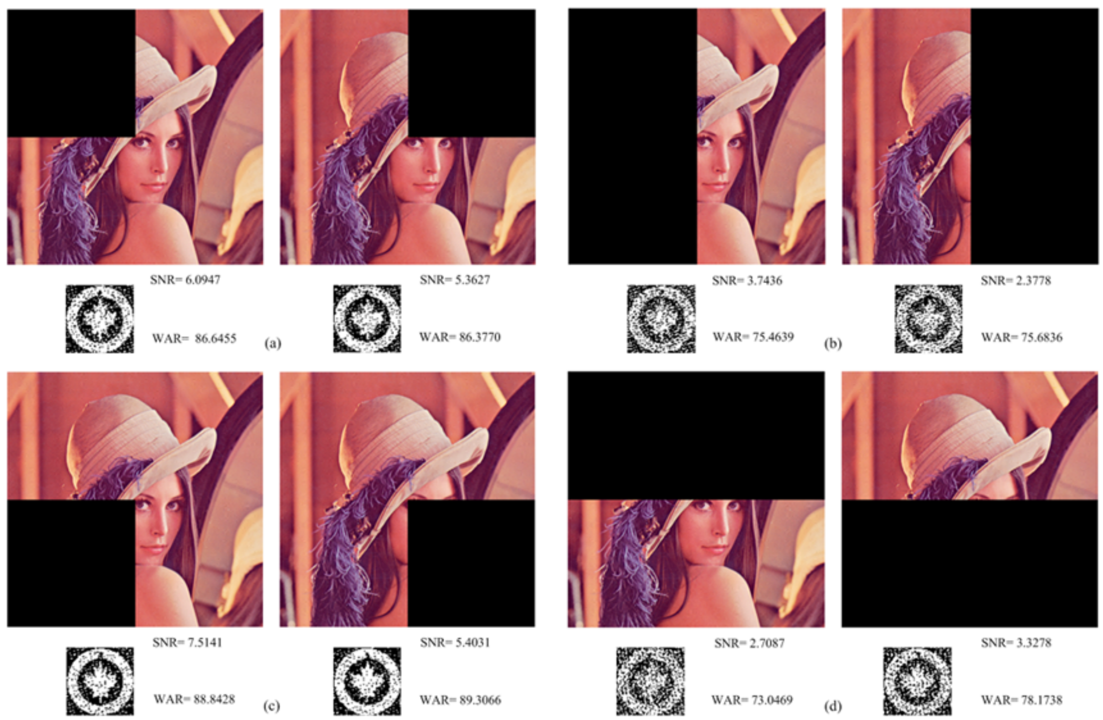


Figure 4.29: Experimental results under block-lost attacks for the watermarked image of Lena: (a) 25 % lost; (b) 50% lost; (c) 25% lost; (d) 50 % lost.

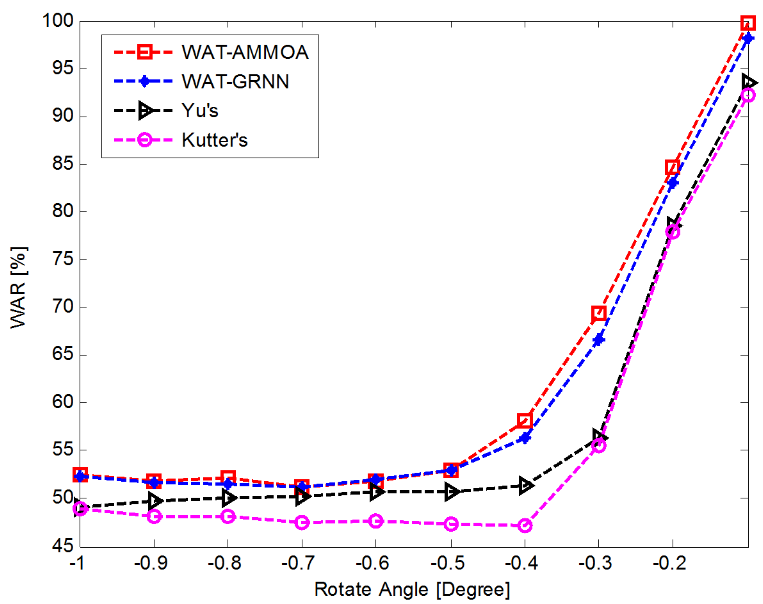


Figure 4.30: Experimental results under rotation attacks for the watermarked image of Lena.

4.5 Summary of Chapter 4

In this paper, a logo watermarking for colour images is formulated as a multiobjective optimization problem of finding the watermarking parameters to maximize the quality of watermarked image and the robustness of the watermark under different attacks. A novel intelligent and robust logo watermarking method based on the general regression neural networks and multiobjective memetic algorithms is proposed to solve this challenging problem. Specifically, the embedding factors and the smooth parameter of the GRNN are searched optimally by the adaptive multiobjective memetic optimization algorithm (AMMOA) to maximize the PSNR and the averaged WARs objectives. The proposed algorithm obtains better results in transparency and robustnesses against classes of additive noise, signal processing, and geometric transformation attacks than previous approaches.

We discuss the application of neural networks for watermarking systems. We evaluated neural networks, and selected GRNN for its good fit to our problem. The GRNN is much superior over the BPNN when solving this problem as it has very fast time convergence and high prediction accuracy. We also uses the relative Rényi fractal dimension spectrum to evaluate the quality of watermarked images with remarkable results.

However, the proposed algorithm has its own disadvantages and needs further improvements. For example, since it needs a sufficient time for the evolutionary and local refining searches to find the best local and global solutions, it is not fast enough for the real-time applications at this stage.

Chapter 5

Multiobjective Joint Spectrum Sensing and Power Control in Cognitive Radio Networks

The chapter deals with the problem of joint spectrum sensing and power control optimization for a multichannel, multiple-user cognitive radio network. In particular, we investigate trade-off factors in designing efficient spectrum sensing techniques to maximize the throughputs and minimize the interference. To maximize the throughputs of secondary users and minimize the interference to primary users, it requires for a joint determination of the sensing and transmission parameters of the secondary users, such as sensing times, decision threshold vectors, and power allocation vectors. There are conflicts between these two objectives, thus a multiobjective optimization problem is introduced. We propose to use a memetic learning algorithm to solve this multiobjective joint optimization problem. The algorithm evolutionarily learns to find optimal spectrum sensing times, decision threshold vectors, and power allocation vectors to maximize the averaged opportunistic throughput and minimize the averaged interference to the cognitive network.

5.1 Introduction

Cognitive radios have been proposed to be the next generation wireless devices that can share underutilized spectrum [MiMa99, Hayk05, HoBh07]. Spectrum sensing and dynamic spectrum access (e.g., dynamic channel selection) are the important principles of cognitive radios [HoBh07, JZOG15]. In spectrum sensing, cognitive radio users (*secondary users* - SU) sense the spectrum of licensed users (*primary users* -PU) to detect and utilize spectrum holes within the PUs' spectrum. The cognitive radio networks adopt a hierarchical access structure by considering PUs as the legacy spectrum holders and SUs as the unlicensed users.

The challenge for a reliable sensing algorithm is to identify suitable transmission opportunities without compromising the integrity of the PUs [PaSc13, AlSt12]. The efficiency of the employed spectrum sensing technique plays a key role in maximizing the cognitive radio network throughput, while protecting the PUs from interference. The popular criteria in designing sensing techniques is to minimize the probability of false alarm as low as possible [AlSt12, PaSc13]. In addition, in order to limit the probability of interfering with PUs, it is desirable to keep the missed detection probability as low as possible. The sensing time is the trade-off factor between the quality and the speed of sensing. Increasing the sensing times allows to have both low false alarm and low missed detection probabilities, but reduces the time available for transmissions which results in low throughputs of SUs. The throughput of an SU depends on the transmission energy (i.e., the transmitted power integrated over a transmission time). Thus, together with the sensing times it also requires to consider the allocation of powers on channels of SUs to maximize the throughput and minimize the interference created by SUs. Another trade-off factor between the false alarm and the missed detection probabilities is the detection thresholds. Low detection thresholds result in high false alarm probability and low missed detection probability and vice versa. Thus, to maximize the throughput of SUs, it requires for a joint optimization of the sensing and transmission parameters of the SUs. They are sensing times, decision threshold vectors, and power allocation vectors.

The joint spectrum sensing design and power control for a single-channel point-to-point cognitive radio network has been introduced in [AlSt12]. In that work, the authors formulated the joint sensing-duration design and power control problem as a two-stage stochastic program with recourse. The numerical results show that the method obtains good achievable throughput for the SU. However, only the sensing time parameter in the spectrum sensing process is considered as an optimization variable while the decision threshold is prefixed. Furthermore, because of using two stages stochastic programming in which the first stage is for finding the optimal sensing time and the next stage is for finding the optimal power, the obtained throughput is not optimal.

The game theory approach for the joint sensing and power allocation optimization problem of a multiple channels, multiple-SU cognitive radio network has been proposed by Pang and Scutari [PaSc13]. In this work, a novel class of Nash problems where each SU considered as a player competes against other SUs to maximize his own opportunistic throughput by choosing jointly the sensing duration, the detection thresholds, and the power allocation vector over multichannel link has been introduced. Several constraints included interference constraints, probability constraints, and power budget constraints have been used to setup the game. The resulting players' optimization problems are so non-convex that is challenging to solve in the traditional game theory. To deal with the non-convexity of the game, the Quasi-Nash Equilibrium is proposed to obtain a considerable performance. However, using too many constraints makes the joint optimization problem less dynamic to obtain a global optimality.

In this chapter, we model the joint optimization problem between spectrum sensing and power allocation for a multichannel, multiple user cognitive radio network as a multiobjective optimization problem. Two conflicting objectives are the throughput of SUs and the interference created by SUs [DaKi14b, DaKi15b]. We propose to use AMMOA to search for optimal sensing times, decision threshold vector, and power allocation vector of each SU to maximize the averaged throughput and minimize the averaged interference of the cognitive network. The main contributions of this work are as follows.

1. A multiobjective joint optimization problem between spectrum sensing and power allocation for a multichannel multiple SU cognitive radio network is introduced; and

2. A novel multiobjective joint spectrum sensing and power allocation method is proposed based on AMMOA. This is the first approach to solve the joint optimization problem using multiobjective optimization.

The rest of the chapter is organized as follows. Sec. 5.2 describes the system model and joint spectrum sensing and power control problem. Sec. 5.3 reports the proposed multiobjective joint spectrum sensing and power control based on AMMOA. Sec. 5.4 discusses experimental results. Finally, the chapter ends with the concluding remarks.

5.2 System Model and Problem Formulation

The model is considered with N_Q active SUs, each formed by a transmitter-receiver pair, coexisting in the same area and sharing the same band. We assume that the *medium access control* (MAC) frame is divided in two time slots: τ-sensing slot, and $T-\tau$ data slot as shown in Fig. 5.1. During the sensing slot τ, the SUs stay silent and sense the electromagnetic environment to look for the spectrum holes. During the data slot $T-\tau$, the SUs transmit simultaneously over the portions of the licensed spectrum detected as available. The sensing problem is introduced in a cognitive radio scenario that consists of one active PU.

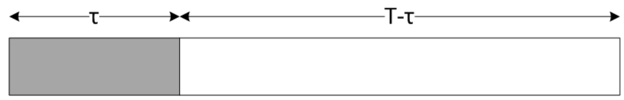


Figure 5.1: Cognitive radio time-slot allocation.

The spectrum sensing problem of SU $i = 1, ..., Q$ on subcarrier $k = 1, ...N$ is formulated as the following binary hypothesis testing at time index $n = 1, 2, ..., K_i$,

$$\begin{cases} \mathcal{H}_{k|0} : y_{i,k}[n] = w_{i,k}[n], \\ \mathcal{H}_{k|1} : y_{i,k}[n] = S_{i,k}[n] + w_{i,k}[n], \end{cases} \tag{5.1}$$

where the hypothesis $\mathcal{H}_{k|0}$ represents the absence of the primary signal over the subcarrier k; $\mathcal{H}_{k|1}$ represents the presence of at least one PU; $w_{i,k}$ is the additive background noise as Gaussian process with zero mean and variance $\sigma^2_{i,k}$; $S_{i,k}[n]$ is the primary signaling like a stationary random process with zero mean and variance $\sigma^2_{S_{i,k}}$; $K_i = \tau_i f_i$ is the number of samples with τ_i is the sensing time, and f_i is the sampling frequency.

Based on the Neyman-Pearson frame work, the decision rule of SU i over carrier k based on the energy detector is [TaSa08, PaSc13]

$$D(Y_{i,k}) = \frac{1}{K_i} \sum_{n=1}^{K_i} |y_{i,k}[n]|^2 \underset{\mathcal{H}_{k|0}}{\overset{\mathcal{H}_{k|1}}{\gtrless}} \gamma_{i,k} \tag{5.2}$$

where $\gamma_{i,k}$ is the decision threshold of SU i for the carrier k to be chosen to meet the required false alarm rate. Based on the central limit theorem, the random variable $D(Y_{i,k})$ can be approximated for sufficient large K_i by a Gaussian distribution.

$$D(Y_{i,k})|\mathcal{H}_{k|l} \sim \mathcal{N}\left(\mu_{i,k|l}, \sigma^2_{i,k|l}/K_i\right) \tag{5.3}$$

where $l = 0, 1$;

$$\mu_{i,k|l} = \begin{cases} \sigma_{i,k}^2, & \text{if } l = 0 \\ \sigma_{S_{i,k}}^2 + \sigma_{i,k}^2, & \text{if } l = 1 \end{cases} \tag{5.4}$$

$$\sigma_{i,k|l}^2 = \begin{cases} \sigma_{i,k}^4, & \text{if } l = 0 \\ E|S_{i,k}|^4 + 2\sigma_{i,k}^2 - \left(\sigma_{S_{i,k}}^2 - \sigma_{i,k}^2\right)^2, & \text{if } l = 1 \end{cases} \tag{5.5}$$

For an easy analysis and computation, in this work it is assumed that the primary signaling is Gaussian, the parameter $\sigma_{i,k|l}^2$ is then calculated by

$$\sigma_{i,k|l}^2 = \begin{cases} \sigma_{i,k}^4, & \text{if } l = 0 \\ \left(\sigma_{S_{i,k}}^2 + \sigma_{i,k}^2\right)^2, & \text{if } l = 1 \end{cases} \tag{5.6}$$

Under this framework, the probability of false alarm and probability of detection are approximated as follows

$$P_{i,k}^{fa}(\gamma_{i,k}, \tau_i) = \mathcal{Q}\left(\sqrt{\tau_i f_i}\frac{\gamma_{i,k} - \mu_{i,k|0}}{\sigma_{i,k|0}}\right) \tag{5.7}$$

$$P_{i,k}^{d}(\gamma_{i,k}, \tau_i) = \mathcal{Q}\left(\sqrt{\tau_i f_i}\frac{\gamma_{i,k} - \mu_{i,k|1}}{\sigma_{i,k|1}}\right) \tag{5.8}$$

where the function $\mathcal{Q}(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$.

The missed detection probability is then calculated by

$$P_{i,k}^{md} = 1 - P_{i,k}^{d} \tag{5.9}$$

The sensing problem is to find optimal values of the detection thresholds $\gamma_{i,k}$ and the sensing time τ_i in order to minimize both $P_{i,k}^{fa}$ and $P_{i,k}^{md}$. However, Eqs. (5.7) and (5.8) show that there exists a trade-off between probability of false alarm and the missed detection probability when selecting the optimal values of $\gamma_{i,k}$ and τ_i. Thus, the optimality of the sensing system cannot be obtained if we just focus on the detection problem to select the parameters $\gamma_{i,k}$ and τ_i. The optimal choice of the detection thresholds and sensing time should be the result of a joint optimization of the sensing and transmission processes.

