College of Engineering, Mathematics and Physical Sciences

Hybrid Optimisation Algorithms for Two-Objective Design of Water Distribution Systems

Submitted by Qi Wang to the University of Exeter

as a thesis for the degree of

Doctor of Philosophy in Engineering

in May 2014

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signature: ....

ABSTRACT

Multi-objective design or extended design of Water Distribution Systems (WDSs) has received more attention in recent years. It is of particular interest for obtaining the trade-offs between cost and hydraulic benefit to support the decision-making process. The design problem is usually formulated as a multi-objective optimisation problem, featuring a huge search space associated with a great number of constraints. Multi-objective evolutionary algorithms (MOEAs) are popular tools for addressing this kind of problem because they are capable of approximating the Pareto-optimal front effectively in a single run. However, these methods are often held by the “No Free Lunch” theorem (Wolpert and Macready 1997) that there is no guarantee that they can perform well on a wide range of cases.

To overcome this drawback, many hybrid optimisation methods have been proposed to take advantage of multiple search mechanisms which can synergistically facilitate optimisation. In this thesis, a novel hybrid algorithm, called Genetically Adaptive Leaping Algorithm for approXimation and diversitY (GALAXY), is proposed. It is a dedicated optimiser for solving the discrete two-objective design or extended design of WDSs, minimising the total cost and maximising the network resilience, which is a surrogate indicator of hydraulic benefit. GALAXY is developed using the general framework of MOEAs with substantial improvements and modifications tailored for WDS design. It features a generational framework, a hybrid use of the traditional Pareto-dominance and the ε -dominance concepts, an integer coding scheme, and six search operators organised in a high-level teamwork hybrid paradigm. In addition, several important strategies are implemented within GALAXY, including the genetically adaptive strategy, the global information sharing strategy, the duplicates handling strategy and the hybrid replacement strategy. One great advantage of GALAXY over other state-of-the-art MOEAs lies in the fact that it eliminates all the individual parameters of search operators, thus providing an effective and efficient tool to researchers and practitioners alike for dealing with real-world cases.

To verify the capability of GALAXY, an archive of benchmark problems of WDS design collected from the literature is first established, ranging from small to large cases. GALAXY has been applied to solve these benchmark design problems and its achievements in terms of both ultimate and dynamic performances are compared with those obtained by two state-of-the-art hybrid algorithms and two baseline MOEAs. GALAXY generally outperforms these MOEAs according to various numerical indicators and a graphical comparison tool. For the largest problem considered in this thesis, GALAXY does not perform as well as its competitors due to the limited computational budget in terms of number of function evaluations.

All the algorithms have also been applied to solve the challenging Anytown rehabilitation problem, which considers both the design and operation of a system from the extended period simulation perspective. The performance of each algorithm is sensitive to the quality of the initial population and the random seed used. GALAXY and the Pareto-dominance based MOEAs are superior to the ε-dominance based methods; however, there is a tie between GALAXY and the Pareto-dominance based approaches.

At the end, a summary of this thesis is provided and relevant conclusions are drawn. Recommendations for future research work are also made.

ACKNOWLEDGEMENTS

First of all, I would like to express my faithful appreciation to my first supervisor Professor Dragan Savić, who encouraged me to pursue this PhD degree in the Centre for Water Systems at the University of Exeter. I am really moved by all his support and considerations throughout the four years of my study. Not only did I obtain an in-depth understanding of my topic under his guidance, but I was also influenced by his invaluable thinking style and characteristics.

I also thank my second supervisor Professor Zoran Kapelan, who provided very useful comments and suggestions for my research work. His wide knowledge of water engineering disciplines and highly efficient working style really impressed me.

I would like to acknowledge the financial support received from both the University of Exeter and the China Scholarship Council, which contributed towards the successful completion of this thesis.

The kind support and assistance offered by the people working in the Centre for Water Systems is also acknowledged. In particular, I wish to express my special gratitude to Drs, Michele Guidolin, Mark Morley, Kent McClymont, Innocent Basupi and Maryam Astaraie-Imani who inspired and facilitated my work. In addition, I gratefully thank Dr Michael Hammond and Mrs. Alexandra Slater for kindly proof-reading this thesis and correcting the English.

I will always remember the generous help received from my Chinese friends at the University of Exeter. In particular, I acknowledge Dr. Dali Kong for providing computing resources and prompt feedback on programming. Also, I thank Ms. Fanlin Meng and Mr. Haixing Liu for their care and patience especially during my feelings of depression.

Finally, I would like to appreciate my family and my fiancé Xiaoqing Yuan for their love, patience and understanding during these years. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AF	Pareto Approximation Front
AMALGAM	A Multi-Algorithm, Genetically Adaptive Multi-Objective Method
$AMALGAM_{ndu}$	A Variant of AMALGAM with NSGA-II, DE and UMDA as Sub-Algorithms
$AMALGAM_{ndug}$	A Variant of AMALGAM with NSGA-II, DE, UMDA and GREEDY as Sub-Algorithms
$AMALGAM_{njg}$	A Variant of AMALGAM with NSGA-II, JG and GREEDY as Sub-Algorithms
$AMALGAM_{njgp}$	A Variant of AMALGAM with NSGA-II, JG, GREEDY and PSO as Sub-Algorithms
AMS	Adaptive Metropolis Search
AS	Pareto Approximation Set
BAK	BakRyan Network
BIN	Balerma Irrigation Network
BLA	Blacksburg Network
CNSGA-II	Controlled Non-Dominated Sorting Genetic Algorithm II
CSM	Chi-Square Matrix Approach
CWS	Centre for Water Systems
DACE	Design and Analysis of Computer Experiments Model
DC	Dither Creeping
DDM	Demand Driven Model
DE	Differential Evolution
DV	Decision Variables
D-W	Darcy-Weisbach
EA	Evolutionary Algorithm
EAF	Empirical Attainment Function
ENGA	An Enhanced Non-Dominated Sorting Genetic Algorithm
EOS	Evolutionary Optimisation System
EPS	Extended Period Simulation
ε-MOEA	ε-Multi-Objective Evolutionary Algorithm

ε-NSGA-II	ε-Dominance Non-Dominated Sorted Genetic Algorithm
EXN	Exeter Network
FOS	Fossolo Network
GALAXY	Genetically Adaptive Leaping Algorithm for Approximation and Diversity
GM	Gaussian Mutation
GOY	GoYang Network
GREEDY	A Population-based Greedy Heuristic Algorithm
HAN	Hanoi Network
hBOA	Hierarchical Bayesian Optimisation Algorithm
H-W	Hazen-Williams
JG	Jumping-Gene Evolutionary Algorithm
kWh	Kilowatt-Hour(s)
LEMMO	Learnable Evolution Model for Multi-Objective Optimisation
LP	Linear Programming
MEX	Matlab Executable Function
ML	Multiple Loading Conditions
MOCE	Multi-Objective Cross Entropy
MOD	Modena Network
MOEA	Multi-Objective Evolutionary Algorithm
MOGA	Multi-Objective Genetic Algorithm
MOHO	Multi-Objective Hybrid Optimisation
MOO	Multi-Objective Optimisation
MS	Multiple Sources
ND	Number of Decision Variables
NFE	Number of Function Evaluation
NLP	Non-Linear Programming
NPGA	Niched Pareto Genetic Algorithm
NSDE	Non-Dominated Sorting Differential Evolution
NSGA	Non-Dominated Sorting Genetic Algorithm
NSGA-II	Non-Dominated Sorting Genetic Algorithm II

NSGA-II-JG	Jumping-gene Genetic Algorithm Based on Non-Dominated Sorting Genetic Algorithm II
NYT	New York Tunnel Network
PAES	Pareto Archived Evolution Strategy
ParEGO	An Extension of the Single-Objective Efficient Global Optimization Algorithm for Pareto Optimization
PCX	Parent-Centric Crossover
PDDM	Pressure Dependent (or Driven) Demand Model
PES	Pescara Network
PESA-II	Pareto Envelope Based Selection Algorithm II
PF	Pareto-Optimal Front
PM	Polynomial Mutation
PS	Pareto-Optimal Set
PSO	Particle Swarm Optimisation
PV	Present Value
PVC	Polyvinyl Chloride
RF	Pareto Reference Front
SBX	Simulated Binary Crossover
SBXI	Simulated Binary Crossover for Integers
SGA	Simple Genetic Algorithm
SL	Single Loading Condition
SMGA	Structured Messy Genetic Algorithm
SOM	Self-Organising Map
SPEA	Strength Pareto Evolutionary Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm 2
SPX	Simplex Crossover
SS	Single Source
TF	Turbulence Factor
TLN	Two Loop Network
TRN	Two Reservoir Network
UM	Uniform Mutation
UMDA	Univariate Marginal Distribution Algorithm

UNDX	Unimodal Normal Distribution Crossover
WDS	Water Distribution System
WDSBA	Water Distribution System Benchmark Archive

LIST OF NOTATIONS

$\lfloor * \rfloor$	an operator that rounds an element toward negative infinity
$\lvert * \rvert$	cardinality of *AF*
$\otimes$	a vector based product
A	resistance coefficient
$a \succ= b$	solution *a* weakly dominates solution *b*
B	flow exponent
C	cost
c_1, c_2	weights of cognitive and social factors of each particle
$C_{Capital}$	capital costs
C_j	uniformity of node *j*
c_n	jump rate which determines the spread of solutions around X_t^i
C_{Tank}^{New}	capital costs of new tanks if any
$C_{Operational}$	operational costs
$C_{Operational}^{Daily}$	daily operational costs
$\text{cov}(X_t^i, X_t^i)$	covariance of the best non-dominated solutions at generation *t*
C_{Pipe}^{Clean}	capital costs of existing pipes which are to be cleaned and lined
$C_{Pipe}^{Duplication}$	capital costs of duplication pipes
C_{Pipe}^{New}	capital costs of new pipes (including risers for new tanks if any)
C_{Total}	total costs
C_u	unit cost corresponding to a specific diameter option
D	set of diameters (or diameter options)

D_i	diameter (or diameter option) of pipe *i*
D_i^j	diameter of pipe *i* connected to demand node *j*
d_i^j	Euclidean distance from solution *i* in the *AF* to the solution *j* in the *RF*
$D^{\min}, D^{\max}$	minimum and maximum diameter options for each pipe, respectively
$f(\kappa, d, q)$	Darcy friction factor
$f_i^{\min}, f_i^{\max}$	minimum and maximum values of the *i*-th objective, respectively
g	equality constraints
H	hydraulic head
H_i	head at node *i*
h_i^{end}, h_i^{start}	end and initial levels of tank *i*, respectively
$h_i^{\min}, h_i^{\max}$	minimum and maximum operation levels of tank *i* during 24 hours, respectively
$H_i^{\min}, H_i^{\max}$	minimum and maximum head requirement at node *i, respectively*
$H_j^{i,t}$	pressure at demand node *i* at time step *t* under loading condition *j*
H_k	actual head of reservoir *k*
H_t	hydraulic head of tank *t*
I_c	binary coverage indicator
I_{EP}	ε performance indicator
I_{GD}	generational distance indicator
I_{HV}	hypervolume indicator
I_{mpd}	maximum pressure deficit
I_n	network resilience

I_r	resilience index
I_{tpd}	total pressure deficit
$I_{\varepsilon+}$	additive ε indicator
J	set of demand nodes
K, F	uniformly generated numbers between (0.2, 0.6) and (0.6, 1.0), respectively
L_i	length of pipe *i*
max	maximisation of a given dataset
min	minimisation of a given dataset
N	population size
n	number of objectives
N/A	not applicable
N_d	number of diameter options available
NI	node index
nl	number of loading conditions
nn	number of demand nodes
norm	normalisation of a given variable
np	number of pipes to be optimised
npj	number of pipes connected to node *j*
npl	number of pipes in loop *l*
npu	number of pumps
nr	number of reservoirs
ns	number of time steps under loading condition *j*
nt	number of tanks in the system
N_t^i	number of offspring to be generated by operator *i* at generation *t*

o	inequality constraints reflecting the operational requirements, such as pressure head at each demand node and flow velocity in a pipe
P	set of pipes
P_d	conditional probability of downward variation
P_{dcm}	probability of dither creeping mutation
$P_{dcm}^{min}, P_{dcm}^{max}$	lower and upper bounds of probability of dither creeping mutation
P_i	power of pump *i*
P_t^i	number of offspring contributed by operator *i* at generation *t*
Q	water consumption at demand nodes
Q_i^{in}	water flowing into node *i*
Q_i^{out}	water flowing out of node *i*
Q_j	water demand at node *j*
Q_k	discharge of reservoir *k*
Q_t	amount of flow leaving tank *t*
r_1, r_2	uniformly distributed random numbers between (0, 1)
$rand$	a uniformly distributed random number sampled between (0,1)
$randi$	a uniformly distributed random integer between [a, b]
$randn$	a normally distributed random number sampled between (0,1)
R_T	a uniformly distributed random number sampled between [-1, 1]
u	uniformly sampled random number between (0, 1)
UB, LB	vectors of upper and lower bounds of decision variables
V	velocity of water in pipes
V_i	flow velocity in pipe *i*
$V_i^{\max}$	maximum flow velocity requirement of pipe *i*

V_t^i	velocity of particle *i* at generation *t*
X_t^a, X_t^b, X_t^c	three randomly selected individuals from current population at generation *t*, and they must be different from each other and from X_t^i
X_t^i	current best non-dominated solution at generation *t*
α^a, α^b	variables calculated based on the selected parents (X_t^a and X_t^b), UB, LB and η_c
$\beta(\alpha)$	a vector randomly generated, given a uniformly sampled random number u between (0, 1) and a distribution index of η_c for Simulated Binary Crossover
γ	specific weight of water
δ	a vector of small variations which are obtained from a polynomial distribution of a given uniformly sampled random number u between (0, 1)
ΔH	head loss in a pipe
ΔH_i^l	head loss in pipe *i* within loop *l*
ε_i	side length of each box in the *i*-th objective
ε_{z^1,z^2}	maximum differences among *n* objectives of two selected solutions, one from the *AF* and the other from the *RF*
ε_{z^2}	maximum differences among *n* objectives of the nearest solution in the *AF* to a given solution z^2 in the *RF*
η_c	distribution index for Simulated Binary Crossover
η_m	distribution index of Polynomial Mutation
κ	roughness height
σ	a scaling factor which is equal to $\frac{(D^{\max} - D^{\min} + 1)}{10}$
ω	inertia factor

1. INTRODUCTION

1.1. Background and Motivation

Water Distribution Systems (WDSs) play a crucial role in a modern city. A WDS distributes potable treated water to households, and commercial and industrial users. Water delivered via a WDS should not only be of adequate quantity and pressure, but also satisfy water quality standards. Building a new WDS, especially a large network, always incurs large (capital) expenditures. Therefore, it is well understood that sufficient consideration of financial and technical issues should be taken during the course of WDS design.

Failures can happen frequently in an existing WDS over time as components deteriorate (e.g., the ageing of pipes due to corrosion). During the operation of a WDS, continuous attention should be paid to the status of its components, as their failure, either hydraulic or structural, would result in the interruption of water supply, thus decreasing the service level of a water company. Replacing existing pipes and/or adding new pipes, can improve the performance of a WDS. However, funding is often strictly limited and thus a cost-efficient plan is required.

Historically (even today in some cases), the design of WDSs mainly relied on engineers' knowledge and experience. Nevertheless, this is not sufficient to cope with large networks, which are common to major cities nowadays. As optimisation algorithms, for instance, linear programming (LP), non-linear programming (NLP) and evolutionary algorithms (EAs), emerged, it became popular to formulate the task as an optimisation problem and solve it using these tools. This trend increased particularly during the late 1960s and 1970s and quite a few applications can be found in the literature from then on (Schaake and Lai 1969; Shamir 1974). On the other hand, various hydraulic models were proposed to simulate the behaviour of a WDS. These mathematical models were able to capture the hydraulic and water quality characteristics of a WDS in steady state or extended period. EPANET (Rossman 2000) is one of these packages which is open-source and regarded as an "industry standard" software tool in this field. For this reason, it is used throughout this thesis (via its Programmer's Toolkit) for the hydraulic simulations.

Different formulations have been presented to account for the optimal design of WDSs. Previously, single-objective formulations, the so called least-cost design, were considered, and focused on minimising the capital cost. The conservation of mass and energy of the network system and the pressure requirement of demand nodes were treated as the constraints. But later, this formulation was often criticised and its drawbacks were highlighted (Engelhardt et al. 2000; Walski 2001) as it did not reflect the true concerns of decision makers and the criteria whereby WDSs are usually assessed. Multi-objective, particularly two-objective, formulations have therefore received more attention, since they properly reflect the nature of engineering design, i.e., minimising the cost and maximising the system performance (e.g., reliability). Constraint violation (e.g., nodal pressure shortfall) was once considered as the other objective and thus forming an unconstrained, multi-objective optimisation (MOO) problem. However, this formulation did not address the issues raised in the single-objective formulation. Furthermore, the optimal solutions following this formulation cannot ensure a reliable network under abnormal conditions (e.g., pipe bursts), which is critical to WDSs. As improvements in constraint handling techniques have been developed for MOO (Deb et al. 2002), modern algorithms are able to deal with constrained optimisation problems. Hence, more insights into the trade-offs between the costs and the benefits (like reliability) can be explored and analysed.

At present, it is a common practice to apply optimisation algorithms to solve the problem of WDS design using a hydraulic simulation model. Especially, EAs (e.g., Genetic Algorithms) have received more attention as they are easy-to-use, flexible and have been successfully applied to solve a wide range of problems in other fields. Meanwhile, the realistic concerns have considerably driven the broad applications of multi-objective evolutionary algorithms (MOEAs) to the design problems.

Nevertheless, the main criticisms of evolutionary algorithms (both single and multi-objective versions) include their robustness (especially on large problems), accuracy and parameterisation issues. That is, these algorithms usually require a large number of function evaluations and multiple runs to find the near-optimal solution(s). Additionally, individual parameters of these algorithms should be fine-tuned, which is generally based on rules of thumb and/or the trial-and-error approach, thus being computationally expensive especially for large and/or complex problems.

On the other hand, most applications of evolutionary algorithms in the literature have focused on a few small-to-medium sized problems, which were inadequate to meet real-life requirements. In other words, there is a gap between the capability of state-of-the-art optimisation algorithms and their performance on large and complex WDS design problems. Although some medium-to-large sized problems were introduced, relevant data are often unavailable for the community of researchers and practitioners alike to test their own algorithms on these problems. Therefore, there is a lack of standardised benchmark tests (including both cases and formulations) for the optimal design of WDSs.

Recently, several hybrid algorithms have been proposed in order to improve the effectiveness and efficiency of multi-objective optimisation by combining different algorithms and/or strategies. The comparative studies (Raad et al. 2011; Hadka and Reed 2013; Reed et al. 2013) have shown that they can outperform the mainstream MOEAs on various test functions and benchmark problems in other fields. However, an in-depth study on these hybrid algorithms for the optimal design of WDSs is still missing, which motivates the work carried out in this thesis.

This thesis investigates existing hybrid frameworks and aims to develop an improved hybrid algorithm in order to further increase the capacity of current optimisation algorithms for the optimal design of WDSs. A WDS benchmark archive (WDSBA) is set up to facilitate the comparison of these algorithms systematically and comprehensively, collecting up to twelve networks from the literature. A two-objective formulation, i.e., minimising the cost and maximising a surrogate indicator of hydraulic benefit, is adopted throughout the experiments. The proposed hybrid algorithm and other referential algorithms including hybrid algorithms and state-of-the-art MOEAs are examined and compared on the problems in the WDSBA, considering both ultimate and dynamic performances.

1.2. Aims and Objectives

The main objective of this thesis is to improve the current hybrid frameworks for the two-objective design or extended design of WDSs. Here, the term “hybrid framework” refers to an algorithm that uses diverse search operators (population-based EAs and other types) and different strategies simultaneously. More specifically, each detailed objective is described as follows.

- To investigate the state-of-the-art hybrid frameworks in the literature and to identify their advantages and disadvantages when applied to the optimal design or extended design of WDSs. These hybrid algorithms include both EA and non-EA based approaches.
- To develop an improved hybrid algorithm by tailoring the general framework of MOEAs systematically for the discrete nature of WDS design and by reconsidering the hybrid paradigm from the viewpoints of exploration and exploitation.
- To set up an archive of benchmark WDS design or extended design problems (as many as possible) collected from the literature. A uniform formulation, focusing on two objectives constrained optimisation, is generalised throughout these benchmark problems. It will facilitate the comparative study on the proposed hybrid algorithm and other reference algorithms.
- To apply the proposed hybrid algorithm to solve the problems in the WDSBA, and to compare its performance with state-of-the-art hybrid algorithms and other MOEAs in terms of both during and at the end of the optimisation process. These will be referred to as dynamic and ultimate performance respectively. Different computational budgets will be considered in accordance with search space sizes of these problems.

In addition, for hybrid algorithms the dynamic variations of search operators will be monitored and analysed to explain the performance of these methods.

- To apply all the algorithms considered in this thesis to solve the Anytown benchmark problem (Walski et al. 1987), which represents a challenging extended design problem as it involves operational considerations (i.e., pump scheduling). The performance on this difficult problem will further demonstrate their potential for solving more complex problems.

In addition, special attention is given in this thesis to the following aspects:

1. The best-known Pareto-optimal front (PF) or the true PFs (for small design problems) are obtained in this work for each of the benchmark problems. These PFs will be available in the public domain and can be used as the reference set for the purpose of quantitative comparison.

2. Diagnoses of the failure of hybrid algorithms on some cases are given. The reasons for the deterioration are taken into account when developing the improved hybrid framework.

1.3. Layout

This thesis is organised into 7 chapters including this introduction.

In Chapter 2 a literature survey is presented. Firstly, a review of multi-objective optimal design of WDSs is given. Then the reviews of hybrid optimisation algorithms, particularly the EAs based ones, are provided. Finally, a review of the performance assessment of multi-objective optimisation is presented.

In Chapter 3 the proposed hybrid algorithm is developed step by step based on the general framework of MOEAs. The developments include the selection of algorithmic framework, dominance concept, variable coding scheme, search operators and paradigm of hybridisation. In addition, some important strategies implemented in the proposed hybrid method are explained. Then, a sensitivity analysis is carried out to verify the robustness of one search operator, which contributes to eliminating the individual parameters of the proposed algorithm. A brief summary is given at the end of this chapter.

In Chapter 4 a WDS benchmark archive is set up by collecting as many as possible networks in the public domain. Firstly, a generalised formulation considering two objectives, i.e., minimising the cost and maximising a surrogate indicator of hydraulic benefit, is presented. Then, twelve benchmark networks are introduced in ascending order of search space size following a uniform format. The features and classification of these networks are also discussed.

In Chapter 5 the proposed hybrid algorithm, as well as the other state-of-the-art MOEAs, are applied to solve the problems in the WDSBA. The experimental setup is described firstly, including (1) a variety of numerical indicators and a graphical tool for performance evaluation; (2) computational budgets in terms of the number of function evaluations (NFEs) for different cases, and (3) the algorithmic setup, i.e., the configurations of individual parameters for each algorithm. Next, multiple independent runs using these algorithms with the same NFEs are carried out on each benchmark problem. Their attainments, both ultimate and dynamic, are compared and discussed. Finally, a summary is provided and relevant conclusions are drawn.

In Chapter 6 all the algorithms considered in the thesis are applied to solve the Anytown rehabilitation problem, which is based on an extended period simulation model. First, a brief history of this problem is given. Then, the formulation is introduced, which is adapted from the one presented in Chapter 4 as the operation of the Anytown network is taken into account. After that, the problem is solved by each algorithm via multiple runs, and the results obtained are compared and discussed. In the end, a summary is given and relevant conclusions are drawn.

In Chapter 7 the key findings of the thesis are summarised and relevant conclusions are drawn. Finally, recommendations for future research work are provided.

2. LITERATURE REVIEW

2.1. Introduction

In this chapter a literature review of multi-objective design or the extended design of WDSs is first presented. Then, hybrid optimisation algorithms, especially those that appear superior to state-of-the-art MOEAs, are reviewed. In addition, a review of current methods for the performance assessment of multi-objective optimisation is covered as it plays an essential role in the comparison of different algorithms.

In Section 2.2 the design of WDSs is posed as a constrained MOO problem. Both general and specific problem formulations are given from a mathematical viewpoint. Then, a review of the applications of optimisation methods to the design of WDSs is provided. Some important issues related to this topic are also discussed.

Hybrid optimisation algorithms, as the main focus of this thesis, are reviewed in more detail in Section 2.3, including both EA and non-EA based approaches. A concise introduction to hybrid optimisation algorithms is first presented. After that, several state-of-the-art multi-objective hybrid algorithms are reviewed individually. At the end of this section, the advantages and disadvantages of current hybrid algorithms are given, and some important issues, e.g., robustness and parameterisation, are discussed.

In Section 2.4 a review of various methods for the performance assessment of MOO is given. It covers both quantitative indicators and qualitative/graphical based approaches. In addition, the main characteristics and suitability of these methods are discussed.

Section 2.5 summarises this chapter.

2.2. Multi-Objective Optimal Design of WDSs

2.2.1. Introduction

A concise history of the multi-objective design or extended design of WDSs is provided in this section. Specifically, the focus is on deterministic design applications using evolutionary algorithms, since this represents a major concern in the literature. Probabilistic design methods are briefly discussed in Section 2.2.4. In this thesis, the word “design” refers to the determination of the sizes of all the pipe components given the configuration (layout) of a network system. In contrast, the word “extended design” or “rehabilitation” refers to the determination of the potential intervention options for a subset of pipe components and the associated sizes. Note that the words “extended design” and “rehabilitation” are used interchangeably hereafter.

In fact, the design of a WDS usually involves the consideration of many criteria (e.g., cost, pressure, reliability or even water quality), which are normally in conflict with each other, and thus, design is intrinsically multi-objective. Although Gessler (1985) presented solutions as the trade-off between nodal pressure and total cost, Halhal et al. (1997) arguably made the first attempt to optimise the rehabilitation problem of WDSs in the MOO sense. Halhal et al. (1997) addressed the problem of minimising capital cost and maximising a combination of benefits using the structured messy genetic algorithm (SMGA).

Since the introduction of various MOEAs from the late 1980s, these algorithms have been gradually adopted for solving the WDS design or rehabilitation problems. Cheung et al. (2003) tested Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele 1999) and Multi-Objective Genetic Algorithm (MOGA) (Fonseca and Fleming 1993) on the same problem tackled by Gessler (1985). SPEA was observed to outperform MOGA in terms of the quality of solutions obtained and time required. Farmani et al. (2003b) discussed the advantages and disadvantages of four MOEAs and compared their performance on two benchmark problems. They concluded that the elitist Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al. 2002) was a viable tool to find Pareto optimal solutions. Keedwell and Khu (2003) proposed a hybrid algorithm based on the well-known NSGA-II and solved the New York Tunnels problem (Schaake and Lai 1969) in an effective way. A local search procedure was incorporated into NSGA-II by increasing and decreasing the sizes of two pipes in small steps until a dominating solution was found. Other MOEAs were also used for the design or rehabilitation purposes, like SPEA2 (Farmani et al. 2005a), Multi-Objective Cross Entropy (MOCE) (Perelman et al. 2008) and a class of algorithms based on the concept of probabilistic model building (Olsson et al. 2009). An incomplete survey of multi-objective design or rehabilitation of WDSs via MOEAs is given in Section 2.2.3.

Recently, there is a growing trend of applying hybrid algorithms in the context of MOO to WDSs. Besides the early work presented by Keedwell and Khu (2003), Raad et al. (2008) employed the jumping gene adaptation of NSGA-II to the design of WDSs, minimising cost and maximising network resilience (Prasad and Park 2004). The modified version of NSGA-II was able to find better solutions which were highly resilience and had low costs. Di Pierro et al. (2009) applied two versions of hybrid algorithms, taking advantage of machine learning techniques, to optimise the design of two real-world networks of medium and large size. They showed that these hybrid methods, especially Learnable Evolution Model for Multi-Objective optimisation (LEMMO), were promising in bridging the gap between the quality of solutions obtained and the computational effort invested. They were capable of approximating the Pareto-optimal front (PF) with a significant reduction of function evaluations (hydraulic simulations). Raad et al. (2009) were the first to introduce the hyper-heuristic approach, known as A Multi-Algorithm, Genetically Adaptive Multi-objective (AMALGAM) method (Vrugt and Robinson 2007), to the design of WDSs. Although different from its original form, this version of AMALGAM demonstrated the best performance with regard to a number of benchmark problems. This point was further highlighted by solving a large real-world rehabilitation problem (Raad et al. 2011). Norouzi and Rakhshandehroo (2011) combined the concept of a Self Organising Map (SOM), a versatile unsupervised artificial neural network (Kohonen 1990), with NSGA-II and tested this SOM-NSGA-II on two small benchmark networks, minimising cost and maximising resilience index (Todini 2000). They claimed that SOM successfully discovered similarities between individuals of the population and could benefit the crossover operator, which resulted in a faster convergence or a wider diversity.

Most recently, Wang et al. (2014b) compared two types of high-level hybrid algorithms, namely the original AMALGAM and the multi-objective hybrid optimisation (MOHO) (Moral and Dulikravich 2008), on a range of benchmark problems collected from the literature. They found that AMALGAM and MOHO may suffer from deterioration in performance especially on large and complex problems. In contrast, Creaco and Franchini (2013) developed a low-level hybrid algorithm based on NSGA-II, which was shown to be dramatically superior to NSGA-II on four small to medium sized real-world design problems.

In short, various MOEAs have been applied to the multi-objective design or extended design of WDSs in the past two decades. For a comprehensive review of this area readers are referred to Raad (2011). He pointed out that a common problem in most of the aforementioned studies lies in the fact that only a small number of networks, if not relatively simple networks, were considered. This makes difficulty in drawing general conclusions about algorithmic performance. Actually, there is also a lack of widely accepted problem formulations when comparing these MOEAs. Some works formulated the design problems as unconstrained MOO problems, whereas others focused on constrained MOO problems. Furthermore, the effect of parameterisation of various MOEAs was not well addressed in the literature. However, it is without doubt that hybrid algorithms are promising for the optimisation of WDSs. Nevertheless, there is obviously a lack of systematic comparative studies among these hybrid methods, which is the main focus of this thesis.

2.2.2. Problem Formulation

General Problem Formulation

Real-world applications of optimal WDS design often involve a number of objective functions as mentioned in Section 2.2.1. In the literature, most relevant works focused mainly on the two-objective optimal design of WDSs, considering the minimisation of total costs and maximisation of system benefit (e.g., reliability). Therefore, this thesis follows this constrained, two-objective formulation and aims at finding the best trade-off between costs and system benefit for a given design problem. Fu et al. (2012a) firstly proposed a many-objective formulation of WDS design, in which up to six goals were considered simultaneously. However, it is out of the scope of this thesis.