The transmission strategy of each SU i is the power allocation vector $p_i = \{p_{i,k}\}_{k=1}^N$ over the N subcarriers, subject to upper and lower bounds constraints. Given the power allocation profile of the SUs, the detection thresholds, and sensing time, the opportunistic throughput of SU i is given by [PaSc13]

$$R_i(\tau_i, \boldsymbol{p}, \boldsymbol{\gamma}_i) = (1 - \frac{\tau_i}{T}) \sum_{k=1}^{N} [1 - P_{i,k}^{fa}(\gamma_{i,k}, \tau_i)] r_{i,k}(\boldsymbol{p}) \tag{5.10}$$

where the maximum achievable rate $r_{i,k}(\boldsymbol{p})$ for a specific power allocation profile $p_{1,k}, ..., p_{Q,k}$ is

$$r_{i,k}(\boldsymbol{p}) = log \left(1 + \frac{|H_{ii}(k)|^2 p_{i,k}}{\sigma_{i,k}^2 + \sum_{r \neq i} |H_{ri}(k)|^2 p_{r,k}} \right) \tag{5.11}$$

where $\{H_{ii}(k)\}_{k=1}^N$ is the channel transfer function of the direct link i and $\{H_{ri}(k)\}_{k=1}^N$ is the cross-channel transfer function between the secondary transmitter r and the secondary receiver i.

Missed detections at the SUs produces the interferences to the PU. We define the interference created by the cognitive user i to the PU given by

$$I_i^{SU}(\tau_i, \boldsymbol{\gamma}_i, \boldsymbol{p}_i) = \sum_{k=1}^{N_c} P_{i,k}^{md}(\gamma_{i,k}, \tau_i).p_{i,k} \tag{5.12}$$

Our objective is to find the optimal sensing time τ_i, the decision threshold vector $\boldsymbol{\gamma}_i$, and the power allocation vector $\boldsymbol{p}_i$ to maximize the throughput R_i and to minimize the interference I_i at each SU i, $i = 1, ..., N_Q$. Thus, the optimization problem can be defined as

$$\underset{\tau_i, \boldsymbol{\gamma}_i, \boldsymbol{p}_i}{\text{maximize}} \; R_i(\tau_i, \boldsymbol{\gamma}_i, \boldsymbol{p}), \text{ and } \underset{\tau_i, \boldsymbol{\gamma}_i, \boldsymbol{p}_i}{\text{minimize}} \; I_i^{SU}(\tau_i, \boldsymbol{\gamma}_i, \boldsymbol{p}_i) \tag{5.13}$$

Motivated from centralized cooperative spectrum sensing methods [YuAr09, TCSH13], a centralized cooperative joint sensing and power allocation mechanism is proposed in this work. The system model is described in the Fig. 5.2. In this centralized cooperative sensing and power allocation, a central unit is used to collect sensing information from cognitive devices, identifies and determines optimal parameters for spectrum sensing and power allocation, and then broadcasts these informations to other cognitive radios. The central unit is also called the fusion center [YuAr09].

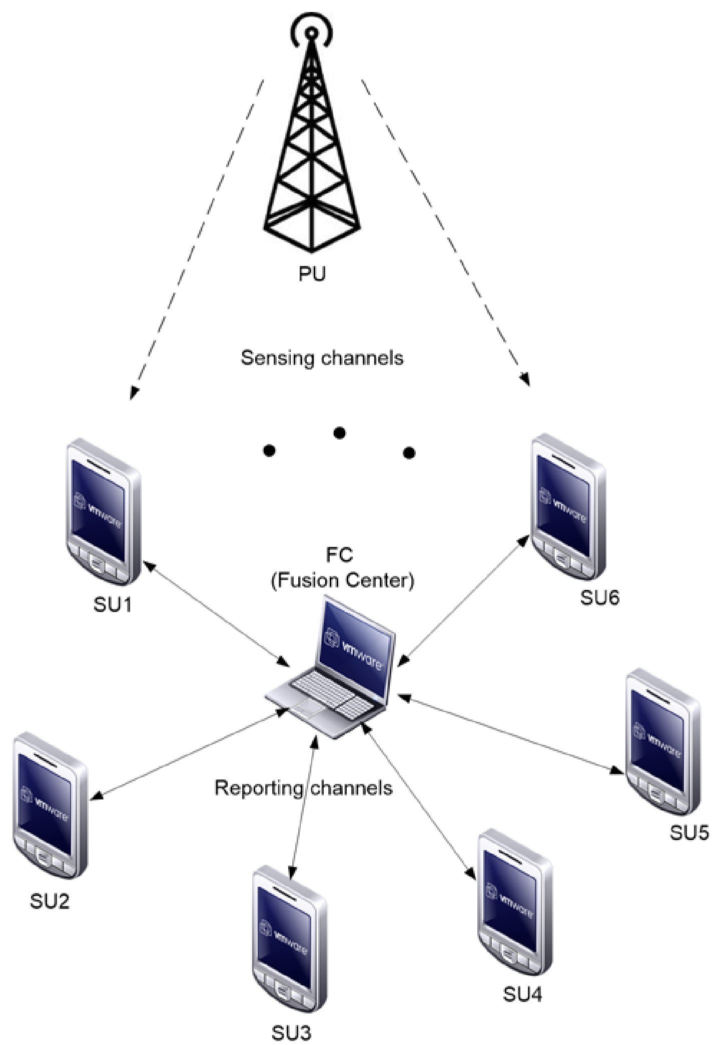


Figure 5.2: Centralized cooperative joint spectrum sensing and power control model.

In this context, SUs send its sensing statistics and parameters to the fusion center. At the fusion center, the probability of false alarm P^{fa} and probability of missed detection

P^{md} are calculated, and so the throughput R_i and the interference I_i^{SU} of SU i. The averaged throughput and the averaged interference of the network are then calculated at the fusion center by

$$R(\boldsymbol{\tau}, \boldsymbol{p}, \boldsymbol{\gamma}) = \frac{1}{N_Q} \sum_{i=1}^{N_Q} R_i(\tau_i, \boldsymbol{p}, \boldsymbol{\gamma}_i) \tag{5.14}$$

$$I^{SU}(\boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{p}) = \frac{1}{N_Q} \sum_{i=1}^{N_Q} I_i^{SU}(\tau_i, \boldsymbol{\gamma}_i, \boldsymbol{p}_i) \tag{5.15}$$

The multiobjective joint optimization problem between spectrum sensing and power control is now formulated by

$$\begin{cases} \underset{\boldsymbol{\tau},\boldsymbol{\gamma},\boldsymbol{p}}{\text{maximize}} \; R(\boldsymbol{\tau}, \boldsymbol{p}, \boldsymbol{\gamma}) \\ \underset{\boldsymbol{\tau},\boldsymbol{\gamma},\boldsymbol{p}}{\text{minimize}} \; I^{SU}(\boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{p}) \end{cases} \tag{5.16}$$

where $\boldsymbol{\tau} = \{\tau_i\}_{i=1}^{N_Q}$; $\boldsymbol{p} = \{p_{i,k}\}_{i=1,k=1}^{N_Q,N_c}$; and $\boldsymbol{\gamma} = \{\gamma_{i,k}\}_{i=1,k=1}^{N_Q,N_c}$.

The problem (5.16) is equivalent to the following multiobjective optimization problem.

$$\begin{cases} \underset{\boldsymbol{\tau},\boldsymbol{\gamma},\boldsymbol{p}}{\text{minimize}} \; f1 = -R(\boldsymbol{\tau}, \boldsymbol{p}, \boldsymbol{\gamma}) \\ \underset{\boldsymbol{\tau},\boldsymbol{\gamma},\boldsymbol{p}}{\text{minimize}} \; f2 = I^{SU}(\boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{p}) \end{cases} \tag{5.17}$$

We propose to use multiobjective memetic learning algorithms to solve the joint optimization in Eq. (5.17). The algorithms are described in the next sections.

5.3 Multiobjective Joint Spectrum Sensing and Power Control Using AMMOA

In this joint spectrum sensing and power control problem, there always exists two conflicting objectives. These are the averaged throughput and the averaged interference of the network. In this work, we apply AMMOA to search for the optimal decision policy (sensing time τ_i, decision threshold vector $\boldsymbol{\gamma}_i$, and power allocation vector $\boldsymbol{p}_i$) for each SU i. The execution flow of our proposed scheme is described in Fig. 5.3. Initially, the population is generated randomly to provide candidate solutions to the problem solver. With each individual (chromosome) in the given population, the algorithm extracts the sensing time vector $\boldsymbol{\tau} = \{\tau_i\}_{i=1}^{Q}$, the decision threshold vectors $\boldsymbol{\gamma} = \{\gamma_{i,k}\}_{i=1,k=1}^{Q,N}$, and the power allocation vectors $\boldsymbol{p} = \{p_{i,k}\}_{i=1,k=1}^{Q,N}$ to combine with the sensing statistics received from SUs ($\boldsymbol{\tau} = \{\mu_{i,k|l}\}_{i=1,k=1}^{Q,N}$, and $\boldsymbol{\sigma}^2 = \{\sigma_{i,k|l}^2\}_{i=1,k=1}^{Q,N}$, where $l = 0, 1$) and the channel transfer parameters $\boldsymbol{H} = \{H_{ii}(k)\}_{k=1}^{N}$ to calculate the averaged throughput R and the averaged interference I^{SU} by Eqs. (5.14) and (5.15), respectively. The parameters R and I are then fit to the objective evaluation to setup the objective vector $\bar{f} = f1 = -R; f2 = I^{SU}$ for AMMOA. AMMOA searches for a better population $\boldsymbol{P}$ after each iteration. The population $\boldsymbol{P}$ then replaces the initialized population $\boldsymbol{P}_{init}$ for the next iteration. Thus, the population is updated after each iteration. The process continues until reaching the termination criteria. When the algorithm finishes, the best solution or best chromosome is selected from the non-dominated population through the post-processing procedure. Finally, the optimal sensing times $\tilde{\boldsymbol{\tau}}$, the optimal decision threshold vectors $\tilde{\boldsymbol{\gamma}}$, and the optimal power allocation vectors $\tilde{\boldsymbol{p}}$ are broadcasted to SUs. The SUs receive these updated parameters and reconfigure their radio operating system.

The inputs consist of N_{pop} chromosomes in population P, N_Q number of cognitive users, and Nc number of channels. Each chromosome consists of $N_Q + 2 * N_c * N_Q$ genes. The first N_Q variables are N_Q genes which represent for the sensing times of N_Q SUs. The next N_Q variables are $N_Q * N_c$ genes representing for decision threshold vectors of N_Q SUs with N_c channels. The last N_Q variables are $N_Q * N_c$ genes representing for power allocation vector of N_Q SUs with N_c channels. The procedure OBJ_EVAL is used to evaluate objectives for each chromosome in the population. In this work, we search for optimal sensing times, decision threshold vectors, and power allocation

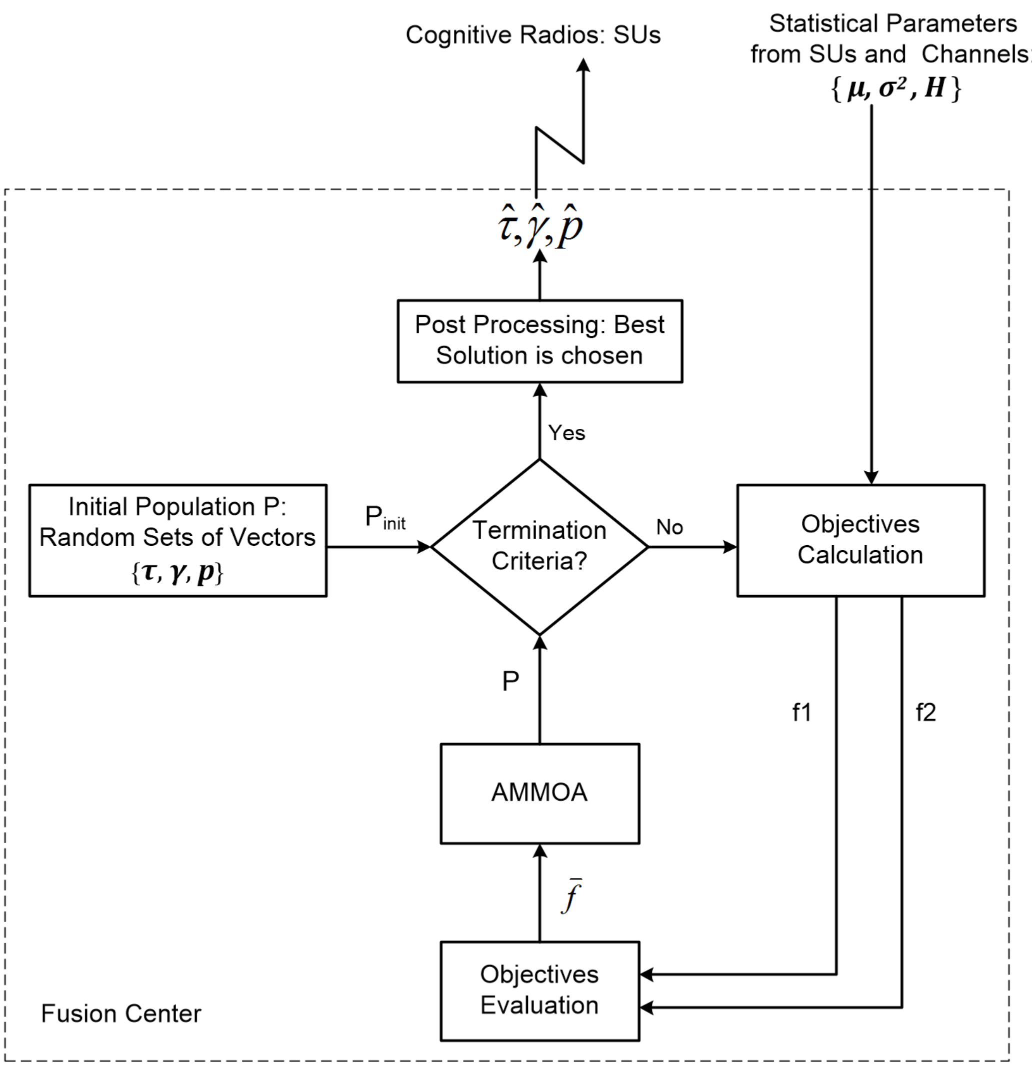


Figure 5.3: Execution flow of the proposed algorithm.

vectors to maximize the averaged throughput of the network R and minimize the averaged interference created by SUs, I^{SU}, defined in Eqs. (5.14) and (5.15), respectively.

The best solution or the best chromosome (S_{best}) is selected from the non dominated population $\boldsymbol{P}$. Finally, the optimal sensing time, the optimal decision threshold vector, the optimal power allocation vector, the resulted throughput, and the resulted interference of each SU are obtained from post processing function POST_PROCESS of the best chromosome. The initialization, and objective evaluation algorithms are discussed as follows.

1) *Initialization*: Each chromosome represents $(1 + 2N_c)N_Q$ real nonnegative parameters to be searched. The first N_Q parameters are sensing time parameters of N_Q SUs, which are searched in the range from τ_{min} to τ_{max}. The next N_QxN_c parameters are N_Q decision threshold vectors (each of size 1xN_c) of N_Q SUs, which are searched in the range from γ_{min} to γ_{max}. The last N_QxN_c parameters are N_Q power allocation vectors (each of size 1xN_c) of N_Q SUs, which are searched in the range from p_{min} to p_{max}.