A general problem formulation of the two-objective design or rehabilitation of WDSs (concerning only pipe components) is given in Eq. (2.1). It includes the objective functions, decision variables and a set of constraints. The main objectives are: (i) minimising the total capital cost and (ii) maximising the hydraulic benefit, which can be considered from diverse perspectives (see the subsequent section). The decision variables are usually the diameter settings (or diameter option indices) of all or a subset of pipes. The MOO model is subject to various constraints. In general, three types of constraints are considered: (1) bounds on decision variables (subject to availability in the market); (2) physical constraints such as conservation of mass and energy (handled implicitly by a hydraulic solver); and (3) operational constraints (regulation requirements).

$$
\begin{aligned}
&\min Cost &&= \textstyle\sum_{i=1}^{np} f_i(C_u, D_i, L_i) \\
&\max Benefit &&= f(D, H, Q) \\
&subject \quad to: && \begin{cases} D_i \in \{1, 2, \ldots, N_d\} \\ g(D, H, Q) = 0 \\ o_{\min} \leqslant o(H, V) \leqslant o_{\max} \end{cases}
\end{aligned}
\tag{2.1}
$$

Where, np - number of pipes to be optimised; C_u - unit cost corresponding to a specific diameter option; D_i - diameter of pipe i; L_i - length of pipe i; D - pipe diameters; H - hydraulic heads; Q - water consumption at demand nodes; N_d - number of diameter options available; V - velocity of water in pipes; g - equality constraints; o - inequality constraints reflecting the operational requirements, such as pressure head at each demand node and flow velocity in a pipe.

Specific Problem Formulation

Total Capital Cost

Constructing or upgrading a WDS requires a great amount of capital expenditure. The total capital costs (initial investment) of such a project are mainly comprised of the purchase, transportation and installation (including excavation) of pipe components. Obviously, these processes cannot be finished at one time, which implies that the expenditure occurs at different stages and time value of money (e.g., discount rate) should be considered.

In addition, a more realistic way to measure the total capital costs is to consider the life-cycle cost (converted to the corresponding net present value), which includes the initial cost of pipes, the cost of replacement of old pipes, the cost of cleaning and lining existing pipes and the expected repair cost for pipe breaks (Jayaram and Srinivasan 2008). However, various factors need to be taken into account in such a situation, and they may vary according to other reasons (e.g., the break rate of pipe varies according to materials and locations), which is out of scope of the thesis.

Hence, for the sake of simplicity, only the initial costs associated with pipe components are considered for the calculation of total capital costs.

Hydraulic Performance Benefit

Many aspects of a WDS can be considered during the design stage, such as system reliability and water quality. Among these concerns, system reliability is of great importance to most water companies and as such it has received attention in the past two decades.

Todini (2000) introduced the resilience index indicator (denoted as I_r) to optimise the design of looped municipal networks, based on the concept of "resilience" of a system under stressed or failure conditions. Generally speaking, a system that contains sufficient surplus power at each node is able to cope with possible failures (either mechanical failure, like pipe bursts or hydraulic failure due to changes in water demand), which inevitably results in increased internal energy dissipation. Increasing I_r will lead to improved network performance under abnormal conditions. Moreover, I_r is computationally efficient compared with other reliability based indicators, which require extensive statistical analyses. The definition of I_r is given in Eq. (2.2).

$$I_r = \frac{\sum_{j=1}^{nn} Q_j (H_j - H_j^{\min})}{(\sum_{k=1}^{nr} Q_k H_k + \sum_{i=1}^{npu} P_i / \gamma) - \sum_{j=1}^{nn} Q_j H_j^{\min}} \tag{2.2}$$

Where, I_r – resilience index; nn - number of demand nodes; Q_j, H_j and H_j^{min} - demand, actual head and minimum head of node j; nr - number of reservoirs; Q_k and H_k - discharge and actual head of reservoir k; npu - number of pumps; P_i - power of pump i; γ - specific weight of water.

However, using I_r as an indicator has some drawbacks: (1) the increase in value of this indicator does not necessarily result in the improvement of network reliability (Prasad and Park 2004); and (2) it cannot handle networks with multiple sources (Jayaram and Srinivasan 2008). Later on, Prasad and Park (2004) proposed an improved resilience based indicator, known as network resilience (denoted as I_n), which took both the nodal surplus head and the uniformity of pipes connected to that node into account. By comparing the results with former hydraulic benefit measures (I_r, minimum surplus head and total surplus head), they substantiated that the maximisation of I_n improved both the surplus power of demand nodes and continuity of size between adjacent pipes, which assisted a network in countering the stressed conditions (e.g., pipe breakage). The equation of I_n will be presented in Chapter 4.

Tanyimboh and Templeman (1993) proposed a flow entropy measure based on the informational entropy function (Shannon 1948). Flow entropy measures the uniformity of pipe flow rates across the network. It is better at representing multi-source networks and considers the reliability issue in a probabilistic way. The application of this indicator to a complex WDS design problem showed that the flow entropy can alleviate the drawbacks of using I_r and thus can be used as a surrogate reliability measure (Prasad and Tanyimboh 2008).

Recently, Raad et al. (2010b) compared four reliability surrogate measures, including flow entropy (Prasad and Tanyimboh 2008), I_r (Todini 2000), I_n (Prasad and Park 2004) and a mixed reliability surrogate (normalised flow entropy multiplied by I_r), and recommended I_n as the most practical measure for general WDS design. Besides being able to eliminate impracticable loops in the network, I_n was proved to be correlated to failure reliability in terms of demand satisfaction under pipe outage conditions.

Therefore, the network resilience indicator is used in this thesis to account for the hydraulic performance benefit of WDSs for design or rehabilitation purposes. EPANET software (Rossman 2000) is employed to run the hydraulic simulation, by which the variables required for the calculation of I_n are obtained.

2.2.3. Multi-Objective Optimisation Methods

As mentioned in Section 2.2.1, various MOEAs have been applied to address the optimal design or rehabilitation of WDSs in the past two decades. Table 2.1 gives a survey of the most representative works (comparative studies) in the literature. These studies are summarised chronologically according to four criteria, i.e., problem formulation (objectives), methodologies (algorithms), experimental setup (computational budget) and performance indicators. It is evident that (1) NSGA-II is still one of the most popular algorithms considered in the water community; and (2) there is a clear trend that more benchmark problems and more advanced methodologies (e.g., formulations, algorithms and performance indicators) are being considered in the research activities.

Table 2.1 Most representative studies of multi-objective optimal design or rehabilitation of WDSs in past two decades

Papers	Objectives[a]	Algorithms	Experimental Setup						Performance Indicators
			Type[b]	Computational Budget			No. of		
				NFE[c]	*Pop*[d]	*Gen*[e]	Runs	Cases	
(Wang et al. 2014b)	Min(C) Max(I_n)	AMALGAM MOHO NSGA-II	NFE	25,000-500,000	100	250-5,000	30	12	(1) ratio of hypervolume (2) number of solutions contributed (3) graphical comparison
(Creaco and Franchini 2013)	Min(C) Max(I_n)	A Low-Level Hybrid Algorithm NSGA-II	NFE	25,000-100,000	100	250-1,000	1	4	graphical comparison
(Raad et al. 2011)	Min(C) Max(I_n)	23 MOEAs	Time	N/A	64-256	N/A	30	9	(1) dominance rank (2) hypervolume metric
(Norouzi and Rakhshandehroo 2011)	Min(C) Max(I_r)	SOM-NSGA-II NSGA-II	NFE	N/A	N/A	100-200	N/A	2	(1) graphical comparison (2) generational distance
(Raad et al. 2010a)	Min(C) Max(I_n)	AMALGAM_{ndu} AMALGAM_{ndug} NSGA-II DE UMDA GREEDY	Time	N/A	100	N/A	30	4	(1) dominance rank (2) hypervolume metric
(Di Pierro et al. 2009)	Min(C) Min(I_{mpd})	ParEGO LEMMO PESA-II	NFE	873-600,000	100-400	100-2,000	10	2	(1) graphical comparison (2) size of Pareto front (3) hydraulic performance metrics
(Olsson et al. 2009)	Min(C) Min(I_{tpd})	NSGA-II UMDA	NFE	200,000-250,000	200-2,500	100-1,250	5-10	3	(1) graphical comparison (2) hypervolume metric

	Min(C) Min(I_{mpd})	hBOA CSM							(3) coverage metric
(Raad et al. 2009)	Min(C) Max(I_n)	$AMALGAM_{njgp}$ $AMALGAM_{njg}$ NSGA-II NSGA-II-JG GREEDY	NFE	10,000	100	100	20	4[f]	(1) graphical comparison (2) hypervolume
(Perelman et al. 2008)	Min(C) Min(I_{mpd})	MOCE NSGA-II	NFE	76,800- 3,360,000	200- 1,000	384- 3,360	30- 50	2	(1) graphical comparison (2) generational distance (3) distance measure (4) distribution measure
(Raad et al. 2008)	Min(C) Max(I_n)	NSGA-II JG	Time	N/A	100	N/A	30	4	(1) dominance count rank (2) hypervolume metric
(Farmani et al. 2005a)	Min(C) Min(I_{mpd})	NSGA-II SPEA2	NFE	100,000- 300,000	100	1,000- 3,000	5-20	3	(1) graphical presentation (2) binary epsilon-indicator (3) binary coverage indicator
(Nicolini 2004)	Min(C) Min(I_{mpd})	ENGA NSGA-II CNSGA-II	NFE	25,000	50	500	10	1	(1) convergence index (2) sparsity index
(Prasad and Park 2004)	Min(C) Max(I_n)	NSGA	NFE	100,000- 2,000,000	100-200	1,000- 10,000	1-3	2	graphical comparison
(Cheung et al. 2003)	Min(C) Min(I_{mpd})	MOGA SPEA	NFE	5,000- 100,000	100-500	50-200	30	1	(1) set coverage metric (2) graphical comparison (3) processing time
(Farmani et al. 2003b)	Min(C) Min(I_{tpd}) and	NSGA-II PAES[g] MOGA	NFE	200,000	200	1,000	2	2	graphical comparison

	Min(C) Min(I_{mpd})	NPGA							
(Keedwell and Khu 2003)	Min(C) Min(I_{tpd})	NSGA-II NSGA-II+Local Search	NFE	5,000-15,000	100-200	50-150	3	1	graphical comparison
(Todini 2000)	Min(C) Max(I_r)	Heuristic Approach	N/A	N/A	N/A	N/A	N/A	N/A	N/A
(Halhal et al. 1997)	Min(C) Max(*Benefit*)	SMGA SGA	NFE	5,000-25,000	40	125-625	3	2	graphical comparison

Note: [a]*C* in the objective function means Cost. [b]The types of optimisation include computational time based and number of function evaluations based. [c]*NFE* means number of function evaluations. [d]*Pop* means population size. [e]*Gen* means number of generations for optimisation. [f]The experimental settings were not applied to the real world case. [g]PAES did not follow the experimental setup specified in the table above, but it was allowed to evaluate the same number of trial solutions (i.e., 200,000). N/A means that the relevant information is not applicable. I_{mpd} means maximum pressure deficit. I_{tpd} means total pressure deficit. For the acronym of each algorithm, please refer to List of Abbreviations.

2.2.4. Other Issues of WDS Design

Demand Driven Model vs. Pressure Dependent Demand Model

It is well understood that the accuracy of a hydraulic simulation model is of great importance to its application, since a model which cannot simulate the real behaviour of a system would result in misguided decision making, thus wasting a huge amount of money and other resources. Most works in the literature address the design problems using a conventional hydraulic model, i.e., the demand driven model (DDM). A DDM (e.g., EPANET) assumes that water demand at each node is always satisfied, and then the equations of mass and energy conservation are solved to calculate the pressures at nodes and flow rates in links. Simulation using such kind of models, either in the steady-state or extended period simulation (EPS) condition, works well for a WDS under normal conditions, in which the minimum nodal pressures are satisfied. However, in real-world networks, the systems may suffer from pressure deficiencies due to various reasons, such as pump failures, pipe bursts, fire fighting or planned maintenance work causing the disconnection of major pipes (Siew and Tanyimboh 2012). The DDM cannot cope with these situations and thus may produce inaccurate results for the nodal demands and pressures. For instance, the entire nodal demand can be fulfilled with water under low or even negative pressure in a DDM, which is unrealistic and proves to be one of the drawbacks of a DDM.

To deal with this issue, many pressure dependent demand (or pressure driven demand) models (PDDM) have been proposed and verified on a number of benchmark networks, considering both normal and pressure deficient scenarios. Some methods (Bhave 1991; Gupta and Bhave 1996; Kalungi and Tanyimboh 2003; Ang and Jowitt 2006) used the DDM repetitively until a satisfactory hydraulic consistency was obtained, thus requiring intensive computation. This may limit its applicability to large networks. In contrast, other methods (Cheung et al. 2005; Rossman 2007; Giustolisi et al. 2008; Morley and Tricarico 2008; Tanyimboh and Templeman 2010; Siew and Tanyimboh 2012) incorporated the PDDM by introducing a head-flow relationship in the hydraulic equations. A comprehensive review of the PDDM can be found in Wu et al. (2009) and Siew and Tanyimboh (2012).

The DDM can exaggerate the pressure shortage and is unable to cope with different pressure-deficient scenarios commonly found in a real network system. Using the PDDM for the optimal design or rehabilitation of a WDS under various scenarios can provide more useful information to modellers and decision makers than the traditional DDM. Thus, a solution found following this approach is likely to significantly improve the reliability and robustness of the system. Nevertheless, it should be noted that the work undertaken in this thesis only considers the normal condition within a WDS, that is, water consumption at demand nodes can be fully met at the adequate pressure. Hence, only the conventional hydraulic model, namely a demand driven EPANET 2 solver, is used throughout the thesis.

Model Uncertainties

WDS simulation models cannot perfectly describe the behaviour of a real-life network, especially for large and complex systems. Even as a practical tool to analyse WDSs, there are numerous factors that introduce uncertainties to the simulation models. Kapelan et al. (2005) provided a summary of various factors that lead to uncertainties in WDS modelling. However, only uncertainties in water consumption and pipe roughness coefficient were considered in their work. In fact, these types of uncertainties are intrinsic to a WDS, mainly caused by the fluctuations of water use and the deterioration of pipe components.

The stochastic optimal design of WDSs (Lansey et al. 1989; Xu and Goulter 1999; Tolson et al. 2004) can be quite computationally demanding. Therefore, probability-based approaches have been frequently employed to cope with the uncertainties associated with WDSs, in which unknown variables (e.g., nodal demands and roughness coefficients) are sampled from a pre-specified probability density function in problem formulation. Babayan et al. (2005) presented such a method by linking an efficient Genetic Algorithm (GA) to an integration-based uncertainty quantification method, considering demand uncertainty. Kapelan et al. (2005) and Giustolisi et al. (2009) considered uncertainties in both nodal demand and pipe roughness, and formulated the robust WDS design as a MOO. These works also focused on improving the computational efficiency (reduced time) when locating the robust solutions on the PF.

Since it is not one of the objectives of this thesis, all the benchmark models (see Chapter 4) used in this thesis are considered deterministically. Nevertheless, it is worth noting that omitting uncertainty in the design phase may produce an under-design network which can be problematic from a long-term perspective (Babayan et al. 2005).

2.3. Hybrid Optimisation Algorithms

There exist many ways of creating hybrid optimisation algorithms. However, those based on metaheuristics (e.g., EAs) are primarily considered in the following section as they generally make few assumptions about the domain-specific knowledge and thus are often used in a wide range of applications. In addition, since MOO is the major concern of the thesis, hybrid algorithms that are proposed for single-objective optimisation are not considered specifically.

2.3.1. Introduction

MOO is aimed at solving a problem involving more than one objective (often conflicting with each other), with or without constraints. In contrast to single-objective optimisation, the outcome of MOO is a set of solutions which are non-dominated to each other, representing the trade-off relationship between different goals. Here, the non-domination means that no solution can outperform the other solutions in all objectives. Such an optimal set of solutions in the decision variable space is called the Pareto-optimal Set (PS), whereas its shape in the corresponding objective space is known as the PF.

Generally speaking, the goal of MOO is to find the PS and PF to a given problem. However, it is not easy to achieve such a goal as MOO is intrinsically multi-modal, which means that an optimiser can be easily trapped in local optima. Therefore, an algorithm which is able to approximate the PF as close as possible and to maintain a wide distribution of solutions along the PF is much preferred by the analysts and decision makers. Among various techniques for dealing with MOO problems, MOEAs have increasingly gained attention due to the success and flexibility in a wide range of applications. Coello (2006) presented a historical review of the field of MOEAs over the past three decades and categorised their development into two distinct generations. The first generation of MOEAs (from 1983 to 1998) was characterised by the simplicity of the algorithms, but there was a lack of methodologies to validate them. In contrast, the second generation of MOEAs (from 1998 to date) emphasised the efficiency aspect (both at the algorithmic level and the data structure level), which was characterised by adopting the mechanism of elitism. In addition, many test problems and performance indicators were proposed during this period to facilitate the validation of the second generation of MOEAs. Hybrid algorithms, which combine different MOEA concepts (mechanisms) into a unified framework, emerged recently in the literature. Jourdan et al. (2009) proposed to extend an existing taxonomy (Talbi 2002) of hybrid algorithms using exact methods and metaheuristics. Four basic types of hybrid algorithms were derived from the taxonomy; that is, low-level relay hybrid, low-level teamwork hybrid, high-level relay hybrid and high-level teamwork hybrid. A review on state-of-the-art hybrid MOEAs is presented in Section 2.3.2.

Currently, it might be too early to say whether hybrid MOEAs can represent the third generation of MOEAs in the community. However, it is worth investigating the mechanism and performance of these hybrid methods on a wide range of real world applications, like the multi-objective design or rehabilitation of WDSs, which is the focus of this thesis. This will facilitate the understanding of hybrid MOEAs from both theoretical and practical perspectives.

2.3.2. State-of-the-Art Multi-Objective Hybrid Algorithms

A summary of several efficient multi-objective hybrid algorithms reported in the literature is presented in the following section. Some are dedicated MOEAs which are tailored for the optimal design or rehabilitation of WDSs; whereas the others are of general purpose type which are suitable for a wide range of problems. Note that not all hybrid algorithms from the literature are included, but only those with useful features for the problem under consideration in this thesis are covered.

ParEGO

ParEGO (Knowles 2005) was proposed as an effective and efficient search tool in the situation where the number of function evaluations is highly restricted due to financial and/or temporal considerations. The method was extended from the single-objective Efficient Global Optimisation algorithm (Jones et al. 1998). ParEGO learns a Gaussian process model of the search landscape, which is frequently updated after each function evaluation. More precisely, a version of the design and analysis of computer experiments (DACE) model (Sacks et al. 1989) was used to approximate the search space, which appeared to work well in cases with low noise and a small number of dimensions. Different objective values were aggregated into a single one using the augmented Tchebycheff function, and at each iteration, a different weight vector was chosen to gradually build an approximation to the PF. A steady state genetic algorithm was embedded within ParEGO to discover the solution that maximises the expected improvement with respect to the DACE model. Since, at each step, ParEGO requires hundreds of matrix inversions based on the solutions found so far, it can be computationally expensive for longer searches. In addition, ParEGO may not be well-suited for the problems involving a large number of decision variables (high dimensionality), which is the case in the design of WDSs.

In Knowles (2005), ParEGO generally outperformed NSGA-II on nine selected benchmark test problems, featuring low-dimensional, non-pathological characteristics. Di Pierro et al. (2006; 2009) applied ParEGO for the first time to the multi-objective design of WDSs, comparing it with another hybrid algorithm, known as LEMMO (Jourdan et al. 2004), and PESA-II (Corne et al. 2001). The results showed that ParEGO could achieve a satisfactory performance with a significant reduction in the number of function evaluations.

LEMMO

Jourdan et al. (2004; 2005a; 2005b) developed a Pareto-dominance based hybrid MOEA by integrating a Learnable Evolution Model (Michalski et al. 2000) into NSGA-II (Deb et al. 2002). Learnable Evolution Model for Multi-Objective optimisation (LEMMO) was applied to solving the optimal design and rehabilitation problems of WDSs. In order to improve the performance of MOEA as well as reduce the NFEs, the whole procedure was comprised of evolution phases and learning phases. In the evolution phases, NSGA-II was implemented as usual and a set of training data was collected. Then, after every 10 generations, the *C4.5* induction algorithm (Quinlan 1993) was implemented to extract rules from the resulting decision tree. The algorithm used the best and the worst 30% of solutions found with respect to a specific objective (randomly swapped in each learning phase) as good and bad sets, respectively. These new criteria (i.e., positive and negative rules from *C4.5*) were applied to generate offspring and repair the ones matching the negative rules.

LEMMO was compared with NSGA-II and PESA-II, which were well-known benchmark MOEAs in the literature. The comparative studies (Jourdan et al. 2005b; Di Pierro et al. 2009) proved that LEMMO was promising in tackling complex network design problems associated with a large number of pipes. It can save substantial computational budget and improves the performance significantly.

AMALGAM

Vrugt and Robinson (2007) proposed a multi-algorithm, genetically adaptive multi-objective method, named AMALGAM, as a high-level teamwork hybrid optimisation framework (Talbi 2002). It employed four sub-algorithms simultaneously, including NSGA-II (Deb et al. 2002), adaptive metropolis search (AMS) (Haario et al. 2001), particle swarm optimisation (PSO) (Kennedy and Eberhart 2001) and differential evolution (DE) (Storn and Price 1997). AMALGAM was designed to overcome the drawbacks of using an individual algorithm, and thus is suitable for a wide range of problems. The strategies of global information sharing and genetically adaptive offspring creation were implemented in the process of population evolution. More specifically, the pool of current best solutions was shared among sub-algorithms for reproduction. The basic idea of adaptive multi-method search was to take full advantage of the most efficient sub-algorithm and to keep a balance in using diverse methods. Each algorithm was allowed to produce a number of children according to the reproductive rate (ratio of the children alive to the children created) in the previous generation. However, if a sub-algorithm failed to contribute even a single individual in the latest population, a minimum number of individuals (5 as the bottom line) were consistently maintained for it to generate the offspring. Therefore, the most successful sub-algorithm (with highest reproductive rate) was favoured by giving more spaces to accommodate its offspring, but none was completely discarded even though it exhibited the worst performance.

In addition, AMALGAM provided a general template which is flexible and extensible, and can easily accommodate any other population-based algorithm. Raad et al. (2009; 2011) subsequently demonstrated that this hybrid framework, with other ingredients specially tailored for WDS design or rehabilitation, convincingly outperformed NSGA-II for a large real-world problem. Nevertheless, Wang et al. (2014b) argued that AMALGAM may suffer a severe deterioration in performance due to the loss of adaptive capabilities when facing complex design problems.

MOHO

In contrast to AMALGAM, Moral and Dulikravich (2008) presented another Pareto-dominance based multi-objective hybrid optimisation algorithm (MOHO) as a high-level relay hybrid metaheuristic, which implemented three sub-algorithms in a sequential manner. MOHO coordinates the Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al. 2002), the Multi-Objective Particle Swarm Optimisation (MOPSO) (Eberhardt et al. 2001) and the Non-Dominated Sorting Differential Evolution (NSDE, a low-level hybrid metaheuristic combining NSGA-II and DE), and decides which one of them is to generate offspring using the automatic switching procedure. More specifically, MOHO proceeds by choosing one of three sub-algorithms for producing the next generation based on the performance of the currently employed algorithm. Five different indicators for measuring improvements on finding non-dominated solutions, including the quality of approximation and distribution, were used to determine whether to continue with a particular sub-algorithm or change to another one.

MOHO evaluated the performance of its sub-algorithms on five distinct improvements: (1) changes in the size of non-dominated set; (2) whether the new solution could dominate any member in the last generation; (3) changes in the hypervolume indicator; (4) changes in average Euclidian distance; (5) increase in the spread indicator. The innovative part of this evaluation strategy was that MOHO considers not only the quality of the non-dominated set in the next generation (i.e., in terms of convergence and diversity), but also takes into account the perturbation introduced by the potential solutions, which might bring substantial improvement in later iterations.

Wang et al. (2014b) showed that MOHO was less efficient than AMALGAM for a diverse range of benchmark design or rehabilitation problems. Therefore, it is not considered in the comparative studies in Chapters 5 and 6.

Borg

Using ε-MOEA (Deb et al. 2005) as its predecessors, Borg (Hadka and Reed 2013) incorporated more advanced features into a unified framework, including ε-dominance, ε-progress (a measure of convergence speed), randomised restart, and auto-adaptive multi-operator recombination (similar to AMALGAM). The comparative study (Hadka and Reed 2012) on thirty-three instances of three well-known test suites revealed that it was efficient and reliable on various problems with difficult characteristics. Besides its flexibility, Borg showed a large region of so-called “sweet spots” (Purshouse and Fleming 2007) in the parameterisation space, indicating a robust, high-performing method.

The advantages of Borg are threefold: (1) usage of ε-box dominance archive contributes to maintaining the convergence and diversity concurrently throughout search; (2) the combination of time continuation (Srivastava 2002), adaptive population sizing, and two types of randomised restart (i.e. ε-progress triggered restart and population-to-archive ratio triggered restart) boosts the algorithm towards PF; (3) simultaneous employment of multiple recombination operators enhances performance on a wide assortment of problem domains. In addition, using the steady-state, elitist model of ε-MOEA (Deb et al. 2005) makes it easily extendable for use on parallel architectures. Borg has also been successfully used to solve various challenging, many-objective, real-world problems in the domain of water resources (e.g., rainfall-runoff calibration, long-term groundwater monitoring and risk-based water supply portfolio planning). For greater details, the reader is referred to Reed et al. (2013).

2.3.3. Other Issues of Hybrid Optimisation Algorithms

Parameterisation and Robustness

EAs are frequently criticised for their heavy computational overhead and parameterisation issue (Kollat and Reed 2006). It is a common practice to allow a great number of function evaluations for optimisation; however, there is no guarantee that the best PF can be found. On the other hand, it is well established that the individual parameters can greatly impact on the performance of EAs. Although there are some recommended settings (rules of thumb), the problem-dependent parameters of EAs usually require a fine-tuning process, which is time-consuming especially for large and/or complex problems. It is also risky to adopt the current best combination of parameters for another application. The issue can be even more challenging for hybrid algorithms, since more parameters are probably involved.

Consequently, some efforts have been made to develop the adaptive or self-adaptive EAs in order to alleviate this issue to some extent. Abbass (2002) presented a self-adaptive DE in which the crossover and mutation rates were altered in the same way as yielding offspring from parents in each generation. Zheng et al. (2013b) encoded the crossover probability and differential weight into the chromosome and updated their values at the individual level during evolution. These approaches can be expected to reduce the effort required to fine-tune an algorithm, thus being robust for a range of applications.

Constraint Handling

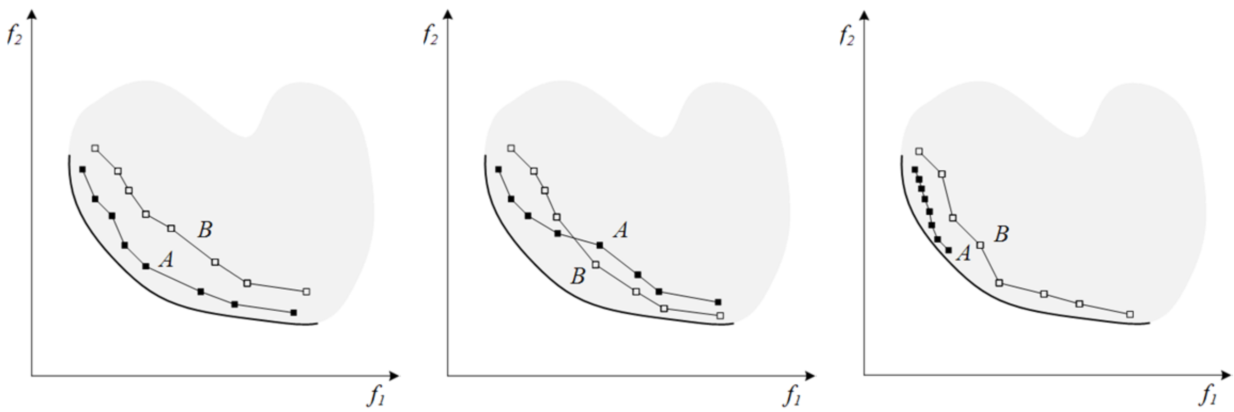
Most real-world optimisation problems contain constraints, including side constraints and/or functional constraints (Hassan et al. 2005). Here, the side constraints refer to the allowed range of decision variables, whereas the functional constraints involve other considerations related to the specific requirements of the problem formulation. The existence of these constraints usually adds another level of difficulty to the problem, since an algorithm must be able to move out of the infeasible region to identify the optima. Various methods have been developed to handle constraints, from the widely used penalty-function to the more advanced strategies, like the Inverse Parabolic Spread method introduced by Padhye et al. (2013b). Penalty-function based strategies are criticised because they usually require the specification of a scaling factor (weight) to properly discriminate among infeasible solutions. In contrast, one commonly used strategy, which is penalty-free, was proposed by Deb et al. (2002) in NSGA-II. It deals with constraints in the selection process by discriminating against infeasible solution with respect to feasible ones. This turned out to be very effective and efficient for solving constrained problems. However, as stated in Singh et al. (2013), such a strategy forced NSGA-II to approach the global optima through the feasible region only. On the contrary, the strategy they applied, i.e., preserving a small portion of ‘good’ infeasible solutions, drove the algorithm from various directions, thus increasing the chance to reach the global optima.

Other constraint-handling methods also exist. For example, MOEAs themselves can be used to convert a problem from constrained to unconstrained by treating the constraint violation as an additional objective. However, this method inevitably increases the computational complexity of the original problem, thereby suffering from the ‘curse of dimensionality’.

2.4. Performance Assessment

It is important to verify the capacity of a newly-developed algorithm against existing ones (usually the most popular algorithms) through comparative studies. Nevertheless, it is always non-trivial to consider how the performance should be evaluated. Note that there are mainly two distinct goals in MOO: (1) to approximate the PF as close as possible (convergence); and (2) to maintain a good spread of solutions in the objective space (diversity) (Deb et al. 2002). Ideally, a good algorithm is expected to discover the solutions lying on the PF and well distributed over the entire objective space (spread). However, this is not always possible when dealing with difficult problems.