2) *Objective Evaluation Function*: The objective functions $f1 = -R$ in Eq. (5.14), and $f2 = I^{SU}$ in Eq. (5.15) are the two objectives to be minimized. We denote $\hat{\boldsymbol{\alpha}} = [\bar{\alpha}_1, \bar{\alpha}_2, ..., \bar{\alpha}_{3N_Q}]$ as the parameters to be searched, where $\{\bar{\alpha}_i\}_{i=1}^{N_Q}$ are sensing times of N_Q SUs, $\{\bar{\alpha}_{i,k}\}_{i=N_Q+1,k=1}^{2N_Q,N_c}$ are decision threshold vectors of N_Q SUs for N_c channels, and $\{\bar{\alpha}_{i,k}\}_{i=2N_Q+1,k=1}^{3N_Q,N_c}$ are power allocation vectors of N_Q SUs for N_c channels. The objectives function is then set up as

$$\bar{f}(\hat{\boldsymbol{\alpha}}) = [f_1(\hat{\boldsymbol{\alpha}}), f_2(\hat{\boldsymbol{\alpha}})] \tag{5.18}$$

where

$$f_1(\hat{\boldsymbol{\alpha}}) = -R(\hat{\boldsymbol{\alpha}}) = -R(\bar{\alpha}_1, \bar{\alpha}_2, ..., \bar{\alpha}_{3Q})$$

$$f_2(\hat{\boldsymbol{\alpha}}) = I^{SU}(\hat{\boldsymbol{\alpha}}) = I^{SU}(\bar{\alpha}_1, \bar{\alpha}_2, ..., \bar{\alpha}_{3Q})$$

Our joint optimization problem is to search for the optimal parameter $\hat{\boldsymbol{\alpha}}$ that can be formed by

$$\underset{\hat{\boldsymbol{\alpha}}}{\text{minimize}}\ \bar{f}(\hat{\boldsymbol{\alpha}}) = \underset{\hat{\boldsymbol{\alpha}}}{\text{minimize}}\ [f_1(\hat{\boldsymbol{\alpha}}), f_2(\hat{\boldsymbol{\alpha}})] \tag{5.19}$$

The pseudocode of our objective evaluation function is described in the Algorithm 15.

5.4 Experimental Results and Discussion

In this scope of the paper, we setup the system for simulation with $N_Q = 10$ SUs; the available bandwidth is divided in $N_c = 12$ subchannels; the time sensing frame for each

Algorithm 15 OBJ_EVAL

procedure OBJ_EVAL($Chromosome, N_{var}, N_Q, N_c$)
  for $i \leftarrow 1, N_Q$ **do**
    $\tau(i) \leftarrow Chromosome(i, 1)$
    for $k \leftarrow 1, N_c$ **do**
      $\gamma(i, k) \leftarrow Chromosome(N_Q + i, k)$
      $p(i, k) \leftarrow Chromosome(2N_Q + i, k)$
    end for
  end for
  $[\boldsymbol{H}_{qq}, \boldsymbol{H}_{rq}] \leftarrow$ Channel_tf(N_Q, N_c) ▷ Channel parameters
  for $i \leftarrow 1, N_Q$ **do**
    for $i \leftarrow 1, N_c$ **do**
      Calculate $P_{i,k}^{fa}$ by Eq. (5.7)
      Calculate $P_{i,k}^{de}$ by Eq. (5.8)
      Calculate $P_{i,k}^{md}$ by Eq. (5.9)
      Calculate achievable rate $r_{i,k}$ by Eq. (5.11)
    end for
  end for
  for $i \leftarrow 1, N_Q$ **do**
    Calculate the throughput $R(i)$ by Eq. (5.10)
    Calculate the interference $I^{SU}(i)$ by Eq. (5.12)
  end for
  $f(1) \leftarrow -R$ calculated by Eq. (5.14) ▷ Objective 1
  $f(2) \leftarrow I^{SU}$ calculated by Eq. (5.15) ▷ Objective 2
  return f
end procedure

SU is $Tf = 0.01$ s; minimum sensing time for each SU for a channel is $\tau_{min} = 10^{-4}$ s, and maximum sensing time for each SU for a channel is $\tau_{max} = 5 * 10^{-3}$ s; the minimum power allocated for a channel is $p_{min} = -30$ dBm; the maximum power allocated for a channel is $p_{max} = -3$ dBm; the minimum decision threshold for a channel is $\gamma_{min} = -30$ dB; the maximum decision threshold for a channel is $\gamma_{max} = -3$ dB. The transmitted power of the primary signal is 0.5 W. In the JSSPA-AMMOA algorithm, the number of initial population N_{pop} is setup to 100. The online stopping criterion is setup with the stopping threshold $\epsilon_s = 0.01$.

The initial populations consisting of 100 chromosomes are described in Fig. 5.4. With the stopping threshold $\epsilon_s = 0.01$, the algorithm converges to the efficient Pareto front at the iteration 380. The obtained Pareto front is described in Fig. 5.5.

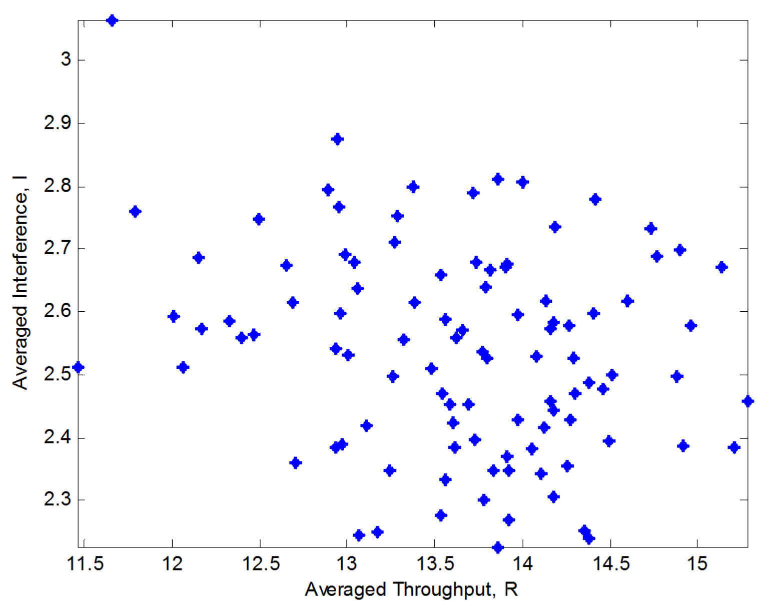


Figure 5.4: The objective space of the initial population.

It can be seen from Fig. 5.5 that our method is able to obtain a set of efficient Pareto solutions, or the Pareto front. The averaged throughput R is maximized, and the averaged interference I^{SU} is minimized significantly. We can see that these two

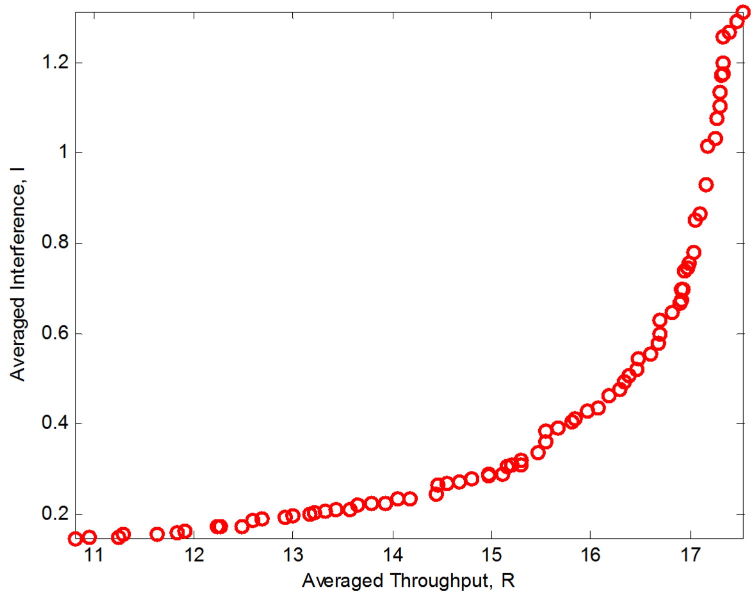


Figure 5.5: The objective space of the obtained Pareto optimal solutions.

objectives are conflicting seriously. There exists a set of efficient solutions for this joint optimization problem instead of one solution. Obtaining these efficient solutions helps the network select the best-suited solution in the context of the working environment at every moments. This is an important behavior of a cognitive system. For instance, at the moment t, for the reason of low-cost and a guaranteed quality of services, the network decides to select a solution that balances between the averaged throughput of the network and the averaged interference. In this case, at the moment t, the network selects the optimal solution with the averaged throughput $R = 16.5984$, and the averaged interference $I^{SU} = 0.5513$. The sensing times for each SU, the power control vector for each SU on each channel, and the decision threshold parameters for each SU on each channel are described in Fig. 5.7, Fig. 5.6, and Fig. 5.8, respectively.

If at the moment $t + 1$ the network is required to increase the network throughput, it can select the optimal solution in the Pareto optimal set, with the higher value of the averaged throughput objective. In this situation, the SUs are required to increase the

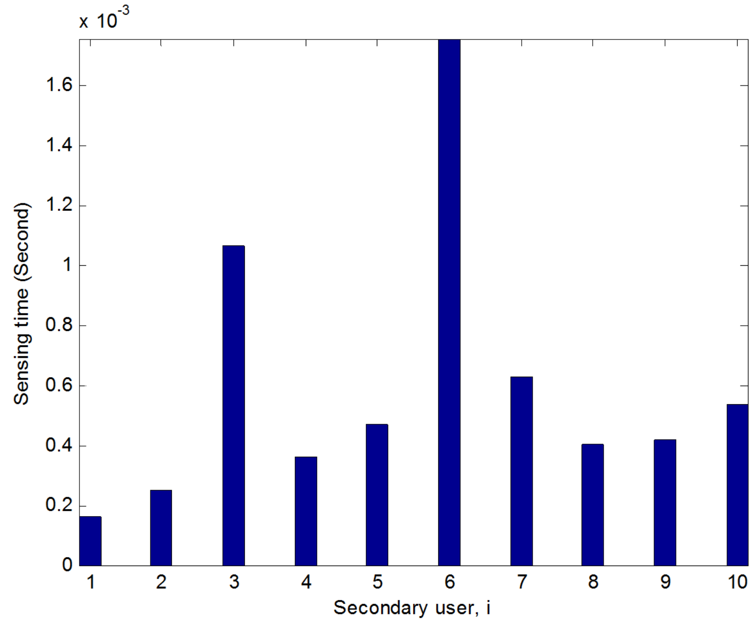


Figure 5.6: The decision threshold parameters of SUs obtained from the best-suited solution in the Pareto optimal set.

power for each channel and reduce the sensing time. The network throughput increases significantly. However, the interference generated by SUs to PU increases rapidly. This is a trade-off and win-win situation.

To observe the convergent behaviour of JSSPA-AMMOA, we run the algorithm without the online stopping criterion. That means the algorithm stops at a specified number of iterations. We observe the stopping indicator values SI_{dq} in 500 simulation run, as illustrated in Fig. 5.9. It can be seen that the algorithm converges well.

5.5 Summary of Chapter 5

In this chapter, a joint spectrum sensing and power allocation problem of a multiple-user multiple-channel cognitive radio network is formulated as a multiobjective optimization of finding sensing times, decision threshold vector, and power allocation vector of each

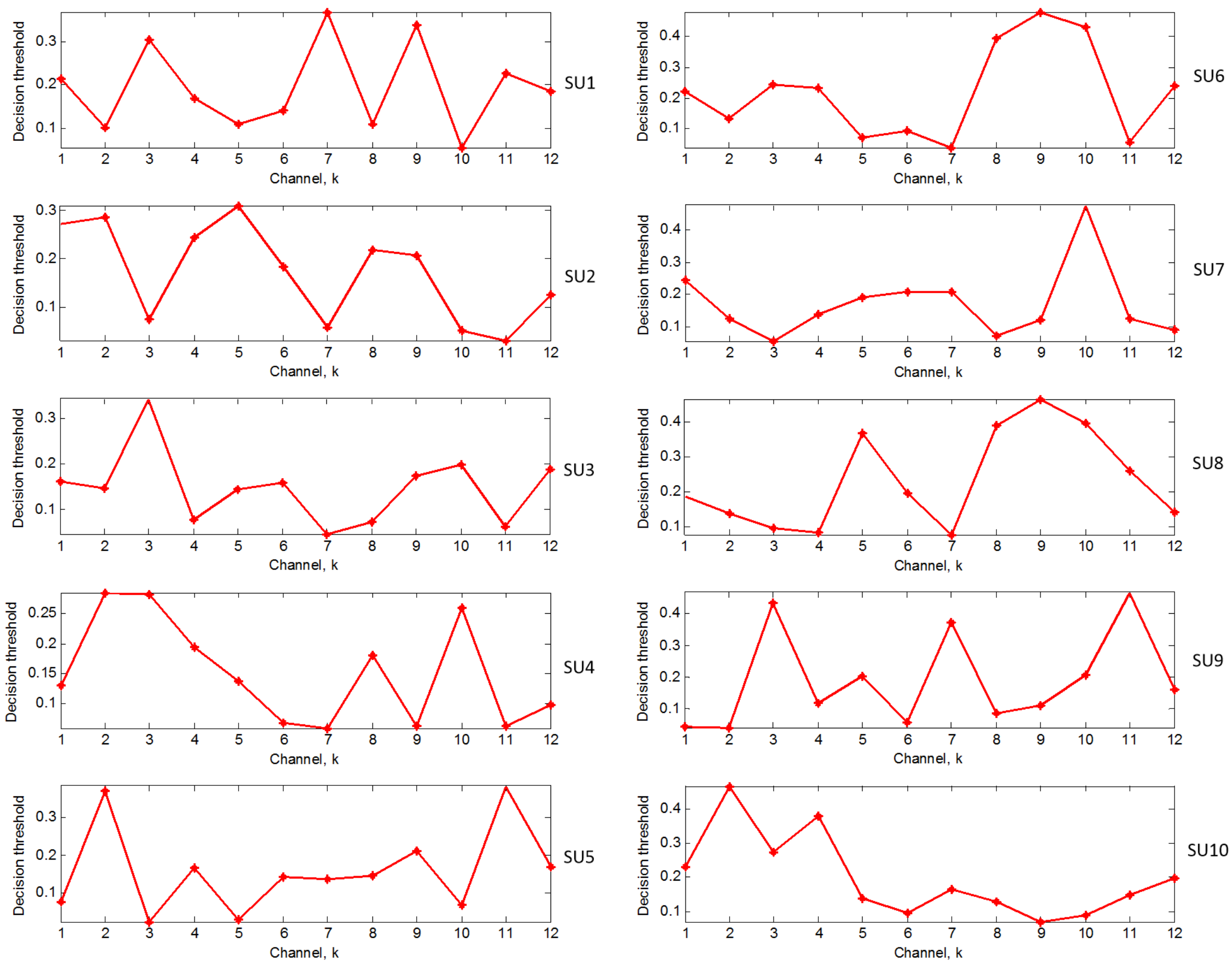


Figure 5.7: The sensing times of SUs obtained from the best-suited solution in the Pareto optimal set.

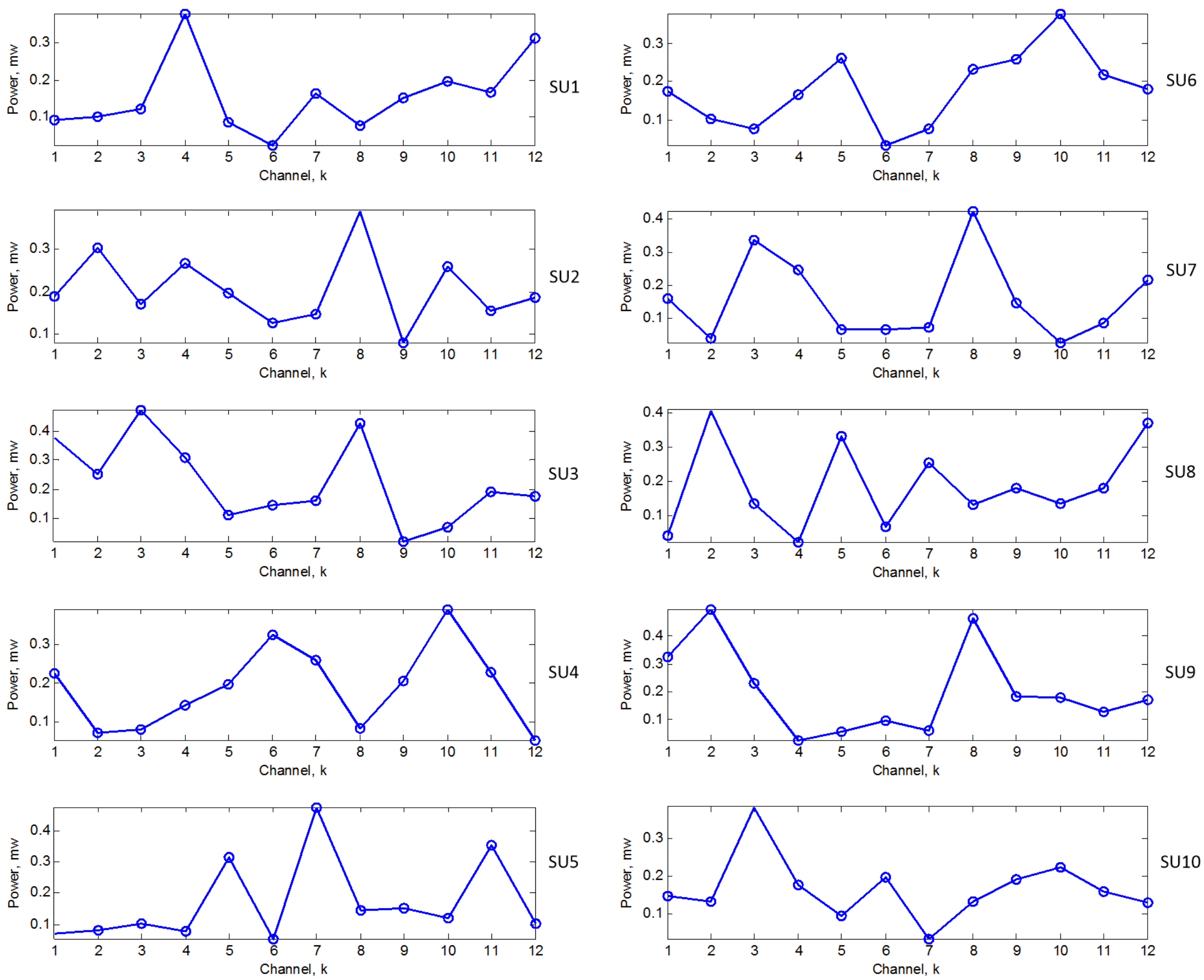


Figure 5.8: The power control vectors of SUs obtained from the best-suited solution in the Pareto optimal set.

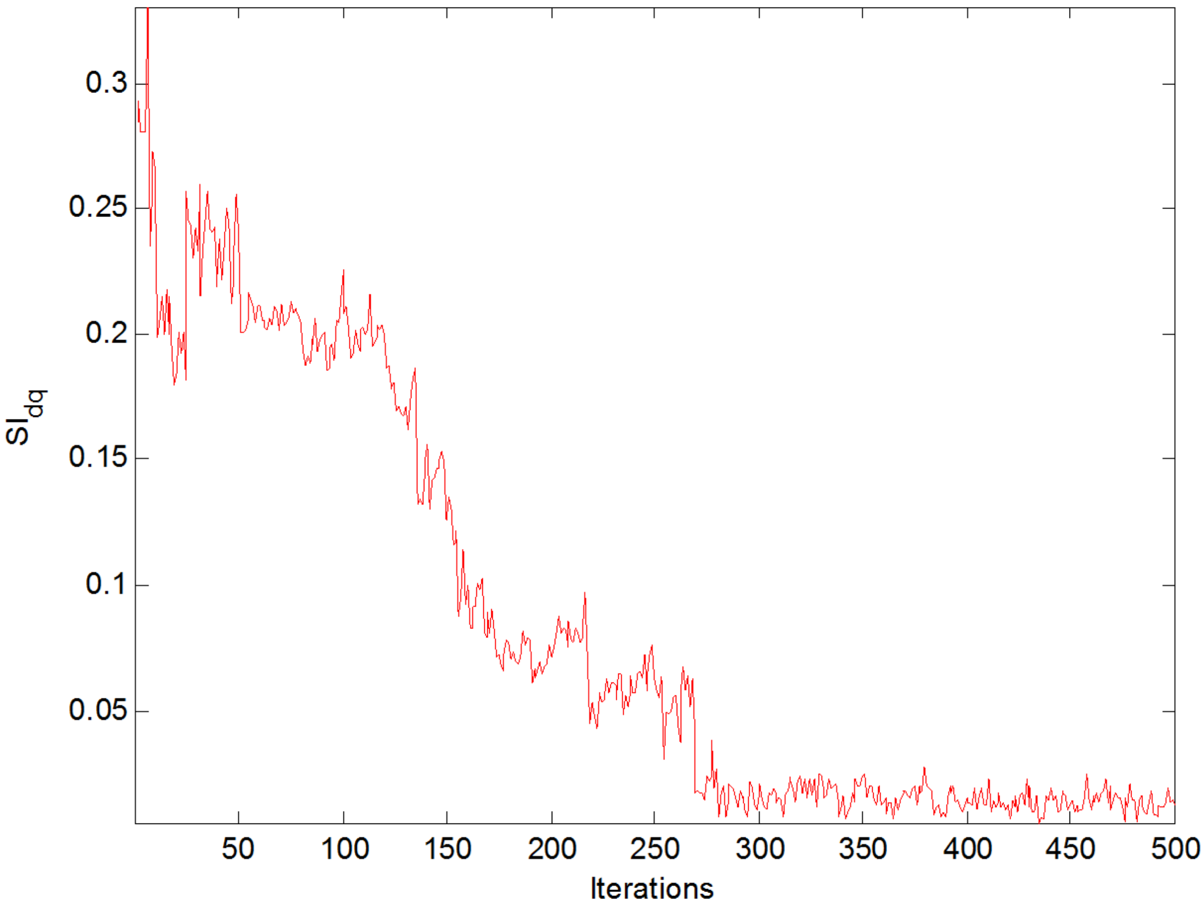


Figure 5.9: The online stopping indicator values SI_{dq}.

cognitive user to maximize the network throughput and minimize the interferences. A multiobjective memetic algorithm is proposed to solve this challenging multiobjective joint optimization problem. The simulation results show that the proposed method obtains very good performance and dynamic parameters in cooperative spectrum sensing and power allocation for cognitive radios. The effectiveness of applying memetic learning algorithm in solving multiobjective optimization is also stressed in this chapter.

In this work, we have so far considered the problem in which there is only one active primary user (PU) for simplification. This work can be extended to the multiobjective joint optimization problem with the presence of multiple PUs. To reduce the network resources, each secondary user should be able to learn cooperatively from the environment to obtain its optimal parameters. Therefore, no fusion center is preferred to used in this context. Distributed-cooperative memetic learning can be developed in the sense that each cognitive user (SU) is an memetic agent in an multiagent memetic learning algorithm. The distributed multiobjective joint spectrum sensing and power allocation based on multiagent memetic learning algorithms will be developing for our future research.

Chapter 6

Conclusions and Possible Extensions

6.1 Summary of Findings

This thesis presents our research results related to the multiobjective memetic optimization problem, including its theory and applications. In particular, a framework for adaptive multiobjective memetic optimization algorithms (AMMOA) has been introduced and studied. An adaptive online stopping criterion is introduced to assist AMMOA to detect its convergence in solving multiobjective optimization problems. An implementation of AMMOA is also presented with remarkable results for both test problems and for two applications of multiobjective image data hiding, and multiobjective joint spectrum sensing and power control in cognitive radio networks.

In adaptive multiobjective memetic optimization algorithms (AMMOA) presented in Chapter 3, we introduce a new information theoretic criterion used in AMOMA for guiding the adaptive learning processes, such as the selection, clustering, local searches, and the online stopping criterion. The experimental results of the implementation of AMOMA show that the framework performs well on both two-objective and three-objective optimization problems, and it outperforms the well-known multiobjective optimization NSGA-II.

In multiobjective image data hiding described in Chapter 4, we pose the issue of optimal image data hiding as a multiobjective optimization problem, and introduce a novel method of embedding and extracting a logo watermark into and from a colour image, based on a wavelet decomposition and a probabilistic neural network. The optimal embedding factors and the parameter of the used neural network are searched by AMMOA to maximize the quality of watermarked image and the robustness of the embedded watermark under different attacks. The proposed algorithm obtains better results in transparency and robustnesses against classes of additive noise, signal processing, and geometric transformation attacks than previous approaches. A new multiscale perceptual measure is also introduced to evaluate the imperceptibility of watermark with remarkable results.

In multiobjective joint spectrum sensing and power control of cognitive wireless networks described in Chapter 5, we have introduced the joint sensing and power control problem in a multichannel and multiple-user cognitive radio network as a multiobjective optimization problem of finding sensing times, decision threshold vector, and power allocation vector of each cognitive user to maximize the network throughput and minimize the interference. AMMOA is used to design a cooperative joint spectrum sensing and power allocation mechanism for a multiple-channel multiple-user cognitive radio network. The simulation results show that the proposed method obtains very good performance and dynamic parameters in cooperative spectrum sensing and power allocation for cognitive radios.

With the theoretic development, implementation, and experimentation for AMMOA, multiobjective image data hiding, and multiobjective joint spectrum sensing and power control in cognitive radio networks, it is concluded that the objectives of this thesis have been achieved. The following section lists the contributions of this thesis.

6.2 Contributions

We believe that the research done towards the thesis's completion has provided the following contributions.

1. An effective information-theoretic criterion is proposed to guide the adaptive processes such as the selection, clustering, and local learning processes in adaptive multiobjective optimization techniques.

2. A framework of adaptive multiobjective optimization algorithms (AMMOA) based on the proposed information-theoretic criterion is introduced.

3. An robust online stopping criterion is introduced for AMMOA.

4. An implementation of the AMMOA framework with the adaptive tournament selection, fuzzy-clustering, Tabu local searches, and an online stopping criterion, all guided by the proposed information-theoretic criterion, is introduced with remarkable results.

5. A multiobjective optimization problem of image data hiding is introduced.

6. Different classes of wavelets are analyzed experimentally to select an appropriate wavelet for robust and perceptual image data hiding based on computational intelligence.

7. A novel logo watermarking method for colour images is proposed based on wavelets and GRNN. The optimality of the method is achieved by using AMMOA.

8. A new multiscale perceptual measure, the relative Rényi dimension spectrum, is introduced for measuring the transparency of the watermark with remarkable results.

9. A multiobjective joint optimization problem between spectrum sensing and power control for a multichannel multiple-user cognitive radio network is introduced.

10. A novel multiobjective joint spectrum sensing and power control method is proposed based on AMMOA. This is the first approach to solve the joint optimization problem using multiobjective optimization.

6.3 Possible Extensions of This Research

The implementation of AMMOA presented in Chapter 3 obtained a very good performance in convergence and diversity for two- and three-objective problems. However, this implementation still have some disadvantages and need to be improved. Base on the work done in this thesis, we provide some possible extensions of this research as follows.

1. Studying for improvements of AMMOA's implementations by developing an effective fitness evaluation method and an efficient global-local learning strategy.

2. Studying on designing and implementing memetic evolvable hardwares based on FPGA for both adaptive and cognitive systems.

3. Studying on practical applications of AMMOA to solve variety of engineering design problems (e.g., multifractal antenna design, multiobjective navigation of robots).

More detailed analyses and discussions for the future research directions 1 and 2 are provided in Appendix E.

References

[AAKV04] Ioannis Anagnostopoulos, Christos Anagnostopoulos, George Kouzas, and Dimitrios D. Vergados, "A generalised regression algorithm for web page categorisation," *Neural Comput. Appl.*, vol. 13, no. 3, pp. 229-263, Sep. 2004.

[AhMo06] Farid Ahmed and Ira S. Moskowitz, "A semi-reversible watermark for medical image aunthentication," in *Proceedings of 1st Distributed Diagnosis and Home Healthcare Conference*, Arlington, Virginia, USA, pp. 59–62, April 2006.

[AlSt12] Sami M. Almalfouh, and Gordon L. Stuber, "Joint spectrum-sensing design and power control in cognitive radio networks: a stochastic approach," *IEEE Trans. Wireless Comm.*, vol. 11, no. 12, pp. 4372 – 4380, Dec. 2012.

[BaBP01] Mauro Barni, Member, Franco Bartolini, and Alessandro Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE Trans. Image Processing*, vol. 10, no. 5, pp. 783–791, May 2001.

[Bezd81] James C. Bezdek, *Pattern recognition with fuzzy objective function algorithms.* Norwell:MA, Kluwer Academic, 1981, 256 pp.

[BWBA09] L. T. Bui, S. Wesolkowski, A. Bender, H.A. Abbass, and M. Barlow, "A dominance-based stability measure for multi-objective evolutionary algorithms," in

Proc. IEEE Inter. Congress on Evolutionary Computation, CEC 2009, pp. 749–756, 2009.

[BTMA12] Christoph Bergmeir, Issac Triguero, Daniel Molina, Jose L. Aznarte, and Jose Manuel Benitez, “Time series modeling and forcasting using memetic algorithms for regime-switching models,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, no. 11, pp. 1841–1847, Nov. 2012.

[Cach04] Christian Cachin, “An information-theoretic model for steganography,” *Information and Computation*, vol. 192, no. 1, pp. 41 – 56, July 2004.

[CBBC11] Fabio Cancare, Sheetal Bhandari, Davide B. Bartolini, Matteo Carminati, and Macro D. Santambrogio, “A bird’s eye view of FPGA-based evolvable hardware,” in *Proc. 2011 NASA/ESA Conf. Adaptive Hardware and Systems (AHS)*, (San Diago, CA; June 6-9), pp. 169-175, 2011.

[ChLi03] Liang-Hua Chen and Jyh-Jiun Lin, “Mean quantization based image watermarking,” *Image and Vision Computing*, vol. 21, no. 8, pp. 717 – 727, August 2003.

[ChLi10] Chun-Hsien Chou and Kuo-cheng Liu, A perceptually tuned watermarking scheme for color images, *IEEE Trans. Image Processing*, vol. 19, no. 11, pp. 2966-2982, November 2010.

[CoCo05] Carlos A. Coello Coello and Nareli Cruz Cortes, “Solving multiobjective optimization problems using an artificial immune system,” *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, June 2005.

[CoCh07] Dinu Coltuc and Jean-Marc Chassery, “Very fast watermarking by reversible contrast mapping,” *IEEE Sig. Process. Letters*, vol. 14, no. 4, pp. 255 – 258, April 2007.

[COLT11] Xianshun Chen, Yew-Soon Ong, Meng-Hiot Lim, and Kay Chen Tan, "A multi-facet survey on memetic computation," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591–607, October 2011.

[CoMM99] Ingemar J. Cox, Matt L. Miller, and Andrew L. McKellips, "Watermarking as communications with side information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999.

[ChWo01] Brian Chen and Gregory W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[Daub90] Ingrid Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Information Theory*, vol. 36, no. 5. pp. 961–1005, Sept. 1990.

[Daub92] Ingrid Daubechies, Ten Lectures on Wavelets. Philadelphia, PA: SIAM, 1992, p. 357.

[DeAg95] Kalyanmoy Deb and Ram B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, pp. 115–148, Nov. 1995.