In most general cases, the outcome of MOO is a set of solutions, known as non-dominated solutions, rather than a single solution (scalar) as in single-objective optimisation. The complexity arises due to the fact that the relationship between the data sets obtained by different algorithms may not be easy to interpret. For instance, Figure 2.1 shows several scenarios which one may encounter during MOO (assuming that both objectives are to be minimised). In case (a), one can easily tell that set *A* is better than set *B* in terms of closeness to the PF (shown in a thicker solid line). However, in case (b), set *A* and *B* are incomparable as part of solutions in set *A* are dominated by those in set *B*, and vice versa. Case (c) demonstrates a more complicated situation, in which set *A* is better than set *B* in terms of convergence, while set *B* covers a wider range of objective space compared to set *A*. In these cases (b and c), the comparison of non-dominated solutions obtained by different algorithms is difficult.



(a) *A* converges better than *B*. (b) *A* and *B* are incomparable. (c) *A* and *B* are incomparable.

Figure 2.1 Several types of relationship between two sets of non-dominated solutions

Roughly speaking, there are two types of approach which are commonly employed in comparative studies: (1) quantitative performance indicators; and (2) graphical approaches. They are complementary to each other in some sense. A review of both approaches is provided in the following sections.

2.4.1. Quantitative Performance Indicators

As MOEAs have been successively applied to a wide range of applications, there is a clear trend that various methods are compared on benchmark problems in a rigorous and systematic manner (Zitzler et al. 2000; Deb et al. 2002; Farmani et al. 2005a; Bradstreet et al. 2007; Raad et al. 2011; Reed et al. 2013). Quantitative performance indicators are widely adopted in this case to measure explicitly the quality of solutions found by algorithms. So far, a number of metrics have been proposed to map the quality of non-dominated solutions to a scalar value. Some are designed for evaluating a unique characteristic of non-dominated solutions, i.e., convergence or diversity, whereas other metrics are used for measuring the quality of solutions in a combined sense. Table 2.2 summarises commonly used unary metrics (i.e., the indicator assigns each approximation set a scalar value that reflects a certain aspect of the quality) in the literature.

Although unary quality indicators (and their combinations) are capable of capturing specific aspects of algorithms on test problems, many of them do not respect the dominance relations between approximation sets (Zitzler et al. 2003). For instance, despite a set *A* may be evaluated to be better than set *B* according to the unary indicator(s); *B* can be superior to *A* with respect to the dominance relations. On the contrary, binary indicators, e.g., the binary ε - indicator and binary coverage indicator (Zitzler et al. 2003), eliminate such theoretical limitations of unary indicators in principle.

2.4.2. Graphical Approaches

When comparing the performance of different algorithms in the context of MOO, it is an intuitive way to illustrate the non-dominated solutions obtained in the objective space (mainly on a two-dimensional graph). Thus, the Pareto fronts obtained via various algorithms can be directly compared by plotting them on the same graph (or side by side). Due to the stochastic nature of MOEAs, multiple runs are usually conducted and the aggregated Pareto front, representing the best solutions ever found, is then to be compared with those reported by other algorithms.

Table 2.2 Commonly used unary quantitative performance indicators

Type	Metric	Range	Ideal Value
Convergence	Error Ratio (Veldhuizen 1999)	[0, 1]	0
	Set Coverage (Zitzler 1999)	[0, 1]	0
	Generational Distance (Veldhuizen 1999)	$[0, \infty)$	0
	Gamma Metric (Deb et al. 2002)	$[0, \infty)$	0
	Runtime Convergence (Deb and Jain 2002)	[0, 1]	0
Diversity	Chi-Square-Like Deviation (Srinivas and Deb 1994)	$[0, \infty)$	0
	Spacing (Schott 1995)	$[0, \infty)$	0
	Delta Metric (Deb et al. 2002)	$[0, \infty)$	0
	Runtime Diversity (Deb and Jain 2002)	[0, 1]	1
Both Aspects	Hypervolume (Zitzler 1999)	$(0, \infty)$	∞
	Unary ε -indicator (Zitzler et al. 2003)	$[0, \infty)$	0
	ε -performance (Kollat and Reed 2005)	[0, 1]	1

However, this can be difficult to present and results in biased interpretations in practice. Moreover, the question of where and by how much (in the objective space) one algorithm outperforms another one cannot be answered adequately. To bridge the gap, López-Ibáñez et al. (2010) proposed a graphical tool, called the empirical attainment function (EAF[1]), to estimate the attainment function (Fonseca et al. 2001) using the data collected from several independent runs. The method is able to demonstrate the boundaries of attainment surface detected by an algorithm through multiple runs and highlight the performance differences between two competing algorithms. This is particularly useful for comparative study and thus is extensively employed in this thesis. In addition to that, only the first-order EAF is considered which gives the probabilistic performance (typically showing the best, the median and the worst attainment surfaces) of obtaining a solution vector in the objective space through multiple runs.

Another good example of graphical approach for comparative study was found in Hadka and Reed (2012; 2013), in which a grey-shaded plot was incorporated into a diagnostic framework to illustrate the overall performance of various MOEAs across thirty-three benchmark problems. However, this is due to the difficulty of showing attainment surface in case of many-objective (four or more) optimisation. Since the two-objective optimisation is considered in this thesis, using the EAF tool, rather than the grey-shaded plot, can provide a direct comparison of the performances of various MOEAs in the objective space.

2.4.3. Advantages and Disadvantages

It is of great importance to carefully select performance indicators as the outcome of the comparison largely depends on the exact definition of the indicators. Therefore, Deb (2001) suggested using at least two metrics when comparing the performance of two or more algorithms. Despite being able to assess the performance in a quantitative way, the aforementioned indicators present some drawbacks.

Knowles and Corne (2002) critically compared a variety of published metrics following the framework of 'outperformance relations' (Hansen and Jaszkiewicz

---

[1] The package of EAF graphical tool is implemented in *R* language and can be accessed via the following hyperlink http://iridia.ulb.ac.be/~manuel/eaftools.

1998). Also, these metrics were analysed according to other qualities, such as dependency on a reference set and computational overhead. They recommended using three *R* metrics (Hansen and Jaszkiewicz 1998) and hypervolume metric (Zitzler 1999). The *R* metrics, which require choosing a set of utility functions and numerical integration, may be difficult to apply in practice. In contrast, the latter possesses more desirable features (e.g., intuitive, scale independent and compatible with the outperformance relations) and is suitable for moderate-sized non-dominated sets in low-dimensional cases.

Later on, Zitzler et al. (2003) developed a mathematical framework for quantitative comparison of the performance of different algorithms in MOO. They proposed a rigorous analysis solely based on the relations of dominance of two approximation sets, i.e., strictly dominates, dominates, better, weakly dominates, and incomparable. They argued that the comparison methods should be at least capable of detecting whether one set is better than another. This means that it should be compatible and complete with respect to as many of the dominance relations (as aforementioned) as possible. Therefore, a finite combination of unary quality indicators, for instance one for convergence and another for diversity, were not able to achieve this goal. Therefore, several binary quality indicators were proposed as they showed compatibility and completeness with most of the dominance relations. However, the number of values to be considered for binary indicators was doubled.

2.4.4. Other Issues

Ultimate Performance vs. Dynamic Performance

Most comparative studies of MOO used several performance indicators, and focused purely on what was achieved at the end of optimisation (i.e., ultimate performance). Therefore, these works may miss some interesting (potentially important) information about the performance of algorithms compared during the course (i.e., dynamic performance), which include how and when a method discovered the final solutions. Therefore, a generation-wise measurement of the performance of an algorithm (as demonstrated in most single-objective optimisation) can provide more insights into its characteristics as well as the features of problems solved. Zitzler et al. (2002) emphasised the importance of tracing the performance of an algorithm over time. This would help to identify the differences on convergence speed and premature convergence (or stagnation). Deb and Jain (2002) discussed the desirable properties of running metrics and proposed two indicators for measuring dynamic convergence and diversity. As some indicators mentioned in Section 2.4.1 have become more and more popular, they have been frequently used to evaluate the dynamic performance in the comparative studies. Such work can be found in Kollat and Reed (2006) and Fu et al. (2012a).

Computational Complexity

The computational complexity of quality indicators is worth taking into account when evaluating the performance of an algorithm. For example, the hypervolume indicator, which measures the volume of objective space dominated by a non-dominated set, has a runtime complexity of $O(N^{M-1})$, where N is the cardinality of the non-dominated set and M is the number of objectives. Although some quicker implementations (Beume and Rudolph 2006; While et al. 2011) exist, it should be borne in mind that the calculation of hypervolume can be time-consuming, especially when used for dynamic performance assessment on high-dimensional problems. Some computational complexity analyses about the indicators mentioned in Section 2.4.1 were given in (Knowles and Corne 2002).

In the context of multi-objective optimal design of WDSs, the PF is multi-modal, discrete and not uniformly distributed. Moreover, it is usually unknown a priori (except for some extremely simple networks). By acknowledging the challenge of performance assessment of MOO, in this thesis both quality indicators and a graphical approach which are complementary to each other are employed to provide a better interpretation of the solutions finally obtained. In particular, four performance indicators (i.e., generational distance, unary hypervolume, unary additive ε-indicator and ε-performance) as well as the EAF tool are jointly used to compare both the ultimate and dynamic performance of various algorithms considered. The reasons for doing so are threefold: (1) they represent the most recommended indicators in many comparative studies (Knowles and Corne 2002; Zitzler et al. 2003; Fonseca et al. 2006; Hadka and Reed 2012); (2) the ε-performance indicator (Kollat and Reed 2005) is added to cope with the MOEAs using the concept of ε-dominance (Laumanns et al. 2002); (3) main functional objectives of MOO, i.e., convergence, diversity and consistency, can be appropriately evaluated by using a combination of these methods.

On the other hand, since all these adopted metrics assume the existence of a reference set (PF or a good approximation to the PF), the best-known PF of each benchmark problems (see Chapter 4) of WDSs is taken as the reference set. These best-known PFs (including the PFs of three simple cases) were obtained by means of running several state-of-the-art MOEAs extensively and filtering the best solutions obtained through multiple runs.

2.5. Summary

In this chapter, a literature review of multi-objective design and rehabilitation of WDSs and hybrid optimisation methods is presented.

In Section 2.2, a review of the multi-objective optimal design or rehabilitation of WDSs is first given. It includes problem formulations and methodologies available. In particular, MOEAs were the main focus as they demonstrate many advantages over classical methods. Other relevant issues (i.e., hydraulic simulation models and model uncertainties) involved in optimal design or rehabilitation of WDSs are also briefly discussed.

In Section 2.3, a review of hybrid optimisation algorithms, which is the central topic of this thesis, is presented. Several state-of-the-art hybrid methods are introduced chronologically, followed by a description of the issues about parameterisation and constraint handling techniques.

In Section 2.4, various performance assessment methods, including numerical indicators and graphical approaches are discussed as they play an important role in the comparative study. The performance assessment method adopted in this thesis is also specified in the end.

3. GALAXY-A NEW HYBRID MOEA FOR MULTI-OBJECTIVE DESIGN OF WDS

3.1. Introduction

In this chapter, an innovative hybrid multi-objective evolutionary algorithm is proposed, which is inspired by state-of-the-art hybrid optimisation methods. This new hybrid algorithm is termed Genetically Adaptive Leaping Algorithm for approXimation and diversitY (GALAXY), and it incorporates up to six search operators into a unified framework, taking advantage of their synergistic effect to improve the effectiveness and efficiency of searches.

In Section 3.2, some fundamental questions at the development stage are first raised and discussed. Then, a general framework of the GALAXY approach is illustrated followed by the detailed description of each component. Finally, several important strategies employed inside GALAXY are explained.

To address the parameterisation issue commonly faced by MOEAs, Section 3.3 carries out a sensitivity analysis to verify the robustness of the dither creeping operator, which is the only part of GALAXY that requires parameter fine-tuning. Thus, the number of user-specified parameters is significantly reduced, which can benefit researchers and practitioners alike in dealing with other applications.

In Section 3.4, the entire chapter is summarised.

3.2. Design of GALAXY

In this section, the development of the GALAXY method is explained in more detail, including some essential questions raised at the design stage and the corresponding methodologies which form the primary components of the proposed hybrid framework. Then, several important strategies are used to facilitate search in a more effective and efficient way. Note that GALAXY is not just an optimisation method but a framework which can be extended to create brand-new hybrid algorithms.

3.2.1. Design Consideration

Before formally introducing the GALAXY method, it is worth analysing some fundamental questions related to the multi-objective design or rehabilitation of WDSs. Two types of questions are considered here, that is, (1) why the WDS design problem is difficult to solve; and (2) what aspects of hybrid algorithms should be well addressed.

Firstly, a natural and typical way for solving a problem is to understand what the problem is and why it is difficult to solve. The constrained MOO formulation of WDS design is a computationally complex, NP-hard problem (Papadimitriou and Steiglitz 1998), of which the search space is enormous, discontinuous, non-convex and multi-modal. In addition, the highly constrained nature of the problem, due to the service standards and operational requirements of a network adds another level of difficulty. A number of sources of difficulties of multi-objective design or rehabilitation of WDSs are summarised as follows:

- The size of the network under consideration, i.e., the number of pipes (decision variables) to be optimised;
- The number of commercially available pipe sizes (a range of discrete diameter options);
- The type of hydraulic simulation involved: i.e., static (snapshot) or extended period simulation;
- The number of loading conditions considered for the design purpose (in case of multiple loading conditions, one objective function evaluation requires several hydraulic simulations);
- Other possible reasons, e.g., complex components (valves or control rules) existing in a network can significantly increase the non-linearity and complexity of the hydraulic simulation.

These above mentioned issues are commonly encountered by a practitioner using an optimiser. Note that in this thesis all the benchmark networks, except the ’Anytown’ network (Walski et al. 1987), are gravity-fed systems or those supplied via a single pump. In other words, no complex hydraulic components (e.g., pressure reducing valve) are included in the majority of benchmark problems, because the size of the network system (i.e., the number of pipes to be optimised) is regarded as the major source of difficulty.

Secondly, in order to design a hybrid algorithm, the following questions should be raised beforehand.

- What kind of algorithmic framework should be adopted, i.e., steady-state or generational?
- Which dominance relationship should be employed, i.e., Pareto-dominance or ε-dominance?
- Which coding scheme should be applied, i.e., binary, real or integer coding?
- What criteria are used for selecting search operators?
- What kind of hybridisation should be considered, i.e., parallel or sequential?
- What specific strategies should be implemented to ensure and even enhance the performance of a hybrid algorithm?

These questions must be properly addressed to propose a new and, hopefully, more powerful search tool for the multi-objective design or rehabilitation of WDSs.

Finally, some common issues faced by MOEAs should also be taken into consideration. For instance, the parameterisation issue can greatly influence the behaviour of an MOEA, which may be alleviated to some extent by using adaptive or self-adaptive techniques (Zheng et al. 2013b). However, if fewer parameters are involved, the issue can be further simplified and is probably much easier to deal with. In addition, it should be borne in mind that the constraint handling method can affect the behaviour of MOEAs. Many MOEAs adopt the constrained-domination comparison (Deb et al. 2002) to handle infeasible solutions during optimisation. Others use a more traditional way, namely the penalty-function, to convert the problem to an unconstrained one.

In short, as discussed above, many aspects of multi-objective design or rehabilitation of WDSs via MOEAs require careful thought before proposing a new hybrid algorithm. In the next section, the GALAXY method is developed step by step by addressing the aforementioned concerns.

3.2.2. Framework of GALAXY

The majority of modern MOEAs were created following the Darwinian evolutionary concepts, i.e., survival of the fittest and adaptation to the environment. Padhye et al. (2013a) presented a *Unified Approach* for evolutionary algorithms (e.g., GAs, PSO and DE) in the context of real-parameter optimisation of unimodal problems, called Evolutionary Optimisation System (EOS). The EOS includes four key steps, that is, *Initialisation*, *Selection*, *Generation* and *Replacement*. This framework also holds for MOEAs, although the *Selection* and *Replacement* steps need to adapt to multi-objective perspectives.

Figure 3.1 illustrates a basic workflow of solving a general optimal design problem of WDS via MOEAs. Firstly, an initial population is randomly generated and evaluated by the objective functions, which involve hydraulic simulations. Secondly, the parents are selected from the population and offspring are generated. Then, the offspring are evaluated using the objective functions. If some of them dominate the members of the current population, the corresponding individuals are replaced by these offspring. This procedure is implemented repeatedly until a certain stopping criterion is satisfied.

In theory, it is possible to propose a brand-new hybrid algorithm by altering all the key steps of the EOS. However, the current version of GALAXY implements hybridisation only at the *Selection* and *Generation* steps. The pseudo-code of GALAXY is illustrated in Figure 3.2 and the developments of main components of GALAXY are explained in subsequent sections in more details.

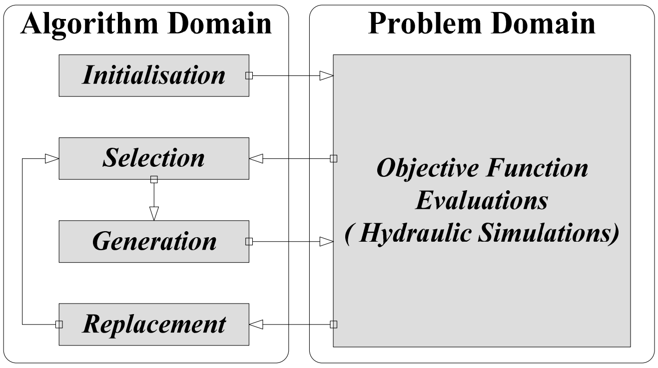


Figure 3.1 Major steps in using MOEAs to solve a WDS design problem (adapted from (Padhye et al. 2013a))

GALAXY Method

Inputs: population size (*N*), number of function evaluations (*NFE*)
Outputs: Pareto approximation set (*AS*), Pareto approximation front (*AF*)

Initialisation:

Generate the initial population of *N* individuals randomly in the specified variable domains.

Initialise the quotas* of six search operators equally such that ($\sum_{J=1}^{6} N_J = N$).

Evaluation:

Evaluate the objective function values of the initial population (involving hydraulic simulations).

Rank the population using the non-dominated sorting procedure (Deb et al. 2002).

Update the current number of function evaluations (i.e., set *I* = *N*).

While *I* <= *NFE*

Selection:

Choose all the members in the current population for the ***Generation*** step.

Generation:

For *J* = 1 to 6

Produce *N* candidate solutions from the current population using operator *J*.

Select N_J offspring randomly from candidate solutions and save them to the offspring set.

End

Check whether the solutions in the offspring set are within the specified variable ranges.

Evaluation:

Evaluate the objective function values of the solutions in the offspring set.

Replacement:

Combine the current population and the offspring set as an intermediate population of size *2N*.

Implement the duplicates handling strategy.
Rank the intermediate population using the non-dominated sorting procedure.
If the number of individuals in the top rank <= *N*
Implement the normal replacement via the crowded-comparison operator (Deb et al. 2002).
Else
Implement the ε-replacement strategy.
End
Form the next population of size *N*.
Update the quotas* of search operators according to their contributions to the next population.
Update the current number of function evaluations (i.e., set $I = I + N$).
End
Set the current population as *AS*.
Set the objective function values of the current population as *AF*.

Note: *A quota of a search operator refers to the number of offspring it is allowed to produce for the next generation.

Figure 3.2 Pseudo-code of the GALAXY method

3.2.3. Component Development

3.2.3.1. Algorithmic Framework

In MOEAs, there are generally two distinct algorithmic structures, namely the generational and the steady-state frameworks. The majority of current MOEAs in the domain are based on the generational framework, in which the offspring are first produced and then compared with their parents. For example, in each generation the NSGA-II (Deb et al. 2002) creates the same number of offspring as the population size, which are then merged with the current population before non-dominated sorting is applied. While in the steady-state framework, a child is compared with the current population immediately after it is created. If it is better than (i.e., dominating) at least one current solution, it is accepted and the population gets updated. This framework makes it possible to find good offspring solutions more efficiently in terms of computational time (Deb et al. 2003).

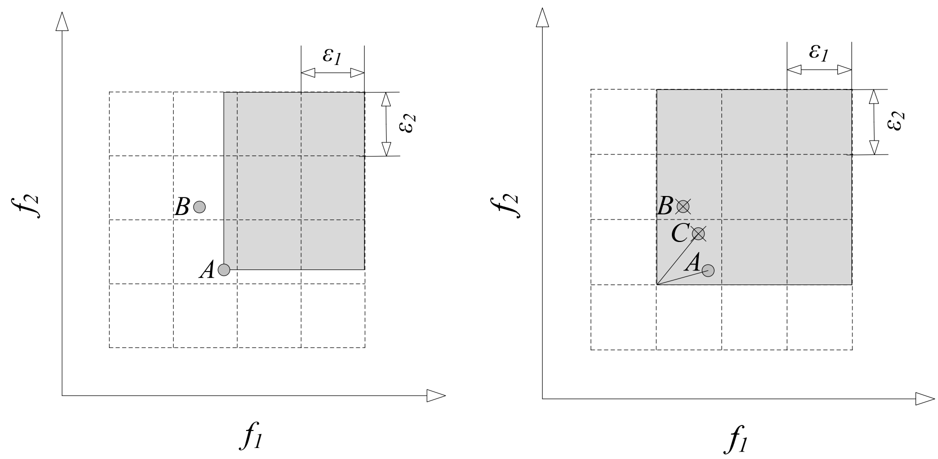
The main consideration towards the algorithmic framework lies in how important the implementation speed is. Since the *Selection* and *Replacement* in the steady-state framework involve only comparisons based on the dominance relationship, computationally expensive sorting and truncation operations are eliminated. This can lead to a significant saving of computational time especially for high-dimensional problems. As shown in Deb et al. (2005), the ε-MOEA evidently outperformed the NSGA-II and other MOEAs in terms of CPU time for various benchmark problems. However, for the two-objective design or rehabilitation of WDSs concerned in this thesis, the marginal gain on efficiency by using the steady-state framework is not significant. The reason is that the bottleneck of computation for this kind of problems lies in hydraulic simulations, rather than in non-dominated sorting and diversity preservation. Therefore, the commonly used generational framework as in the NSGA-II is taken as the algorithmic structure of the GALAXY method.

On the other hand, both algorithmic frameworks involve elitism, which prevents good solutions found so far from being removed from the population subsequently. In the generational structure, the elitism is implicitly achieved by applying the non-dominated sorting operation to a combined set of parents and their offspring. In contrast, the steady-state structure explicitly implements elitism by comparing the offspring with each member in the population and in the external archive.

3.2.3.2. Dominance Concept

Although Goldberg (1989) did not present a Pareto-dominance based evolutionary algorithm in his book '*Genetic Algorithms in Search, Optimization and Machine Learning*', it indeed influenced the creation of the first generation MOEAs (Coello 2006). In the Pareto-dominance concept, a solution is said to dominate another one if it is better than this solution in at least one objective and is no worse than this solution in the other objectives. Otherwise, they are said to be non-dominated with each other. In contrast, the ε-dominance concept (Laumanns et al. 2002) does not regard two solutions with a difference less than ε_i in the *i*-th objective as non-dominated, thereby ensuring a good convergence and diversity simultaneously. Most recently, a more advanced dominance relationship, called the grid-dominance concept (Yang et al. 2013), was proposed for solving many-objective optimisation problems; however, it is not considered in this thesis.

In Figure 3.3, the Pareto-dominance and the ε-dominance concepts are compared supposing that both objectives are to be minimised. The region dominated by a solution (e.g., solution *A*) is depicted in grey. Solution *A* and *B* are non-dominated to each other under the Pareto-dominance concept (see Figure 3.3(a)), as solution *B* is not located in the grey area. So, both of them are likely to be selected in the *Selection* or *Replacement*. On the contrary, the region dominated by solution *A* is enlarged under the ε-dominance concept (see Figure 3.3(b)). Thus, solution *B* is ε-dominated by solution *A* and discarded in the *Selection* or *Replacement*. In addition, solution *C*, which is located in the same ε-box as solution *A*, is also removed because it has a larger Euclidean distance towards the lower-left corner of that ε-box.



(a) Pareto dominance concept (b) ε-dominance concept

Figure 3.3 Comparison of Pareto-dominance concept with ε-dominance concept

It can be seen from the comparison of different dominance concepts that the main differences lie in the trade-off between accuracy and efficiency. Actually, the ε-dominance and others are relaxed forms of the Pareto-dominance, which can ensure good convergence and diversity at the price of losing some accuracy. This trade-off must be considered for solving high-dimensional problems, as the Pareto-dominance suffers from the curse of dimensionality with many solutions becoming non-dominated, which in turn decreases the selection pressure and lead to deterioration.

For solving low-dimensional problems in terms of the number of objectives, like the two-objective optimal design of WDSs considered in this thesis, it is not necessary to incorporate advanced dominance concepts (e.g., ε-dominance). On the other hand, adopting the ε-dominance concept for solving discrete combinatorial optimisation problems may lead to 'genetic drift', as the solutions containing useful gene information, but with the objective values less than ε_i in the *i*-th objective, are discarded. Therefore, the approximation front obtained by an ε-dominance concept based approach is likely to be concentrated in a particular region of objective space. In other words, boundary solutions may be lost by using this concept. As it will be shown in Chapter 5, both Borg and ε-MOEA have difficulty in discovering solutions in the region of low and high network resilience. In addition, the ε-dominance concept introduces at least two more parameters (i.e., ε value for each objective) to an end user, although it also brings some flexibility in controlling the precision of the obtained solutions. It is not clear whether such flexibility is beneficial to the decision makers, as the proper setting of ε is usually difficult a priori, but the values can significantly affect the shape of the Pareto approximation set obtained.

3.2.3.3. Variable Coding Scheme

In the multi-objective design or rehabilitation of WDSs, the goal is to find a set of non-dominated solutions, of which the pipe diameter options are chosen from a list of commercially available sizes. Therefore, this kind of problem involves decision variables which are discrete in nature, or it is formally known as combinatorial optimisation. Many coding schemes are available for representing integer values, including the binary coding and the Gray coding (Simpson and Goldberg 1994). The binary coding expresses a gene using a series of bits, whose length depends on the range of a decision variable. For example, if 14 sizes are available in the market, using the binary coding scheme requires at least 4 bits to encode a gene (pipe diameter). Then, a chromosome is comprised of a group of genes by a concatenation of strings (0 or 1) of 4-bit length.

However, the binary coding has some obvious drawbacks such as the so-called ‘Hamming Cliffs’ (Goldberg 1989), i.e., when a small distance in genotype would lead to a large difference in phenotype. The Gray coding (Simpson and Goldberg 1994) was therefore developed to alleviate this problem to some extent. However, both of them tend to be inefficient when dealing with high-dimensional problems in terms of number of decision variables, which is always the case with WDS design. In contrast, the real coding scheme, which eliminates the encoding and decoding processes from optimisation, expresses a variable in a straightforward way, thereby significantly shortening the length of a chromosome. Real coding has been shown to be superior to the binary coding scheme on many benchmark problems (Deb et al. 2002) and can increase the efficiency and robustness of evolutionary algorithms (Awad and Poser 2005).

On the other hand, using the real coding scheme, which is specially designed for solving continuous problems, requires further adaptation to accommodate for discrete combinatorial optimisation problems. A simple yet meaningful way is to round off the real values to the nearest integer; however, it does not ensure one-to-one mapping from genotype to phenotype. Even if this is acceptable, the real coding scheme can still be problematic as the fractional part of any decision variable value is meaningless in terms of both genotype and phenotype of a solution in the context of optimal design of WDSs. Hence, a scheme which is chosen for handling integer variables is expected to be more efficient. Hereinafter, the integer coding scheme is used which is distinct from both the binary and the real coding schemes.

Figure 3.4 illustrates the integer coding scheme in more detail. For each solution in the population of size *N*, the length of chromosome is equal to the number of decision variables *ND* (i.e., the number of pipes to be optimised). For each gene in the chromosome (e.g., G_2 of *Solution 2*), there are a total of *S* diameter options to choose from. By using the integer coding scheme, there is no redundant gene representation as seen in the binary coding scheme, and there is no waste of precision as suffered in the real coding scheme.

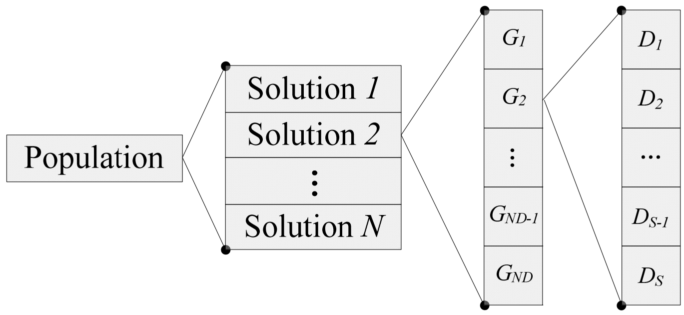


Figure 3.4 Integer coding scheme used for the GALAXY method

3.2.3.4. Search Operators

Typically, an MOEA intrinsically balances two aspects of multi-objective optimisation during the search, which are known as exploration and exploitation. The former is aimed at searching the space globally in order to identify the near-optimal regions, where the global optima are likely to be. In contrast, the latter attempts to conduct a fine-grained search locally in order to further improve the quality of current solutions. As such, an ideal search process (see the schematic shown in Figure 3.5) would first focus the search on the exploration aspect and locate the potential near-optimal areas; then the search would gradually switch from exploration to exploitation to locate the global optima.

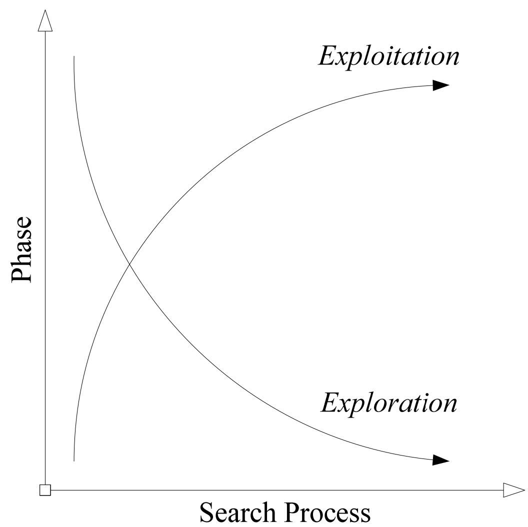


Figure 3.5 An ideal search process using an MOEA

It is of great importance to realise that these two aspects are achieved at the level of search operators, rather than at the level of topmost structure of an MOEA (i.e., *Selection and Replacement*). This fact can be attributed to the nature of each search operator, although it is difficult to accurately control its behaviour at present. In designing a powerful hybrid algorithm, several questions naturally arise, such as: what kinds of operators should be selected and what criteria are used for selecting search operators?

For the first question, a diverse class of operators are preferred because it is likely that an operator cannot be effective and efficient for a wide range of applications, so that a combined use of different operators may perform well on many problems. Hence, an ‘optimised’ portfolio of these operators is expected to facilitate search and yield satisfactory outcome. However, the second question is not trivial and normally requires a trial-and-error approach to determine the criteria for selecting operators.