[DaKi14a] Hieu V. Dang and Witold kinsner, "Optimal colour image watermarking using neural networks and multiobjective memetic optimization," in *Proc. of the 2014 Intern. Conf. on Neural Networks and Fuzzy Systems, ICNN-FS 2014*, Venice, Italy, March 2014, pp. 64–73 .

[Dawk89] Richard Dawkins, *The Selfish Gene.* London:UK, Oxford University Press, 1989, 366 pp.

[Deb01] Kalyanmoy Deb, *Multi-Objective Optimization using Evolutionary Algorithms.* New York: NY, John Wiley & Sons, 2001, 497 pp.

[DeGo96] Kalyanmoy Deb and Mayank Goyal, "A combined genetic adaptive search for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[DaKi12b] Hieu V. Dang and Witold Kinsner, "An intelligent digital colour image watermarking approach based on wavelets and general regression neural networks," in *2012 IEEE 11th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, Kyoto: Japan, Aug. 2012, pp. 115–123.

[DaKi12a] Hieu V. Dang and Witold Kinsner, "A perceptual data hiding mathematical model for color image protection," *Journal of Advanced Mathematics and Applications*, vol. 1, no. 2, pp. 218–233, Dec. 2012.

[DaKi14b] Hieu V. Dang and Witold Kinsner, "Multiobjective memetic optimization for spectrum sensing and power allocation in cognitive wireless networks," in *Proc. of the Canadian Conf. on Electrical and Computer Engineering, CCECE 2014*, Torronto, Canada, May 2014.

[DaKW14c] Hieu V. Dang, Witold kinsner, and Yingxu Wang, "Multiobjective image data hiding based on neural networks and memetic optimization," *WSEAS Trans. Signal Processing*, 2014.

[DaKW14a] Hieu V. Dang, Witold kinsner, and Yingxu Wang, "Optimal colour image watermarking using neural networks and multiobjective memetic optimization," in *Proc. of the 2014 Intern. Conf. on Neural Networks and Fuzzy Systems, ICNN-FS 2014*, Venice, Italy, March 2014, pp. 64–73 .

[DaKi15a] Hieu V. Dang and Witold Kinsner, "A multiscale information theoretic criterion for adaptive multiobjective memetic optimization," Submitted to *IEEE Trans. Evolv. Compt.*, 2015.

[DaKi15b] Hieu V. Dang and Witold Kinsner, "An analytical multiobjective optimization of joint spectrum sensing and power control in cognitive radio networks," To appear in *Proc. of the 14th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing*, ICCI*CC 2015, (China, Beijing; August 22-24, 2015), pp. , 2015.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyaruvan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[Dono95] David L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[DTLZ02] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler, "Scalable multi-objective optimization test problems," in *2002. CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 1, May 2002, pp. 825–830.

[Dunn74] J. Dunn, "A fuzzy relative of ISODATA process and its use in detecting compact well separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, Sep. 1974.

[FaPC10] Garza-Fabre, M. and Toscano-Pulido, G. and Coello Coello, Carlos A, "Two novel approaches for many-objective optimization," in *2010 IEEE Congress on Evolutionary Computation (CEC)*, July 2010, pp. 1–8.

[FaPC09] Mario Garza-Fabre, Gregorio Toscano Pulido, and Carlos A. Coello Coello, "Ranking methods for many-objective optimization," *Lecture Notes in Computer Science*, vol. 5847, pp. 633–645, Nov. 2009.

[GBIB06] Lahouari Ghouti, Ahmed Bouridane, Mohammad K. Ibrahim, and Said Boussakta, "Digital image watermarking using balanced multiwavelets," *IEEE Trans. Sig. Process.*, vol. 54, no. 4, pp. 1519 – 1536, April 2006.

[GMBG10] Jose L. Guerrero, Antonio Berlanga, Jesus Garcia, and Jose M. Molina, "Introducing a robust and efficient stopping criterion for MOEAs," in *Proc. IEEE Inter. Congress on Evolutionary Computation*, CEC 2010, pp. 1–8, 2010.

[Glov90] Fred Glover, "Tabu search: a tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, Aug. 1990.

[GLZC07] John Y. Goulermas, Panos Liatsis, Xiao-Jun Zeng, and Phil Cook, "Density-driven generalized regression neural networks (DD-GRNN) for function approximation," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1683–1696, Nov. 2007.

[GoBe02] Timothy G. W. Gordon and Peter J. Bentley, "On evolvable hardware," *Soft Computing in Industrial Electronics*, vol. 101, pp. 279-323, 2002.

[Gold89] David E. Goldberg, *Genetic algorithm in search, optimization, and machine learning.* Reading: MA, Addison-Wesley, 1989, 423 pp.

[GoTa09] Chi-Keong Goh and Kay Chen Tan, *Evolutionary Multi-Objective Optimization in Uncertain Environments: Issues and Algorithms.* Berlin: Heidelberg, Springer, 2009, 269 pp.

[GrTy07] Garrison W. Greenwood and Andrew M. Tyrrell, *Introduction to evolvable hardware: A practical guide for designing self-adaptive systems*, John Wiley & Sons, NJ: New Jersey, pp. 202, 2007.

[Hayk99] Simon Haykin, *Neural networks: A comprehensive foundation.* Upper Saddle River: NJ, Pearson, 1999, 823 pp.

[Hayk05] Simon Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201 – 220, Feb. 2005.

[HoBh07] Ekram Hossain and Vijay K. Bhargava, *Cognitive Wireless Communication Networks.* New York: NY, Springer, 2007, 440 pp.

[HoNG94] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," in *Proceedings of 1994 IEEE Congress on Evolutionary Computation*, vol. 1, June 1994, pp. 82–87.

[Hugh05] Evan J. Hughes, "Evolutionary many-objective optimisation: Many once or one many?" in *Proceedings of 2005 IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 222–227.

[HHBW06] Simon Huba, Phil HingstonLuigi Barone,and Lyndon While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Compt.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[HsWu98] Chiou-Ting Hsu and Ja-Ling Wu, "Multiresolution watermarking for digital images," *IEEE Trans. Circuits. Systems*, vol. 45, no. 8, pp. 1097–1101, August 1998.

[IsYM03] Hisao Ishibuchi, Tadashi Yoshida, and Tadahiko Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 2, pp. 204–224, April 2003.

[IHTY09] Hisao Ishibuchi, Yasuhiro Hitotsuyanagi, Noritaka Tsukamoto, and Yusuke Nojima, "Implementation of multiobjective memetic algorithms for combinatorial optimization problems: A knapsack problem case study," in *Multi-Objective Memetic Algorithms*, Y.-S. O. Chi-Keong Goh and K. C. Tan, Eds. Berlin: Springer, 2009, pp. 27–49.

[IsMu98] Hisao Ishibuchi and Tadahiko Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Systems, Man, and Cybernetics - Part C*, vol. 28, no. 3, pp. 392–403, August 1998.

[Jasz02] Andrzej Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problema comparative experiment," *IEEE Trans. Evol. Compt.*, vol. 6, no. 4, pp. 402–412, Aug. 2002.

[Jasz03] Andrzej Jaszkiewicz, "Do multiple-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 2, pp. 133–144, April 2003.

[JiBr05] Yaochu Jin and Jurgen Branke, "Evolutionary optimization in uncertain environments a survey," *IEEE Trans. Evolutionary Computation*, vol. 9, no. 3, pp. 303–318, June 2005.

[Jara14] Emiliano Carreno Jara, "Multiobjective optimization by using evolutionary algorithms: The p-optimality criteria," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 2, pp. 167–179, April 2014.

[JaJS93] Nikil Jayant, James Johnston, and Robert Safranek, "Signal compression based on models of human perception, *Proc. IEEE*, vol. 81, no. 10, pp. 1385–1422, Oct. 1993.

[JZOG15] Lei Jiao, Xuan Zhang, B. John Oommen, and Ole-Christoffer Granmo, "Optimizing channel selection for cognitive radio networks using a distributed Bayesian learning automata-based approach," *Applied Intelligence* (Accepted for publication, July 2015).

[Kasd95] N. Jeremy Kasdin, "Discrete simulation of colored noise and stochastic processes and 1/f? Power law noise generation," *Proceedings of IEEE*, vol. 83, no. 5, pp. 802–827, May 1995.

[KuAS05] P. Kumsawat, K. Attakitmongcol, and A. Srikaew, "A new approach for optimization in image watermarking by using genetic algorithms," *IEEE Trans. Signal Processing*, vol. 53, no. 12, pp. 4707–4719, Dec. 2005.

[KaBe06] Irene G. Karybali and Kostas Berberidis, "Efficient spatial image watermarking via new perceptual masking and blind detection schemes," *IEEE Trans. Infom. Forensics Secur.*, vol. 1, no. 2, June 2006.

[KnCo00] Joshua D. Knowles and David W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, June 2000.

[KiDa06] Witold Kinsner and Richard Dansereau, "A relative fractal dimension spectrum as a complexity measure," in *Proc. 5th IEEE Int. Conf. on Cognitive informatics (ICCI'06)*, 2006, pp. 200–208.

[Kins02] Witold Kinsner, "Compression and its metrics for multimedia," in *Proc. of the 11th IEEE Intern. Conf. on Cognitive Informatics*, Calgary, Canada, August 2002, pp. 107 – 121.

[Kins12] Witold Kinsner, "Towards cognitive security systems," in *Proc. of the 11th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing, ICCI*CC 2012*, Kyoto, Japan, August 2012, pp. 539-539.

[Kins12b] Witold Kinsner, *Switching Automata Theory.* Winnipeg, MB: University of Manitoba, 2012, 930 pp.

[KuJB98] Martin Kutter, Frederic D. Jordan, and Frank Bossen, "Digital watermarking of color images using amplitude modulation," *J. Electronic Imaging*, vol. 7, no. 2, pp. 326–332, April 1998.

[Kins12c] Witold Kinsner, *Fractal and Chaos Engineering*, Lecture notes, Winnipeg, MB; Department of Electrical and Computer Engineering, University of Manitoba, 2012.

[KhYD03] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization*, Series of Lecture Notes in Computer Science, C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, and K. Deb, Eds. Berlin: Heidelberg, Springer, 2003, vol. 2632, pp. 376–390.

[KrSm05] Natalio Krasnogor and Jim Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Trans. Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, Oct. 2005.

[LeKi13] Ki-Baek Lee and Jong-Hwan Kim, "Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots," *IEEE Trans. Evol. Compt.*, vol. 17, no. 6, pp. 755–766, Dec. 2013.

[LeKn92] A. S. Lewis and G. Knowles, "Image compression using the 2-d wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 244–250, April 1992.

[LiBW11] Chaofeng Li, Alan C. Bovik, and Xiaojun Wu, "Blind image quality assessment using a general regression neural network," *IEEE. Trans. Neural Netw.*, vol. 22, no. 5, pp. 793–799, May 2011.

[LiYu12] Wang LinLin and Chen Yunfang, "Diversity based on entropy: A novel evaluation criterion in multi-objective optimization algorithm," *I.J. Intelligent Systems and Applications*, vol. 10, pp. 113–124, Apr. 2012.

[LSCS10] Andriana Lara, Gustavo Sanchez, Carlos A. Coello, and Oliver Schutze, "HCS: A new local serach strategy for memetic multiobjective evolutionary algorithms," *IEEE Trans. Evolutionary Computation*, vol. 14, no. 1, pp. 112–132, Feb. 2010.

[MaDD04] Benoit Macq, Jana Dittmann, and Edward J. Delp, "Benchmarking of image watermarking algorithms for digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 971–984, June 2004.

[Mall89] Stephan G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[MaHi93] L. Marquez and T. Hill, "Function approximation using backpropagation and general regression neural networks," in *Proceedings of the 16th IEEE Int. Conf. System Sciences*, Hawaii, 2010, pp. 607–615.

[Mess15] Achille Messac, *Optimization in practice with Matlab: For engineering students and professionals*. New York: NY, Cambridge University Press, 2015, 469 pp.

[Miet99] Kaisa Miettinen, *Nonlinear Multiobjective Optimization*. Norwell: MA, Kluwer Academic, 1999, 298 pp.

[MiMa99] Joseph Mitola and Gerald Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Communications Magazine*, vol. 6, no. 4, pp. 13-18, Aug. 1999.

[MLMH10] Daniel Molina, Manuel Lozano, Carlos Garcia-Martinez, and Francisco Herrera, "Memetic algorithms for continuous optimisation based on local search chains," *Evolutionary Computation*, vol. 18, no. 1, pp. 27–63, Feb. 2010.

[MoKo05] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, no. 12, pp. 2083–2126, Dec. 2005.

[MoSu03] P. Moulin and J. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Information Theory*, vol. 49, no. 3, pp. 563–593, Mar. 2003.

[Mosc89] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms," California Inst. Technol., Pasadena, CA, Tech. Rep. 826, 1989.

[MuMA04] Dipti P. Mukherjee, Subhamoy Maitra, and Scott T. Acton, "Spatial domain digital watermarking of multimedia objects for buyer authentication," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 1 – 15, Feb. 2004.

[NeCo12] Ferrante Neri and Carlos Cotta, "Memetic algorithms and memetic computing optimization: a literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, Feb. 2012.

[PaHJ04] Jeng-Shyang Pan, Hsiang-Cheh Huang, and Lakhmi C. Jain, *Intelligent Watermarking Techniques.* New Jersey, MA: World Scienstific, 2004, 852 pp.

[Padm13] Dhanesh Padmanabhan, "A clustering-based methodology for discontinuous pareto frontier representation," *Journal of Multi-Criteria Decision Analysis*, vol. 20, pp. 235–253, November 2013.

[PeAK99] Fabien A. P. Petitcolas, R. Anderson, and M. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[PaSc13] Jong-Shi Pang, and Gesualdo Scutari, "Joint sensing and power allocation in nonconvex cognitive radio games: Quasi-nash equilibria," *IEEE Trans. Sig. Process.*, vol. 61, no. 9, pp. 2366–2382, May 2013.

[Reny70] Alfred Rényi, *Probability Theory.* Budapest: Akademiai Kiado, 1970, 666 pp.

[RaRa11] K. Ramanjaneyulu and K. Rajarajeswari, "Wavelet-based oblivious image watermarking scheme using genetic algorithm," *IET Image Processing*, vol. 6, no. 4, pp. 364–373, June 2012.

[SiMD13] Karthik Sindhya, Kaisa Miettinen, and Kalyanmoy Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evolutionary Computation*, vol. 17, no. 4, pp. 495–511, Aug. 2013.

[TCSH13] Karaputugala M. Thilina, Kae W. Choi, Nazmus Saquib, and Ekram Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE J. Selected Areas in Communications*, vol. 31, no. 11, pp. 2209–2221, Nov. 2013.

[StGB11] Miro Stoyanov, Max Gunzburger, and John Burkardt, "Pink noise, 1/f? noise, and their effect on solutions of differential equations," *International Journal for Uncertainty Quantification*, vol. 1, pp. 257–278, March 2011.

[ShFL05] Rui-min Shen, Yong-gang Fu, Hong-tao Lu, "A novel image watermarking scheme based on support vector regression," *Journal of Systems and Software*, vol. 78, no. 1, pp. 1 – 8, Oct. 2005.

[SuOb03] Mohamed A. Suhail and Mohammad S. Obaidat "Digital watermarking-based DCT and JPEG model," *IEEE. Trans. Intrum. Meas.*, vol. 52, no. 5, pp. 1640-1647, Oct. 2003.

[SHWP04] Chin-Shiuh Shieh, Hsiang-Cheh Huang, Feng-Hsing Wang, and Jeng-Shyang Pan, "Genetic watermarking based on transform-domain techniques," *Pattern Recognition*, vol. 37, no. 3, pp. 555–565, March 2004.

[Spec91] Donald F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, no. 6, pp. 568–576, Nov. 1991.