In AMALGAM, four search operators were involved including GA, PSO, AMS and DE. These operators themselves have been found to be effective and efficient on a wide range of applications. However, when dealing with discrete combinatorial optimisation problems, their capabilities cannot be guaranteed. Quite opposite, as it will be shown in Chapter 5, the overall performance of the AMALGAM deteriorates when the complexity of problems increase, due to the fact that several of these operators are not effective and efficient in the context of WDS design. On the other hand, the AMALGAM involves setting of a total of 11 individual parameters in addition to the population size and the number of function evaluations. A full list of these parameters are summarised in Table 3.1 along with their default settings in the original AMALGAM.

Since AMALGAM was originally proposed for coping with real-valued optimisation problems, it is found that the search operators within AMALGAM were actually not suitable for solving discrete combinatorial problems. However, such a combination of four operators inspires the selection of candidate operators dedicated for solving discrete problems. In other words, these operators need to adapt to the discrete problems and new operators should be introduced in order to improve the search ability. In the subsequent section, the 'leaping ability' of a search operator is first defined, then six search operators employed in the GALAXY method are described in a descending order of leaping ability in the global sense within the solution space.

Table 3.1 Individual parameters of AMALGAM

Search Operators	Parameters	Default Settings
GA (SBX+PM)	SBX rate	0.9
	distribution index for SBX	20
	PM rate	1/*ND*
	distribution index for PM	20
PSO	inertia factor	0.5+0.5*u(0,1)*
	cognitive weight	1.5
	social weight	1.5
	turbulence factor	*u(-1,1)*
AMS	jump rate	$2.4\sqrt{ND}$
DE	scaling factor *K*	*u(0.2,0.6)*
	scaling factor *F*	*u(0.6,1.0)*

Note: *ND* - number of decision variables; *u(a, b)* - a uniformly distributed random number between *a* and *b*; SBX - Simulated Binary Crossover; PM - Polynomial Mutation.

Solving a discrete combinatorial optimisation problem is different from solving a continuous, real-valued problem. In particular, the values of decision variables are restricted to integers, thus the fractional part of decision variables will be omitted during the evaluation of objective functions. MOEAs essentially sample solutions randomly from within the search space to progressively approach the PF. When dealing with a continuous problem, MOEAs can achieve any specified precision, leading the stochastic search to being nearly 'smooth'. In contrast, these algorithms have to explore solutions at a limited interval (at least equal to 1) in the search landscape when facing a discrete optimisation problem. Therefore, their behaviour is not 'smooth' anymore and appears to be 'leaping' in both the objective and the decision variable spaces.

Consequently, when developing a hybrid algorithm, it is of great importance to employ search operators that are good at 'leaping' in the global or local sense.

More specifically, the good ‘leaping’ ability in the global sense refers to the search capability which can explore the space extensively, whereas the good ‘leaping’ ability in the local sense refers to the search capability which is effective and efficient around a small area, aiming at further improving the fitness of solutions.

Figure 3.6 illustrates this ‘leaping’ concept in more detail. The dark solid circles denote the targeted solutions in the reference front; whereas the grey solid circles denote the solutions obtained in the Pareto approximation front. The light grey circles represent the solutions obtained by the operator with good ‘leaping’ ability in the local sense; in contrast, the dark grey circles represent the solutions obtained by the operator with good ‘leaping’ ability in the global sense. The approximation front identified at generation *t* is annotated as AF_t. As it can be seen from the figure below, solutions identified by the operator good at global search (exploration) move rapidly towards the boundary solutions in the reference set; while solutions found by the operator good at local search (exploitation) steadily approach the solutions in the reference set. As such, a combined use of search operators which are good at ‘leaping’ in the global and local sense is anticipated to drive the hybrid algorithm towards the PF quickly and consistently. To this end, up to six search operators are deployed in GALAXY according to their ‘leaping’ ability in the search space.

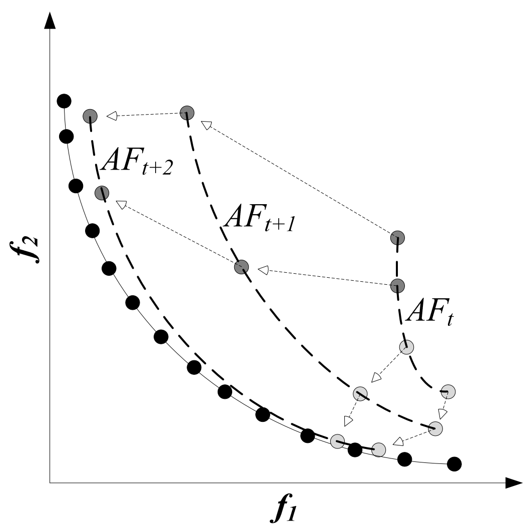


Figure 3.6 Schematic of the ‘leaping’ ability of search operators

Search Operator 1: Turbulence Factor (TF)

The PSO is a population-based stochastic optimisation algorithm, which mimics the movement of organisms in a bird flock or fish school. The initial position and velocity of the population, also known as particles, are randomly generated. Afterwards, the velocity and position of each particle is frequently updated according to its own flying experience (p_{best}) and that of all other particles (g_{best}). Such updating strategies in PSO are given in Eq. (3.1) (velocity) and Eq. (3.2) (position). The definition of p_{best} and g_{best} in the context of multi-objective optimisation are not as straightforward as in single-objective optimisation. In the AMALGAM, they were judged based on the Euclidean distance of particles' objective function values to the best values found so far. In addition, a turbulence factor (denoted as R_T) was subsequently used to perturb the particles' positions in order to avoid the local optima (see Eq. (3.3)).

$$V_{t+1}^{i} = \omega \cdot V_{t}^{i} + c_1 \cdot r_1 \cdot (p_{best} - X_{t}^{i}) + c_2 \cdot r_2 \cdot (g_{best} - X_{t}^{i}) \tag{3.1}$$

$$X_{t+1}^{i} = X_{t}^{i} + V_{t+1}^{i} \tag{3.2}$$

$$X_{t+1}^{i} = X_{t}^{i} + R_T \cdot X_{t}^{i} \tag{3.3}$$

Where, V_t^i and X_t^i - current velocity and position of particle *i*, respectively; ω - inertia factor; c_1 and c_2 - weights of cognitive and social factors of each particle; r_1 and r_2 - uniformly distributed random numbers between (0, 1); R_T - a uniformly distributed random number sampled between [-1, 1].

In trial runs, PSO was observed to make significant contributions only at the beginning of search. In other words, it was completely ineffective after the first few generations (Wang et al. 2014b). This implies that the PSO wasted the opportunities which could be taken by other operators to find better solutions. Such a poor performance might be attributed to the definition of p_{best} and g_{best} , which made it easier for the algorithm to get trapped in local optima. Interestingly, for solving discrete problems, the turbulence factor alone turned out to be very efficient, which enabled a superfast global search at the initial stage of optimisation. Therefore, only this kind of perturbation is maintained in GALAXY, thus eliminating all the individual parameters in the original PSO.

Search Operator 2: Differential Evolution (DE)

The DE is another popular stochastic algorithm because of its simplicity. The method was proposed for real-valued problems and it works by iteratively improving a candidate solution by using weighted differences between other randomly sampled solution vectors. Eq. (3.4) shows such a strategy used by the DE within AMALGAM. The DE has been proved elsewhere to exhibit rotationally invariant feature (Storn and Price 1997), which means that it can cope with the strong interdependencies between decision variables. This is a favourable characteristic for the optimal design of WDSs, since the pipe sizes in a network are correlated, rather than being independent from each other.

$$X_{t+1}^{i} = X_{t}^{i} + K \cdot (X_{t}^{a} - X_{t}^{i}) + F \cdot (X_{t}^{b} - X_{t}^{c}) \tag{3.4}$$

Where, X_t^i - current position of individual *i*; X_t^a, X_t^b and X_t^c - three randomly selected individuals from current population, and they must be different from each other and from X_t^i; *K* and *F* - uniformly generated numbers between (0.2, 0.6) and (0.6, 1.0), respectively. These ranges encompass the recommended values (i.e., *K*=0.4 and *F*=0.8) reported by Iorio and Li (2005).

In GALAXY, to adapt real-valued DE in AMALGAM to discrete problems, the basic form of DE (see Eq. (3.4)) is maintained but the weights *K* and *F* are set to 1 to suit the integer coding scheme.

Search Operator 3 & 4: Simulated Binary Crossover for Integers (SBXI) & Uniform Mutation (UM)

The GA usually consists of two parts, which are known as crossover and mutation. Currently, it remains unclear which one is the dominant driving force for evolutionary optimisation. The Schema Theorem (Holland 1975) provided a theoretical justification that crossover was the primary search mechanism while mutation was regarded as less important, being helpful in preserving the diversity of population. Spears (1993) pointed out that both operators demonstrated different characteristics, i.e., disruption (exploration) and construction (exploitation). Crossover was more effective in constructing high order building blocks from low order ones compared with mutation; whereas mutation was good at disruption, thus providing higher level of exploration. However, Fogel (2006) and Zheng et al. (2010) challenged the Schema Theorem by finding that uniform crossover (more disruptive) outperformed one-point and two-point crossover on many case studies. Others also asserted that mutation is the main operator for driving evolution (Vose 1994; Palmes et al. 2005). It has been shown sufficiently that a GA without mutation tends to converge prematurely.

Similarly, in the NSGA-II used by AMALGAM, it was unknown that whether Simulated Binary Crossover (SBX) or Polynomial Mutation (PM) (Deb et al. 2002) played the major role. Moreover, SBX and PM were developed for solving problems with continuous search space, which does not necessarily mean that they can work well for the discrete search space. Some evidence exists that SBX can achieve similar performance to single-point crossover under binary coding scheme in solving discrete test functions (Deb and Agrawal 1995). Hence, trial runs were conducted to verify their performances for the multi-objective design or rehabilitation of WDSs. It was found that SBX was not particularly useful in generating high quality solutions. Quite opposite, PM alone worked very well for a range of design problems. On the other hand, the spread of children from their parents after applying SBX and PM is determined by an exponential function, whose distribution of density is controlled by an index for SBX (see Eq. (3.5)-(3.7)) and PM (see Eq. (3.8)-(3.10)), respectively. A large value of the index usually leads to a small variation, which focuses the search around the current parents. In contrast, a small index allows farther jumps in the decision space.

$$\begin{cases} X_{t+1}^{a} = 0.5 \times [(X_t^a + X_t^b) - \beta(\alpha^a) \cdot (X_t^b - X_t^a)] \\ X_{t+1}^{b} = 0.5 \times [(X_t^a + X_t^b) + \beta(\alpha^b) \cdot (X_t^b - X_t^a)] \end{cases} \tag{3.5}$$

$$\begin{cases} \beta(\alpha) = (u \cdot \alpha)^{\frac{1}{1+\eta_c}}, u <= 1/\alpha \\ \beta(\alpha) = (\frac{1}{2 - u \cdot \alpha})^{\frac{1}{1+\eta_c}}, u > 1/\alpha \end{cases} \tag{3.6}$$

$$\begin{cases} \alpha^a = 2 - (1 + 2 \times \frac{X_t^a - LB}{X_t^b - X_t^a})^{\frac{1}{1+\eta_c}} \\ \alpha^b = 2 - (1 + 2 \times \frac{UB - X_t^b}{X_t^b - X_t^a})^{\frac{1}{1+\eta_c}} \end{cases} \tag{3.7}$$

Where, X_t^a and X_t^b - two different parents selected at generation *t*; $\beta(\alpha)$ - a vector randomly generated, given a uniformly sampled random number *u* between (0, 1) and a distribution index of η_c for SBX; *UB* and *LB* - vectors of upper and lower bounds of decision variables; α^a and α^b - variables calculated based on the selected parents (X_t^a and X_t^b), *UB*, *LB* and η_c.

$$X_{t+1}^i = X_t^i + \delta \cdot (UB - LB) \tag{3.8}$$

$$\delta = [2u + (1-2u) \cdot (1 - \frac{X_t^i - LB}{UB - LB})^{(1+\eta_m)}]^{\frac{1}{1+\eta_m}} - 1, u <= 0.5 \tag{3.9}$$

$$\delta = 1 - [2(1-u) + 2(u-0.5) \cdot (1 - \frac{UB - X_t^i}{UB - LB})^{(1+\eta_m)}]^{\frac{1}{1+\eta_m}}, u > 0.5 \tag{3.10}$$

Where, X_t^i - current solution at generation *t*; *UB* and *LB* - vectors of upper and lower bounds of decision variables; δ - a vector of small variations which are obtained from a polynomial distribution of a given uniformly sampled random number u between (0, 1); η_m - distribution index of PM.

However, when dealing with discrete problems, a typical variation step must be equal to or larger than 1, which implies that a fine-grained fractional search would be meaningless in both decision and objective spaces. Therefore, to adapt SBX and PM to discrete problems, in this thesis, crossover is separated from mutation and both operators are modified to suit discrete problems. In particular, Eq. (3.6) and (3.7) are omitted deliberately and Eq. (3.5) is replaced with the simplified variations shown in Eq. (3.11). This modified crossover, which is named Simulated Binary Crossover for Integers (SBXI), is found to be compatible with the integer coding scheme in trial runs. Note that the probability of SBXI is fixed at 1.0.

$$\begin{cases} X_{t+1}^{a} = X_{t}^{a} + randi[0,(X_{t}^{b} - X_{t}^{a})], u <= 0.5 \\ X_{t+1}^{b} = X_{t}^{b} - randi[0,(X_{t}^{b} - X_{t}^{a})], u > 0.5 \end{cases} \tag{3.11}$$

Where, $randi$ - a uniformly distributed random integer between [*a*, *b*]; u - uniformly sampled random number between (0, 1).

On the other hand, PM is replaced with Uniform Mutation (UM) to sample an integer value from within the variable domain. The mutation rate is equal to the inverse of number of decision variables (1/*ND*), which is the most recommended value for mutation (Goldberg 1989). It is worth mentioning that such an adaptation of SBX and PM also removes all the individual parameters (i.e., probability of SBX and PM and their distribution indices).

<u>Search Operator 5: Gaussian Mutation (GM)</u>

The AMS operator (Haario et al. 2001) was introduced to the AMALGAM with the attempt to prevent the ‘genetic drift’ suffered by many population-based algorithms, in which most individuals tend to converge towards a single solution. AMS is a sample method based on the Markov Chain Monte Carlo methodology that allows the offspring with lower fitness to replace their parents. Another advantage of AMS is the capability to sample effectively in high-dimensional distributions. The offspring are sampled from a normal distribution with the current best non-dominated solutions and their covariance as mean and standard deviation, respectively (see Eq. (3.12)):

$$X_{t+1}^{i} = N(X_{t}^{i}, c_{n}^{2} \times \mathrm{cov}(X_{t}^{i}, X_{t}^{i})) \tag{3.12}$$

Where, X_{t}^{i} and $\mathrm{cov}(X_{t}^{i}, X_{t}^{i})$ - position and covariance of the best non-dominated solutions at generation *t*; c_n - jump rate which determines the spread of solutions around X_{t}^{i}.