[TaKL10] Kay Chen Tan, Eik Fun Khor, and Tong H. Lee, *Multiobjective evolutionary algorithms and applications.* London UK, Springer, 2010, 295 pp.

[TaSa08] Rahul Tandra, and Anant Sahai, "SNR walls for signal detection," *IEEE J. Selected Topics in Signal Process.*, vol. 2, no. 1, pp. 4–17, Feb. 2008.

[TeSw97] A. Tewfik and M. Swanson, "Data hiding for multimedia personalization, interaction, and protection," *IEEE Signal Processing Magazine*, vol. 14, no. 4, pp. 41–44, July 1997.

[TWNP09] H. Trautmann, T. Wagner, B. Naujoks, M. Preuss, and J. Mehnen, "Statistical methods for convergence detection of multi-objective evolutionary algorithms," *Evolutionary Computation*, vol. 17, no. 4, pp. 493–509, 2009.

[TsSu07] Hung-Hsu Tsai and Duen-Wu Sun, "Color image watermark extraction based on support vector machines," *Information Sciences*, vol. 177, no. 2, pp. 550–569, Jan. 2007.

[USCSIPI] [USCSIPI] The USC-SIPI Image Database, "http://http://sipi.usc.edu/database/", *Signal and Image Processing Institute, University of Southern California*, California, United States.

[VlDM02] Christophe D. Vleeschouwer, Jean-Francois Delaigle, and Benoit Macq, "Invisibility and application functionalities in perceptual watermarking," *Proceedings of IEEE*, vol. 90, no. 1, pp. 64–77, Jan. 2002.

[WaLi04] Shih-Hao Wang and Yuan-Pei Lin, "Wavelet tree quantization for copyright protection watermarking," *IEEE Trans. Image Processing*, vol. 13, no. 2, pp. 154–165, Feb. 2004.

[WaYC08] Xiang-Yang Wang, Hong-Ying Yang, and Chang-Ying Cui, "An SVM-based robust digital image watermarking against desynchronization attacks," *Signal Processing*, vol. 88, no. 9, pp. 2193 – 2205, Sep. 2008.

[WABB12] Yingxu Wang, James A. Anderson, George Baciu, Gerhard Budin, D. Frank Hsu, Mitsuru Ishizuka, Witold Kinsner, Fumio Mizoguchi, Kenji Sugawara, Shusaku Tsumoto, and Du Zhang, "Perspectives on ebrain and cognitive computing," *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, vol. 6, no. 4, pp. 1–21, October 2012.

[Wand95] A. B. Wandell, Foundation of vision, Sinuaer Associates, Inc., Sunderland, Massachussetts, Standford University edition (1995).

[WaCa12] Yong Wang and Zixing Cai, “Combining multiobjective optimization with differential evolution to solve constrained optimization problems,” *IEEE Trans. Evol. Compt.*, vol. 16, no. 1, pp. 117–134, Feb. 2012.

[Wick96] 34. V. M. Wickerhauser, Adapted wavelet analysis: from theory to software, A K Peters/CRC Press, Wellesley, MA, 1996.

[WuLi03] MinWu and Bede Liu, “Data hiding in image and video .i. fundamental issues and solutions,” *IEEE Trans. Image Processing*, vol. 12, no. 6, pp. 685–695, June 2003.

[WoPD99] Raymond B. Wolfgang, Christine I. Podilchuk, and Edward J. Delp, “Perceptual watermarks for digital images and video,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, Jul. 1999.

[WoPD99] Raymond B. Wolfgang, Christine I. Podilchuk, and Edward J. Delp, “Perceptual watermarks for digital images and video,” *Proceedings of IEEE*, vol. 87, no.7, pp. 1108-1126. July 1999.

[WinJET] [WinJET] The Oficial Website of The Winnipeg Jet, ”http://jets.nhl.com”, Winnipeg, MB, Canada.

[XiZH06] Ling Xie, Jiashu Zhang, and Hongjie He, “Robust audio watermarking scheme based on nonuniform discrete fourier transform,” *IEEE Intern. Conf. Engi. Intelligent Systems*, 2006, pp. 1–5.

[YuAr09] Tevfik Yucek, and Huseyin Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE Comm. Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, March 2009.

[Yage86] R. R. Yager, "On the theory of bags," *International Journal of General Systems*, vol. 13, no. 1, pp. 23–37, Dec. 1986.

[YuTL01] Pao-Ta Yu, Hung-Hsu Tsai, and Jyh-Shyan Lin, "Digital watermarking based on neural networks for color images," *Signal Processing*, vol. 81, no. 3, pp. 663 – 671, March 2001.

[ZAJP08] Yicong Zhou, Sos Agaian, Valencia M. Joyner, and Karen Panetta, "Two fibonacci p-code based image scrambling algorithms," in *SPIE Proceedings: Image processing - algorithms and systems VI*, vol. 6812, San Jose, Jan. 2008, pp. 1–12.

[ZiBa11] E. Zio and R. Bazzo, "A clustering procedure for reducing the number of representative solutions in the pareto front of multiobjective optimization problems," *European Journal of Operational Research*, vol. 210, no. 3, pp. 624–634, May 2011.

[ZiDT00] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, June 2000.

[ZiLT01] Eckart Zitzler, Marco Laumanns, and Lothar Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.

Appendix A

Pareto-optimal Fronts of Test Problems

In this appendix, we provide the Pareto-optimal fronts of test problems provided in Chapter 2. These reference Pareto-optimal fronts play an important roles in evaluating and comparing our proposed adaptive multiobjective memetic optimization algorithms with existing and widely-used multiobjective optimization evolutionary algorithms.

A.1 Test Problem: ZDT1

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ g(\mathbf{x}) &= 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \\ f_2(\mathbf{x}) &= 1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}} \end{aligned} \tag{A.1}$$

where n is the number of variables; $x_i \in [0, 1]$

The Pareto-optimal front of ZDT1 is described in Fig. A.1.

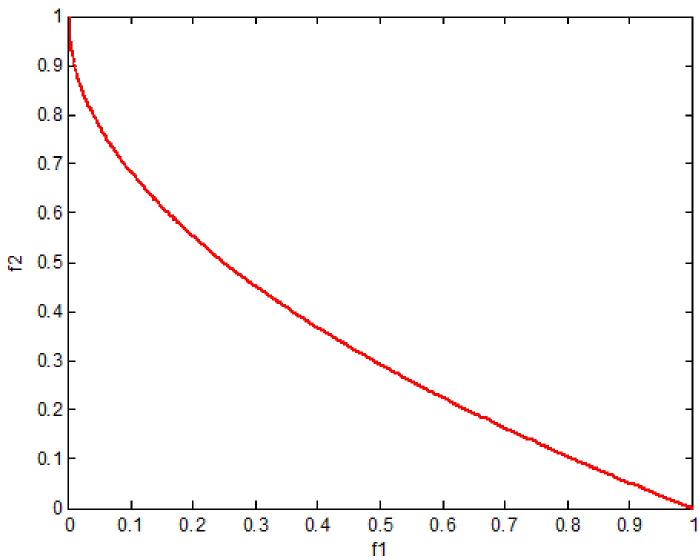


Figure A.1: The Pareto-optimal front of the test problem ZDT1

A.2 Test Problem: ZDT2

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ g(\mathbf{x}) &= 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \\ f_2(\mathbf{x}) &= g(\mathbf{x})\left(1 - \left(\frac{f_1(\mathbf{x})}{g(\mathbf{x})}\right)^2\right) \end{aligned} \tag{A.2}$$

where n is the number of variables; $x_i \in [0, 1]$

The Pareto-optimal front of ZDT2 is described in Fig. A.2.

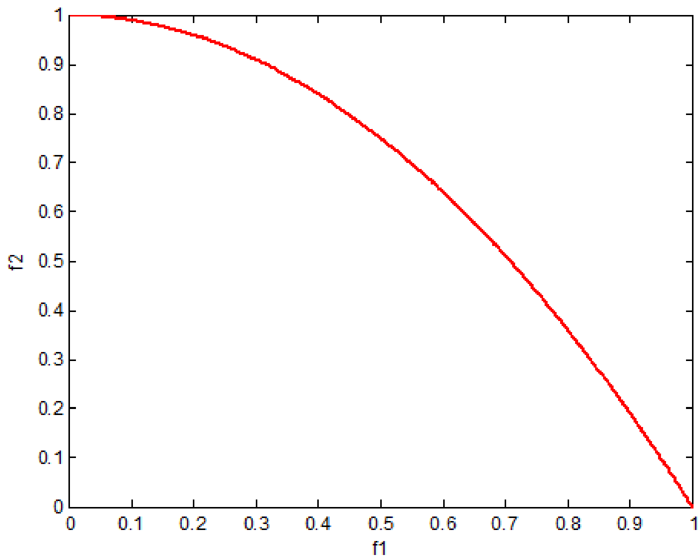


Figure A.2: The Pareto-optimal front of the test problem ZDT2

A.3 Test Problem: ZDT3

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ g(\mathbf{x}) &= 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \\ f_2(\mathbf{x}) &= g(\mathbf{x})\left(1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}} - \frac{f_1(\mathbf{x})}{g(\mathbf{x})}\sin(10\pi f_1(\mathbf{x}))\right) \end{aligned} \tag{A.3}$$

where n is the number of variables; $x_i \in [0, 1]$

The Pareto-optimal front of ZDT3 is described in Fig. A.3.

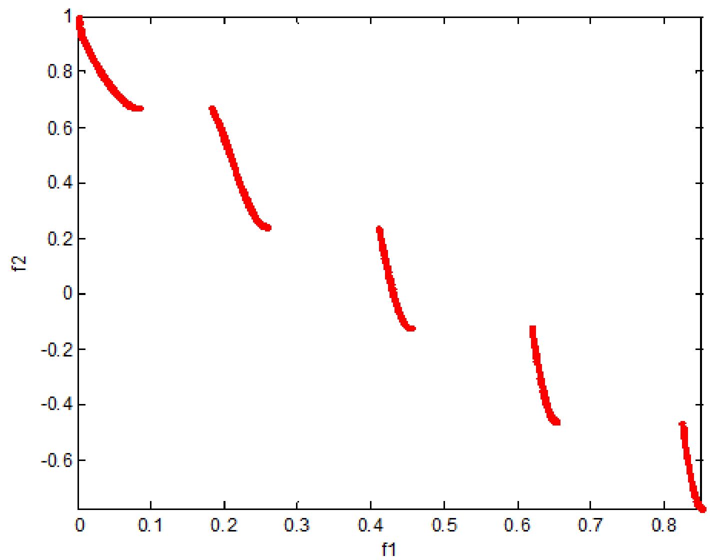


Figure A.3: The Pareto-optimal front of the test problem ZDT3

A.4 Test Problem: DTLZ2

$$
\begin{aligned}
f_1(\mathbf{x}) &= (1+g(\mathbf{x}_M)).\cos(0.5\pi x_1)...\cos(0.5\pi x_{M-1}) \\
f_2(\mathbf{x}) &= (1+g(\mathbf{x}_M)).\cos(0.5\pi x_1)...\sin(0.5\pi x_{M-1}) \\
&... \\
f_M(\mathbf{x}) &= (1+g(\mathbf{x}_M)).sin(0.5\pi x_1) \\
g(\mathbf{x}_M) &= \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2
\end{aligned}
\tag{A.4}
$$

where $x_i \in [0,1]$, $i = 1, 2, ..., n$; M is the number of objectives; n is the number of variables. The last $k = (n - M + 1)$ variables are represented as $\mathbf{x}_M$.

The Pareto-optimal front of DTLZ2 is described in Fig. A.4.

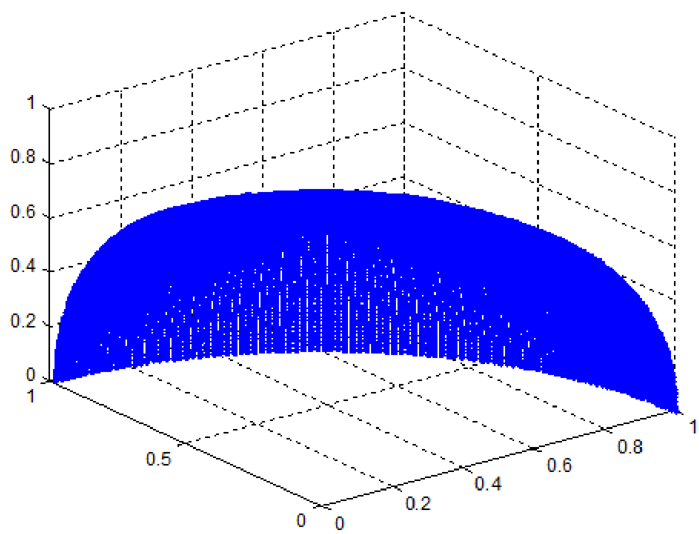


Figure A.4: The Pareto-optimal front of the test problem DTLZ2

Appendix B

Proofs of Lemmas and Theorems

In this appendix, we provide the proofs for Lemmas 3.1 – 3.7 in Chapter 3.

B.1 Proof of Lemma 3.1

Lemma 3.1: Let $\mathbf{x}_t$ and $\mathbf{x}_q$ ($t \# q$, $1 \leq t, q \geq N_P$) are the feasible solutions of the population $\boldsymbol{P}$. The solution $\mathbf{x}_t$ is said to dominate the solution $\mathbf{x}_q$ if the relative probabilities of objective functions satisfies $p_m(\mathbf{x}_t) \geq p_m(\mathbf{x}_q)$, $\forall m = \{1, 2, ..., M\}$, and $\exists n \in \{1, 2, ..., M\} : p_n(\mathbf{x}_t) > p_n(\mathbf{x}_q)$.

Proof of Lemma 3.1. The Eq. (3.3) is equivalent to

$$\mathbb{1}_{(\max(f_m(\mathbf{x}_j) - f_m(\mathbf{x}_i), 0))} = \begin{cases} 1, & \text{if } f_m(\mathbf{x}_i) < f_m(\mathbf{x}_j); \\ 0, & \text{if } f_m(\mathbf{x}_i) \geq f_m(\mathbf{x}_j); \end{cases} \tag{B.1}$$

From Eqs. (3.2) and (B.1), it can be stated that $p_m(\mathbf{x}_t) > p_m(\mathbf{x}_q)$ if $f_m(\mathbf{x}_t) < f_m(\mathbf{x}_q)$, $\forall m = 1, 2, ..., M$. This means that the solution $\mathbf{x}_q$ is dominated by the solution $\mathbf{x}_t$. $\square$

B.2 Proof of Lemma 3.2

Lemma 3.3: solution $\mathbf{x}^* \in \boldsymbol{P}$ is said to be non-dominated solution in the population $\boldsymbol{P}$ if $\forall \mathbf{x} \in \boldsymbol{P}$, $p_m(\mathbf{x}^*) \geq p_m(\mathbf{x})$, $\forall m = \{1, 2, ..., M\}$, and $\exists n \in \{1, 2, ..., M\} : p_n(\mathbf{x}^*) > p_n(\mathbf{x})$.

Proof of Lemma 3.2. Follow the proof of Lemma 3.1, it can be easily inferred that $f_m(\mathbf{x}^*) \leq f_m(\mathbf{x})$, $\forall m = 1, 2, ..., m$ and $\exists n \in \{1, 2, ..., M\} : f_n(\mathbf{x}^*) > f_n(\mathbf{x})$ **iff** $\forall \mathbf{x} \in \boldsymbol{P}$, $p_m(\mathbf{x}^*) \geq p_m(\mathbf{x})$, $\forall m = \{1, 2, ..., M\}$, and $\exists n \in \{1, 2, ..., M\} : p_n(\mathbf{x}^*) > p_n(\mathbf{x})$. This means $\mathbf{x}^*$ is a non-dominated solution in the population $\boldsymbol{P}$. $\square$

B.3 Proof of Lemma 3.3

Lemma 3.3: The distance measure between $\boldsymbol{r}$ and $\boldsymbol{p}(\mathbf{x})$, $f^{Rq}(\mathbf{x})$, is a monotonic non-increasing function in $q\#0, 1$.

Proof of Lemma 3.3. Consider the first derivative of $f^{Rq}(\mathbf{x})$ with respect to q as follows.

$$\begin{aligned}\frac{f^{Rq}(\mathbf{x})}{dq} &= \frac{\left[\log \frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{M}\right]_q'(q-1) - \log \frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{M}}{(q-1)^2} \\ &= \frac{\frac{(q-1)\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}\log p_j(\mathbf{x})}{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}} - \log \frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{M}}{(q-1)^2} \\ &= \frac{-\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}\log(p_j(\mathbf{x}))^{1-q} - \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}. \log \frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{M}}{(q-1)^2\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}} \end{aligned} \quad \text{(B.2)}$$

We must show that Eq. (B.2) is negative everywhere. Since the denumerator of Eq. (B.2) is always positive, we just need to proof the numerator of Eq. (B.2) is always negative. According to the log sum inequality, we have

$$\begin{aligned}-\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}\log(p_j(\mathbf{x}))^{1-q} &\leq \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{1-q}}{\sum_{j=1}^{M}1} \\ &= \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{1-q}}{M} \end{aligned} \quad \text{(B.3)}$$

The numerator of Eq. (B.2) can now derive as

$$
\begin{aligned}
Fp &= -\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}\log(p_j(\mathbf{x}))^{1-q} - \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}.\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{M} \\
&\leq \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}.\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{1-q}}{M} - sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}.\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{M} \\
&= \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}.\left(\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{1-q}}{M} - \log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{M}\right) \\
&= \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}.\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{1-q}}{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}} \\
&= \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}.\log\frac{\sum_{j=1}^{M}p_j(\mathbf{x})(p_j(\mathbf{x}))^{-q}}{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}} \\
&\leq \sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}.\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}}{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{-q}} \\
&\leq 0
\end{aligned}
\tag{B.4}
$$

Therefore, $f^{Rq}(\mathbf{x})$ is a monotonic non-increasing function in $q\#0,1$. □

B.4 Proof of Lemma 3.4

Lemma 3.4: The measure $f^{Rq}(\mathbf{x}) = 0$ when $q = 0$.

Proof. When $q = 0$, $f^{Rq}(\mathbf{x})$ reduces to

$$
\begin{aligned}
f^{Rq}(\mathbf{x}) &= -\log\frac{\sum_{j=1}^{M}(p_j(\mathbf{x}))^{0}}{M} \\
&= -\log\frac{M}{M} \\
&= 0
\end{aligned}
\tag{B.5}
$$

□

B.5 Proof of Lemma 3.5

Lemma 3.5: If the distribution $\boldsymbol{p}(\mathbf{x})$ has reached the reference vector $\boldsymbol{r}$, then the distance $f^{Rq}(\mathbf{x}) = 0$.

Proof. When the relative probability vector $\boldsymbol{p}(\mathbf{x})$ reach the reference vector $\boldsymbol{r}$, we have $p_j(\mathbf{x}) = 1$, $\forall j = \{1, 2, .., M\}$. Then, $f^{Rq}(\mathbf{x})$ deduce to

$$\begin{aligned} f^{Rq}(\mathbf{x}) &= \frac{1}{q-1} \log \frac{\sum_{j=1}^{M} (1)^{-q}}{M} \\ &= \frac{1}{q-1} \log \frac{M}{M} \\ &= 0 \end{aligned} \tag{B.6}$$

□

B.6 Proof of Lemma 3.6

Lemma 3.6: For $q \leftarrow 1$, $f^{Rq}(\mathbf{x})$ becomes Kullback-Leibler divergence:

$$\begin{aligned} \underset{q \leftarrow 1}{f^{Rq}(\mathbf{x})} &= \sum_{j=1}^{M} r_j \log \frac{r_j}{p_j(\mathbf{x})} \\ &= \sum_{j=1}^{M} - \log p_j(\mathbf{x}) \end{aligned} \tag{B.7}$$

B.7 Proof of Lemma 3.7

Lemma 3.7: Let $R_{RSE}(\mathbf{x}_1)$ and $R_{RSE}(\mathbf{x}_2)$ are the R_{RSE} of the solution $\mathbf{x}_1$, and the R_{RSE} of the solution $\mathbf{x}_2$ in the feasible population $\mathcal{P}$, respectively. if $R_{RSE}(\mathbf{x}_1) < R_{RSE}(\mathbf{x}_2)$, then solution $\mathbf{x}_1$ dominates the solution $\mathbf{x}_2$.

Proof. Let define function $f(\mathbf{x}_1, \mathbf{x}_2) = R_{RSE}(\mathbf{x}_1) - R_{RSE}(\mathbf{x}_2)$. We now prove that if $\mathbf{x}_1$ is said to dominate the solution $\mathbf{x}_2$, then $f(\mathbf{x}_1, \mathbf{x}_2) < 0$.

From Eq. (3.5), we can derive $f(\mathbf{x}_1, \mathbf{x}_2)$ as follows

$$\begin{aligned}
f(\mathbf{x}_1, \mathbf{x}_2) &= \sum_{q=1}^{20} \left(f^{Rq}(\mathbf{x}_1) - f^{Rq}(\mathbf{x}_2)\right) \\
&= \sum_{q=1}^{20} \frac{1}{q-1} \left(\log \frac{\sum_{j=1}^{M} (p_j(\mathbf{x}_1))^{-q}}{M} - \log \frac{\sum_{j=1}^{M} (p_j(\mathbf{x}_2))^{-q}}{M} \right) \\
&= \sum_{q=1}^{20} \frac{1}{q-1} \log \frac{\sum_{j=1}^{M} (p_j(\mathbf{x}_1))^{-q}}{\sum_{j=1}^{M} (p_j(\mathbf{x}_2))^{-q}}
\end{aligned} \tag{B.8}$$

- If $q = 1$, based on Lemma 3.6, we have

$$\begin{aligned}
f(\mathbf{x}_1, \mathbf{x}_2) &= \sum_{j=1}^{M} -\log p_j(\mathbf{x}_1) - \sum_{j=1}^{M} -\log p_j(\mathbf{x}_2) \\
&= \sum_{j=1}^{M} \log \frac{p_j(\mathbf{x}_2)}{p_j(\mathbf{x}_1)}
\end{aligned}$$

  Since $\mathbf{x}_1$ dominates $\mathbf{x}_2$, according to Lemma 3.1, $p_j(\mathbf{x}_2) < p_j(\mathbf{x}_1)$, $j = 1, 2, ..., M$. Thus, following Eq. (B.7), $f(\mathbf{x}_1, \mathbf{x}_2) < 0$.

- if $q > 1$, we have, $\mathbf{x}_1$ dominates $\mathbf{x}_2$ then $p_j(\mathbf{x}_1) > p_j(\mathbf{x}_2)$, $j = 1, 2, ..., M$. It is equivalent to

  $(p_j(\mathbf{x}_1))^{-q} < (p_j(\mathbf{x}_2))^{-q}$, $j = 1, 2, ...M$. This leads to

$$\begin{aligned}
f(\mathbf{x}_1, \mathbf{x}_2) &= \sum_{q=1}^{20} \frac{1}{q-1} \log \frac{\sum_{j=1}^{M} (p_j(\mathbf{x}_1))^{-q}}{\sum_{j=1}^{M} (p_j(\mathbf{x}_2))^{-q}} \\
&< \sum_{q=1}^{20} \frac{1}{q-1} \log 1 = 0
\end{aligned} \tag{B.9}$$

Therefore, if $1 \leq q \leq 20$, $R_{RSE}(\mathbf{x}_1) < R_{RSE}(\mathbf{x}_2)$ if the solution $\mathbf{x}_1$ is said to dominate the solution $\mathbf{x}_2$. $\square$

Appendix C

Benchmark Test Images

We use four highly-texture colour images which are commonly used in image processing research for the test images. The test images of Lena, Babbon, Airplane-F16, and House images with the size of (512-by-512)-by-3 from [USCSIPI] are shown in Fig. C.2, C.3, C.4, and C.5, respectively. The watermark image used for testing is the Winnipeg Jet logo with the size of 64-by-64 from [WinJET] shown in Fig. C.1.

Figure C.1: The Winnipeg-Jet logo watermark image.

Figure C.2: Lena test image.

Figure C.3: Baboon test image.

Figure C.4: Airplane-F16 test image.

Figure C.5: House test image.

Appendix D

Wavelets and HVS for Perceptual Data Hiding

In this appendix, we provide some basis of *human visual system* (HVS) models for image data hiding, wavelet decomposition technique, and a brief introduction of our perceptual image watermarking based on HVS and wavelets.

D.1 Concept of Using HVS Models for Perceptual Data Hiding

The human visual system has been extensively studied over the years in order to use this knowledge for image and video processing applications [LeKn92, JaJS93, Wand95, WoPD99]. Lewis and Knowles [LeKn92] defined the HVS as an information processing system, receiving spatially sampled images from the cones and rods in the eye, and deducing the nature of the objects it observes by processing this image data. More structurally, Wandell [Wand95] defined the HVS as an information system of three successive processing stages: encoding, representation, and interpretation. The lowest-level stage (e.g., encoding) encodes light into electrical signals by the photocells of the retina. The representation turns encoded visual signal from the en-coding stage to specific characteristics of the image. The interpretation stage is the highest stage of human vision. This stage is located in the brain, and it depends on each individual experience. The effects of motion, depth, color appearance, and visual understanding are created in the interpretation stage.

The HVS has a limited sensitivity, depending on the anatomy of the eye, its limitations and imperfections, and the characteristics of the visual signals [WoPD99, VlDM02]. In the context of image coding, there are three important sensitivities: frequency sensitivity, luminance sensitivity, and contrast sensitivity. The frequency sensitivity provides a basic visual model that depends only on viewing conditions and is independent of image content. Luminance sensitivity is a way to measure the effect of the detectability threshold of noise on a constant background of the image. The contrast sensitivity refers to the detectability of one signal in the presence of another signal. The effect of contrast sensitivity is strongest when both signals are of the same spatial frequency, orientation, and location [WoPD99]. These three sensitivities create visibility thresholds of the HVS, and are a basis for visual masking effects.

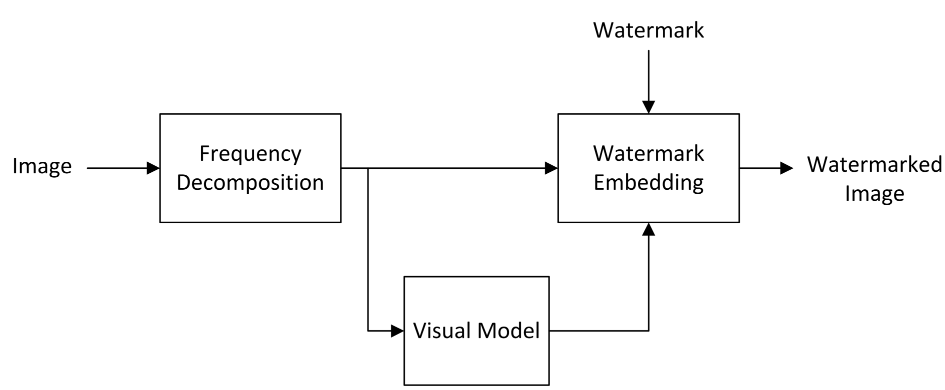


Figure D.1: Using HVS for perceptual data hiding.

An effective perceptual model allows us to take advantage of characteristics of the human visual systems not only in order to remove redundancy in designing optimal compression algorithms, but also to select suitable locations for perceptually embedding data into the image. How can we create a mathematical visual model that effectively exploits these visual masking effects for image and video processing applications? To find the most effective visual model has been an open-ended issue for researchers for many years. The frequency decomposition techniques such as the DCT and wavelets can mimic the HVSs structure in order to gain the most in terms of visual masking. The visual models based on the DCT and wavelets have been studied widely for optimal compression and perceptual watermarking [LeKn92, JaJS93]. The framework for applying a visual model for data hiding is described in Fig. D.1.

D.2 Wavelets Decomposition

Extensive experimental researches about HVS have been conducted by visual psychologists over the years. They discovered that the human eye filters the image into a number of bands, each approximately one octave wide in frequency [LeKn92]. Wavelet decomposition is considered to closely mimic the HVSs structure in perception [LeKn92, WoPD99]. A wavelet transform is very suitable for identifying the disturbed areas where tamperings can be hidden more easily. This property allows one to exploit the HVS frequency masking effect. If a wavelet coefficient is modified, only the region of the image where the particular frequency corresponding to that coefficient is present is modified [JaJS93, ChLi10]. Due to space limitation, this section provides just a short review of wavelet decomposition.

One dimensional wavelets (wavelet bases) are functions generated from one single function so-called a single mother wavelet, ψ, by dilation and translations [Daub90, Daub92]. The mother wavelet ψ has to satisfy $\int \psi(x)dx = 0$, and $\int |\Psi(w)|^2|w|^{-1}dw < \infty$, where $\Psi(w)$ is the Fourier transform of the mother wavelet function $\psi(t)$. A wavelet atom $\psi_{i,j}(t)$, localized around the point $2^i j$ and has a support size proportional to the scale 2^i, is defined by

$$\left[\psi_{i,j}(t) = \frac{1}{2^i}\psi\left(\frac{t-2^i j}{2^i}\right)\right]_{i\in\mathbb{Z},j\in\mathbb{Z}} \tag{D.1}$$

The basic idea of the wavelet transform is to represent any arbitrary function f as a superposition of wavelets [Daub92]. That means the function f is decomposed into different scale levels, where each level is then further decomposed with a resolution adapted to it. The wavelet decomposition can be then expressed by

$$f = \sum c_{i,j}(f)\psi_{i,j} \tag{D.2}$$

where the wavelet coefficients

$$c_{i,j}(f) = \langle\psi_{i,j}, f\rangle = \int \psi_{i,j}(x)f(x)dx \tag{D.3}$$

In a multiresolution analysis, there have two functions: the mother wavelet ψ, and a scaling function ϕ. The scaling function ϕ defines an orthogonal basis using dyadic dilations and translations. The dilated and translated versions of the scaling function is given by

$$\phi_{i,j}(x) = 2^{-i/2}\phi(2^{-i} - j) \tag{D.4}$$

The wavelet transform can be implemented by quadrature mirror filters [Mall89, Wick96]. In the forward wavelet transform, we use analytic filters including the low-pass filters $L = (l(n))$, $n \in \mathbb{Z}$, and the high-pass filters $H = (h(n))$, $n \in \mathbb{Z}$, where $l(n)$ and $h(n)$ are defined by

$$l(n) = \frac{1}{2}\langle\phi(x/2), \phi(x-n)\rangle \tag{D.5}$$

$$h(n) = (-1)^n l(1-n) \tag{D.6}$$

The reconstruction filters including the low-pass filter $\tilde{L}$, and the high-pass filter $\tilde{H}$ have impulse responses $\tilde{l}(n) = l(1-n), \tilde{h}(n) = h(1-n)$, respectively. The filter bank structure for one-dimensional (1D) wavelet decomposition and reconstruction in 1-level is illustrated in Fig D.2.

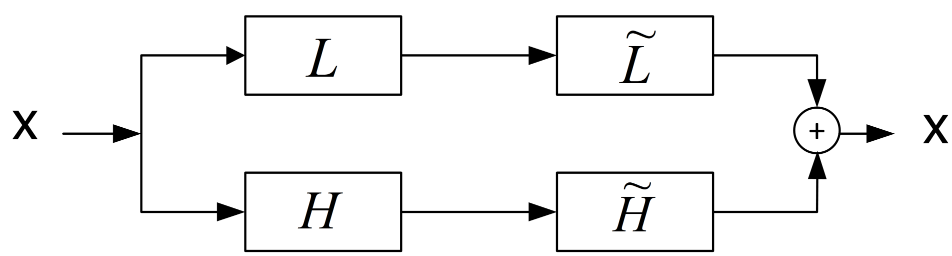


Figure D.2: Filter bank structure for 1-level wavelet transform.

For two dimensional signals (i.e., images), we use the hierarchical wavelet decomposition introduced by Mallat [Mall89]. The low-pass and high-pass filters L, H are applied to the image in both horizontal and vertical directions. The filter outputs are then subsampled by a factor of two to generate three orientation selective high-pass subbands HH, HL, LH, and a low-pass subband LL. The process is then repeated on the subband LL to create the next level of the wavelet decomposition. The 4-level wavelet decomposition using the filter bank technique resulting in thirteen subbands can be displayed by filter bank structure in Fig. D.3, or can be displayed by wavelet tree structure by Fig. D.4.

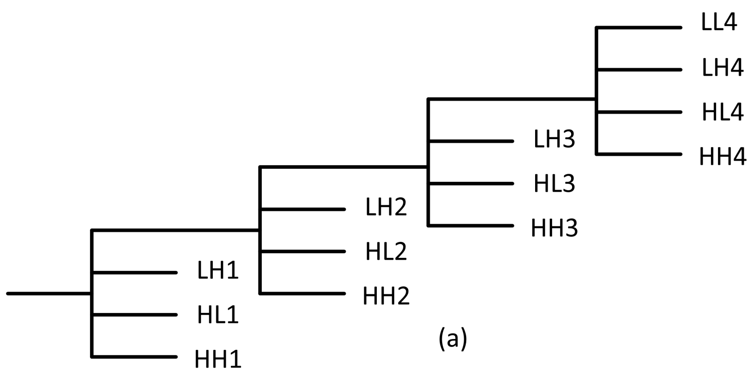


Figure D.3: Four-level wavelet decomposition (Displaying by filter bank structure).

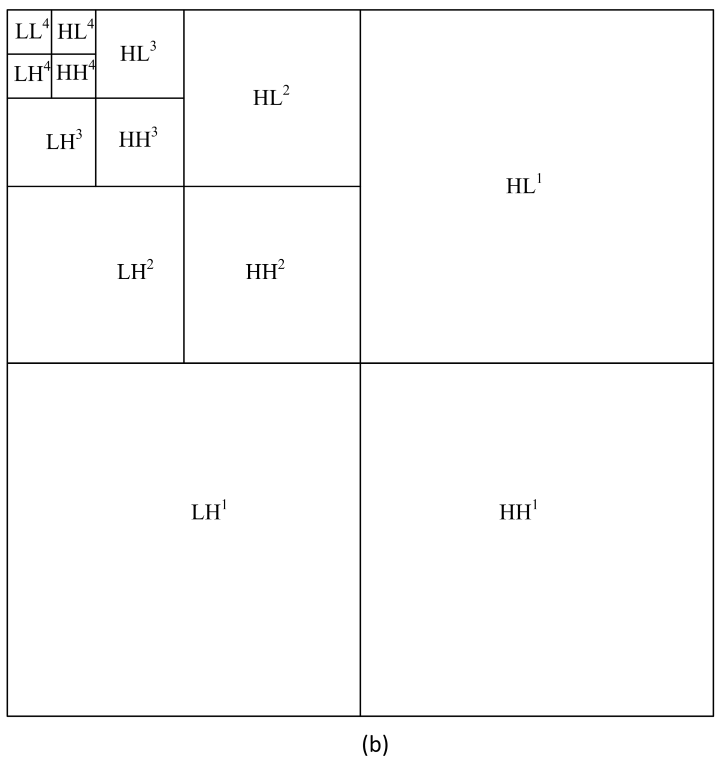


Figure D.4: Four-level wavelet decomposition (Displaying by wavelet tree structure).

D.3 A Perceptual Image Watermarking Based on HVS and Wavelets

In this section, we provide an example of using HVS in wavelet domain for a perceptual image watermarking. Studies in HVS models have shown that the human eye is: (i) less sensitive to noise in high resolution bands; (ii) less sensitive to noise in those areas of the image where brightness is high or low; and (iii) less sensitive to noise in highly texture areas but, among these, more sensitive near the edges. To have a perceptual image watermarking, the watermarking system should adopt HVS to analyze the local properties of the host signal to find the suitable area in transformed or non-transformed domain for embedding the watermark. To adapt the watermarking system to the local properties of the image, we use the quantization model based on HVS introduced in [LeKn92] to calculate the HVS mask in wavelet domain for a perceptual watermark embedding algorithm.

The watermark embedding scheme is depicted in Fig. D.5.

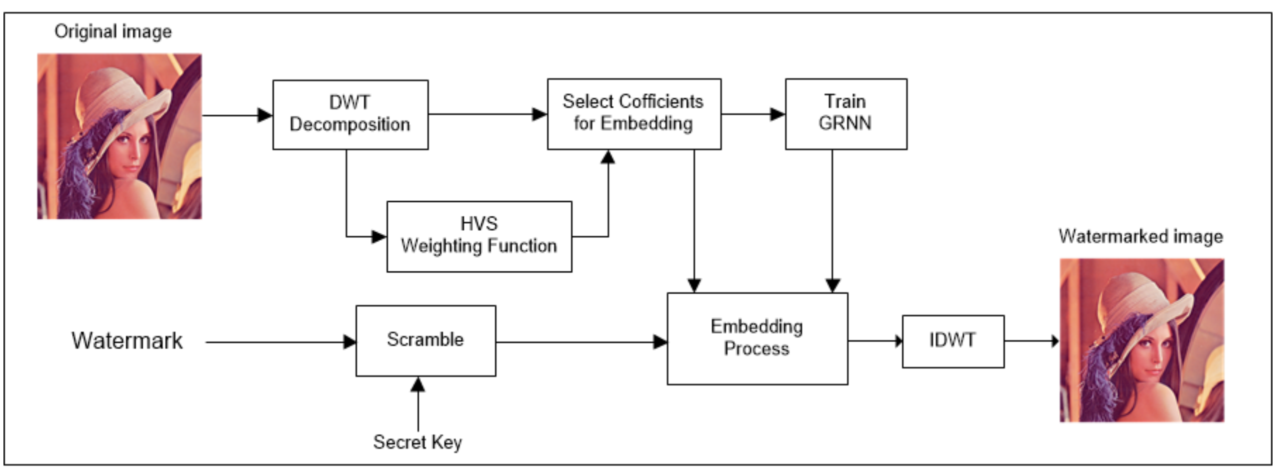


Figure D.5: Block diagram of the watermark embedding scheme based on HVS and wavelets.

The embedding algorithm is briefly described as follows. The RGB image is first converted to YCrCb colour image. The luminance component Y is decomposed by wavelet transform in 4-level of decomposition. The wavelet coefficients in each subband are grouped into 3-by-3 non-overlapping blocks. The HVS weights are then calculated for each blocks by the HVS weighting function. Based on the HVS weights, the algorithm decides which blocks are suitable for embedding watermark. The wavelet coefficients in selected coefficient blocks are used to train the general *regression neural network* (GRNN). The watermark bits which are scrambled by a scrambling technique with a secret key are embedded to selected coefficients by the trained GRNN. Finally, inverse wavelet transform IDWT is applied to reconstruct the watermarked image.

The intensity display of 4-level Daubechies-2 wavelet decomposition of Lena image is shown in Fig. D.6.

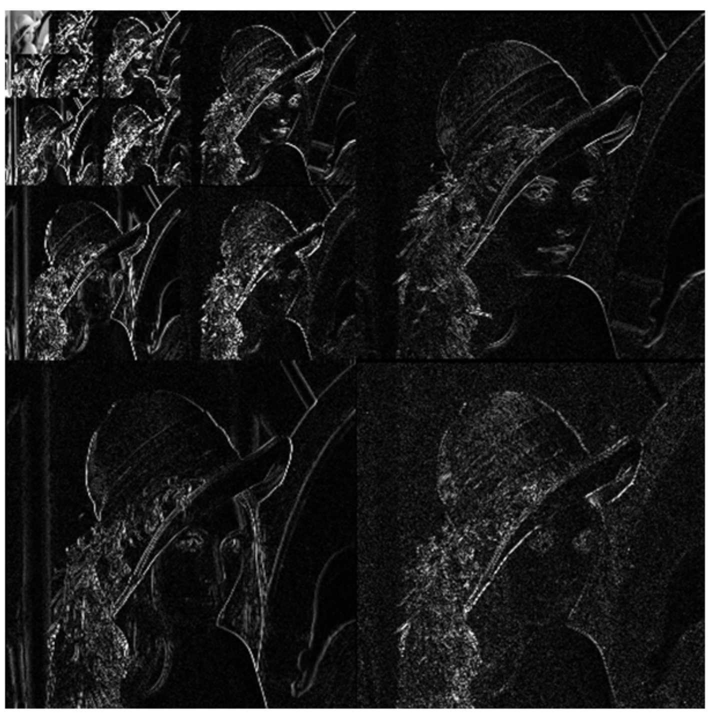


Figure D.6: Intensity display of 4-level Db2 wavelet decomposition of Lena image.

The HVS model calculates the HVS mask for embedding the watermark in 4-level wavelet transformed domain by calculating the weighting function for each block, $Wf(r,s,x,y)$. we denote $I^{(r,s)}(x,y)$, as the central wavelet coefficient value of the block $B^{(r,s)}(x,y)$ at subband s, level r. For 4-level wavelet decomposition, $r = 1,2,3,4$, and s is the subband of HH, HL, LD, LL. The HVS weighting function is then calculated from

$$Wf(r,s,x,y) = F(r,s) * L(r,x,y) * T(r,x,y)^{\alpha} \tag{D.7}$$

where α is the weighting parameter, deciding the perceptual quality of HVS mask. In this work, we choose $\alpha = 0.17$. The functions $F(r,s)$, $L(r,x,y)$, and $T(r,x,y)$ are calculated by Eq. (D.8), Eq. (D.11), and Eq. (D.12), respectively.

$$F(r,s) = f_1(s) * f_2(r) \tag{D.8}$$

where

$$f_1(s) = \begin{cases} \sqrt{2}, & \text{if } s = HH \\ 1, & \text{otherwise} \end{cases} \tag{D.9}$$

$$f_2(r) = \begin{cases} 1.00, & \text{if } r = 1 \\ 0.32, & \text{if } r = 2 \\ 0.16, & \text{if } r = 3 \\ 0.10, & \text{if } r = 4 \end{cases} \tag{D.10}$$

$$L(r,x,y) = \frac{1}{256} I_{LL}^{4} \left(1 + \left|\frac{x}{2^{4-r}}\right|, 1 + \left|\frac{1}{2^{4-r}}\right|\right) \tag{D.11}$$

$$\begin{aligned} T(r,x,y) \;=\; & \sum_{k=1}^{4-r} 16^{-k} \sum_{s}^{HH,HL,LH} \sum_{i=0}^{1}\sum_{j=0}^{1} \left(I^{k+r,s}\left(i + \frac{x}{2^k}, j + \frac{y}{2^k}\right)\right)^2 * \\ & Var\left\{ I_4^4 \left(1 + i + \frac{x}{2^{4-r}}, 1 + j + \frac{y}{2^{4-r}}\right)\right\} \end{aligned} \tag{D.12}$$

The suitable blocks (HVS mask) for embedding the watermark are selected by comparing the center coefficient values $I^{(r,s)}(x,y)$ with its corresponding weighting function value $Wf(r,s,x,y)$, given by

$$B^{(r,s)*}(x,y) = \left\{ B^{(r,s)}(x,y) : I^{(r,s)}(x,y) > \frac{1}{4} Wf(r,s,x,y) \right\} \tag{D.13}$$

The process of training GRNN to embed the watermark bits into the selected wavelet coefficients has the same procedure described in Section 4.3 of Chapter 4. The watermark extraction process is the inverse of the embedding process. In the experiment, the binary logo watermark image of "Winnipeg Jet" is embedded into the colour images of Lena, Baboon, Airplane, and House, with the same watermarking factor $\eta = 18$. The watermarked image of Lena and Baboon are shown in Fig. D.7 and Fig. D.8, respectively. More experimental results can be found in [DaKi12a].

Figure D.7: Watermarked image of Lena: (a) The original Lena image; (b) The watermarked image with PSNR=42.83 dB.

PSNR=42.90 dB

(a) (b)

Figure D.8: Watermarked image of Baboon: (a) The original Baboon image; (b) The watermarked image with PSNR = 42.90 dB.

Appendix E

Possible Research Extensions

E.1 Effective fitness evaluation and global-local learning strategy of AMMOA for many-objective optimization problems

The implementation used Pareto ranking and crowding distance for updating the population can work well with a two- or three-objective optimization problems. However, it scales poorly with many objective problems because of the nature of using the Pareto dominance principle for ranking. Thus, to improve the performance for many objective optimization problems (e.g., more than 4 objectives), AMMOA needs to adopt an efficient ranking method that can improve the convergence and preserve the diversity of populations well.

We have proposed an information theoretic criterion for guiding the selection, clustering, and local learning processes. This criterion can be used for ranking the individuals in the population for updating processes. This information criterion based ranking can improved the convergence well; however, it is poor in preserving the diversity of the population. Thus, a careful study of using this information theoretic criterion with supporting methods for an effective elitist strategy needs to be placed in our priority for the future research.

The issue of adopting a right strategy for local learning and global learning in memetic optimization algorithms is always a challenging problem. In the implementation of AM-MOA in Chapter 3, we used a strategy based on experimental results for the local learning and global learning. In particular, the probability of the local search 1 in one simulation run was selected based on the experimental results. In our future research, this issue should be carefully investigated for an adaptive problem driven strategy.

E.2 FPGA based memetic evolvable hardware for cognitive systems

Evolvable hardware combines together reconfigurable hardware, computational intelligence, fault tolerance, and autonomous systems. Evolvable hardware uses simulated evolution to search for new hardware configurations. The evolution is performed by a variety of different stochastic search algorithms such as evolutionary algorithms [CBBC11, GrTy07, GoBe02]. The evolvable hardware is implemented on reconfigurable devices such as *field programmable gate arrays* (FPGA), *field programmable analog arrays* (FPAA), or *field programmable transitor arrays* (FPTA) [GrTy07]. Each device is configured to define its best architecture by itself for the given application.

This research area has achieved important progresses in the last decade; however, because of the complexity, it still stops at simple example demonstrations. It should start dealing with more complex and real-world applications. Thanks to the fast development of FPGA technology, FPGAs have changed to a powerful reconfigurable devices with more resources and features, such as dynamic partial reconfiguration makes FPGAs be able to implement different types of evolution. The principle of adaptive memetic algorithm we developed for AMMOA is a good candidate for FPGA based memetic evolvable hardware because of its fast convergence and adaptiveness. It is necessary to scale down the complexity and the size of AMMOA when applying it for a memetic evolvable hardware. It can be applied for both single-objective and multiobjective applications. With a single-objective, the number of clusters in the clustering process of AMMOA will be scaled down to 0, and the non-dominated sorting procedures will be applied with the probability of 0. Each evolution in this situation just includes simple global and local searches for a single-objective function.

The autonomous navigation of spacecrafts is one of the potential applications of the memetic evolvable hardware. This is an important feature of future space missions. An intelligent and autonomous spacecraft must be able to adapt to new environments and be able to deal with unexpected changed situations, fast and robustly. Thus, adaptive electronic hardware that is able to reconfigure its electronics circuits by itself is a special need for autonomous spacecrafts.