As shown by Wang et al. (2014b), the AMS operator worked very well for small to medium sized WDS design problems. However, for larger and more complex design problems, it was not able to generate solutions of high quality as expected. To maintain the consistency of search operators within the GALAXY framework, this operator is excluded. Furthermore, as the mutation operator alone showed satisfactory performance in trial runs, another mutation operator, called Gaussian Mutation (GM) is therefore introduced into the GALAXY framework. It differs from the UM operator in the distribution of randomly sampled values in the domain of decision variables. In particular, a discrete Gaussian distribution (also known as normal distribution) is adopted as shown in Eq. (3.13). The probability of mutation for each decision variable is equal to the UM operator, namely 1/*ND*. The reason why σ is set to a tenth of the range of diameter options is that it prevents a majority of random numbers generated from falling outside the available pipe sizes.

$$X_{t+1} = \left\lfloor \frac{D^{\min} + D^{\max}}{2} + \sigma \times randn \otimes X_t \right\rfloor \tag{3.13}$$

Where, $D^{\min}$ and $D^{\max}$ - minimum and maximum diameter options for each pipe; σ - a scaling factor which is equal to $\frac{\left(D^{\max} - D^{\min} + 1\right)}{10}$; $randn$ - a vector of random numbers drawn from the normal distribution between (0, 1) with the same size of X_t; $\otimes$ - a vector based product; $\lfloor * \rfloor$ - an operator that rounds an element toward negative infinity.

<u>*Search Operator 6: Dither Creeping (DC)*</u>

To facilitate an intensified local search, a new operator, known as the dither creeping mutation (DC) (Zheng et al. 2013c), is introduced in the GALAXY framework, which combines the creeping mutation (Dandy et al. 1996) and dither mutation strategy (Das et al. 2005) to solve discrete pipe sizing problem. The original DC operator is characterised by a variant probability of mutation (denoted as P_{dcm}) for each individual, which is uniformly sampled from within a small range centred about 1/*ND*. Moreover, the pipe size is changed to the nearest smaller or larger diameter option, depending on the probability of downward variation (denoted as P_d). Zheng et al. (2013c) revealed that a GA with DC but without crossover outperformed its counterparts for a range of benchmark design problems.

Note that here the original DC is modified by yielding the probability of dither creeping at the level of gene, rather than at the level of chromosome. This is expected to bring more perturbation to each solution. On the other hand, an interesting feature of DC lies in the fact that the direction of creeping can be simply controlled by the P_d. More specifically, using a smaller P_d steers the population towards the high cost region (also high in network resilience); whereas using a larger P_d pushes the population towards the low cost region. This characteristic can be very useful if a particular region of the Pareto-optimal front (PF) is of greater interest, although this is realised at the price of losing the whole picture of PF. Consequently, the P_d of the modified DC is uniformly sampled between (0, 1) rather than fixed at 0.5 as in the original DC, in which case the size of each pipe has an equal chance to be enlarged or reduced. Figure 3.7 illustrates the flowchart for both versions of DC operators.

It should be highlighted that the search operators play the essential role in MOEAs, although the *Selection* and/or *Replacement* mainly steer the population towards the PF. After all, the capability of these operators determines whether the search space can be sampled in an effective and efficient manner. The reasons for selecting TF, DE, SBXI, UM, GM and DC as search operators for the GALAXY method are threefold. Firstly, their natures and behaviours are distinct from each other, which make it possible for each one to serve as the main driving force at different stages of search. Secondly, DE generates offspring by combining the information passed by other solutions; while the others work on an individual solution independently and perturb the solution at the level of gene. Therefore, the difference of operators in producing solutions may prevent GALAXY from getting trapped at local optima. Thirdly, from the viewpoint of leaping ability, TF, DE and SBXI are more likely to identify better solutions globally at early generations. In contrast, UM, GM and DC emphasise local variations, being good at fine-tuning current solutions. As a result, a synergistic effect is anticipated by using a combination of six such operators to ensure a good balance of exploration and exploitation throughout the search.

Dither Creeping

P_{dcm}: probability of dither creeping mutation;

$P_{dcm}^{\min}$ and $P_{dcm}^{\max}$: lower and upper bounds of probability of dither creeping mutation;

N and *ND*: population size and number of decision variable;

P_d: conditional probability of downward variation;

X_t^i: decision variable *i* at generation *t*;

$D^{\min}$ and $D^{\max}$:lower and upper bounds of diameter option;

rand: a uniformly distribution random number sampled between (0,1);

Original DC	***Modified DC***
For *i*=1, 2, …, *N*	**For** *i*=1, 2, …, *N*
$P_{dcm}^i = P_{dcm}^{\min} + rand^i \times (P_{dcm}^{\max} - P_{dcm}^{\min})$	**For** *j*=1, 2, …, *ND*
For *j*=1, 2, …, *ND*	$P_{dcm}^j = P_{dcm}^{\min} + rand^j \times (P_{dcm}^{\max} - P_{dcm}^{\min})$
If $rand^j < P_{dcm}^i$	**If** $rand^j < P_{dcm}^j$
If $rand^k <= P_d$	**If** $rand^k <= P_d$
$X_{t+1}^i = \max[D^{\min}, X_t^i - 1]$	$X_{t+1}^i = \max[D^{\min}, X_t^i - 1]$
Else	**Else**
$X_{t+1}^i = \min[D^{\max}, X_t^i + 1]$	$X_{t+1}^i = \min[D^{\max}, X_t^i + 1]$
End If	**End If**
End If	**End If**
End For	**End For**
End For	**End For**

Figure 3.7 Pseudo-code of the dither creeping mutation

3.2.3.5. Paradigm of Hybridisation

As mentioned in Chapter 2, Talbi (2002) presented a taxonomy of hybrid algorithms, defining four basic types including low-level relay hybrid, low-level teamwork hybrid, high-level relay hybrid and high-level teamwork hybrid. A low-level hybrid method, either in the relay or teamwork fashion, usually embeds another metaheuristic into the main algorithm. Alternatively, a certain functional part of the main algorithm is replaced by another metaheuristic. For example, many low-level hybrid GAs incorporate a local search procedure (typically a deterministic method) to intensify the refinement of solutions found so far. In contrast, a high-level hybrid method uses different metaheuristics as 'self-contained', which means that no interruption exists in between these operators.

As discussed in the previous subsection, the reason why six distinct operators are selected for the GALAXY method is that they are probably suitable to be highly efficient at different stages of the optimisation. Therefore, it may not make sense to blend these operators in a low-level hybrid framework. Furthermore, as concluded in Wang et al. (2014b), a high-level teamwork hybrid turned out to be more effective than a high-level relay hybrid, since the former was able to take full advantage of each method promptly and eliminated on-line performance evaluation of each operator (saving computational time). On the other hand, a high-level teamwork hybrid framework fits itself well for parallel computation, although it is out of the scope of this thesis.

Therefore, it is believed that a high-level teamwork hybrid paradigm is more likely to produce better results in an efficient way. Thus, the GALAXY method adopts such a hybrid paradigm and embeds six aforementioned search operators in a unified structure.

3.2.4. Strategies of GALAXY

3.2.4.1. Constraint Handling Strategy

Many constraint handling techniques exist in the literature (as discussed in Chapter 2). Two commonly used methods are the penalty-function based approach and the constrained-domination based approach. The penalty-function based method usually requires the specification of a scaling factor to discriminate infeasible solutions from feasible ones by adding a reasonable large penalty to their objective values. A drawback of this method lies in the fact that a proper factor often relies on trial-and-error which may not be viable when dealing with large and complex design problems. In contrast, the constrained-domination based method does not compare constraints with objective function values, thus eliminating the penalty parameter. It works as follows: assuming that two solutions (*A* and *B*) are randomly chosen from the population for comparison, solution *A* is selected if: (i) *A* has a less constraint violation than *B*; or (ii) *A* dominates *B* when both are feasible solutions. Otherwise, one of them is chosen at random.

Although some more advanced constraint handling techniques have been proposed, such as the Inverse Parabolic Spread method introduced by Padhye et al. (Padhye et al. 2013b), in this thesis a traditional penalty-free, constrained-domination strategy (Deb et al. 2002) is adopted due to its simplicity and effectiveness.

3.2.4.2. Genetically Adaptive Strategy

One effective strategy for a high-level hybrid framework is the genetically adaptive, multi-algorithm search initially proposed in Vrugt and Robinson (2007), which was adopted by both AMALGAM and Borg. This strategy within the original AMALGAM framework works as follows. Firstly, the offspring pool Q_0 of size N is created from the initial population P_0 using four search operators simultaneously, with each operator contributing the same number of individuals (i.e., $N/4$). Next, a combination of the parents (P_0) and the offspring (Q_0), namely R_0 (size $2N$), is produced and ranked via the fast non-dominated sorting procedure (Deb et al. 2002). A number of individuals, N, from R_0 are then selected based on their rank and crowding distance, forming the population for the next generation. Thereafter, the number of individuals produced by each search operator is determined according to the reproductive rate (ratio of the children alive to the children created) in the previous generation (see Eq. (3.14)). However, if a search operator fails to contribute even a single individual in the latest population, a minimum number of individuals are consistently maintained for it to generate in the next round.

$$N_t^i = N \cdot (P_t^i / N_{t-1}^i) / \sum_{i=1}^{k} (P_t^i / N_{t-1}^i) \qquad (3.14)$$

Where, N_t^i - the number of offspring to be generated by operator i at generation t; N - population size; P_t^i - number of offspring contributed by operator i at generation t.

Since GALAXY incorporates six operators for solving discrete combinatorial problems, this genetically adaptive strategy is also adopted to facilitate an effective search using multiple methods simultaneously. Nevertheless, this strategy needs to be enhanced to maximise its impact on the dynamic performance of various search operators, which is specified in the following paragraphs.

In fact, the threshold made in the original AMALGAM seems to be arbitrary and thus can be further improved. Because the genetically adaptive strategy itself can award or penalize search operators automatically according to their performances, it may not make sense to keep the inefficient operators at such a minimum value (i.e., 5). As shown in Wang et al. (2014b), when dealing with a more complex design problem, the behavior of AMALGAM was similar to NSGA-II with a smaller effective population size, leading to worse non-dominated solutions. The reason for such a poor performance was that all the operators except the GA became incapable of finding high quality solutions, therefore AMALGAM worked like NSGA-II with the effective population size shrunk by 15%.

To overcome this drawback, a subtle modification is introduced by reducing the minimum threshold from 5 to 1. In case that one of six operators fails to contribute a single child to the next population, it borrows one opportunity from the topmost operator in the previous round. If two of six operators fail, each of them borrows one opportunity from the two topmost operators, and so on. Therefore, the most successful operator (with the highest reproductive rate) is always favored by getting the highest number of offspring in the reproduction process, and no operator is completely discarded (i.e., at least one offspring to be generated) even though it exhibits the worst performance.

3.2.4.3. Global Information Sharing Strategy

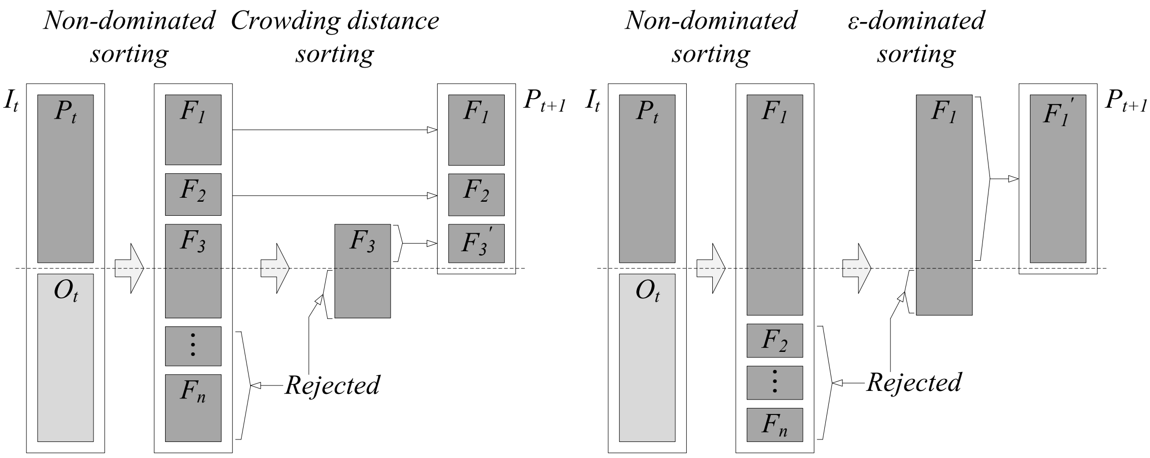
This strategy was first mentioned by Vrugt and Robinson (2007), but was not well explained. The essential idea is to share the current population among each search operator in the *Generation* step. In particular, each operator generates the same number of candidate solutions, which is equal to the population size, rather than just the number of individuals it is allowed to produce. This permits the not-so-good operators to benefit from the high quality solutions contributed by other operators. Then, a number of offspring, which is determined by the genetically adaptive strategy, are chosen at random from the candidate solutions produced by each operator. This global information sharing strategy allows the search operators with poor performance in previous generations to recover subsequently, thus increasing their opportunities to contribute better solutions in the later run. Note that this strategy can be only realised in the generational framework.

3.2.4.4. Duplicates Handling Strategy

It is worth noting that there is a much higher possibility that duplicate solutions will be generated in the context of discrete combinatorial problems than in real-parameter continuous problems. Consequently, the possibility of discovering more and/or better solutions is wasted due to some solutions being repeated in the population. Furthermore, the actual number of non-dominated solutions is less likely to reach the population size at the end of the search. Some researchers (Chaiyaratana et al. 2007; Črepinšek et al. 2013) have suggested discarding duplicate individuals to improve the overall performance (mainly for diversity) of evolutionary algorithms. Hence, a 'unique solution strategy' is applied in the GALAXY method after the combination of the offspring and the current population, as shown in Figure 3.2. This strategy is aimed at checking and removing duplicate solutions that exist in the population thereby increasing the diversity by accommodating more non-dominated solutions. This method is also believed to prevent premature convergence of the population.

3.2.4.5. Hybrid Replacement Strategy

In MOEAs, the *Replacement* procedure plays a crucial role in steering the population towards the PF. As shown in Figure 3.2, a hybrid replacement strategy is implemented in the GALAXY framework. Specifically, after the creation of an intermediate population (I_t) using the unique solutions in the current population (P_t) and the offspring (O_t), the individuals are ranked using the fast non-dominated sorting (Deb et al. 2002). If the number of individuals in the top rank is no greater than the population size, the normal replacement is carried out as in the NSGA-II. Figure 3.8(a) demonstrates this procedure that the solutions with rank F_1 and F_2 are copied directly into the next population (P_{t+1}). The solutions with rank F_3 are first sorted in a descending order of their crowding distance values. Then, a number of solutions (F_3'), which is equal to the remaining spaces in the next population, are chosen from top to bottom. However, if the number of individuals with rank F_1 exceeds the population size, the ε-replacement (see Figure 3.8(b)) is carried out instead, in which the non-dominated solutions in the first front are sorted once again based on the ε-dominance concept. Therefore, the ε-non-dominated solutions are copied into the next population (P_{t+1}). If there are still free spaces left, some ε-dominated solutions are also selected which have smaller distances to the ideal global optima (i.e., the point with the best value according to each objective function).



(a) normal replacement ($F_1 <= N$) (b) ε-replacement ($F_1 > N$)

Figure 3.8 Concept of hybrid replacement strategy

This hybrid replacement strategy is used to preserve the non-dominated solutions which may have reached the PF. In other words, the ε-dominance comparison is employed to overcome the shortcoming of the crowding distance based sorting, which by aiming to achieve a relatively uniform spread usually leads to losing the solutions on the PF. Actually, for a discrete combinatorial optimisation problem, it is rare that the PF is uniformly distributed. Therefore, the crowding distance based sorting, which was proposed for solving continuous problems, may become problematic.

On the other hand, the ε-replacement procedure implemented in this thesis differs from the traditional ε-dominance comparison in two aspects. Firstly, it does not require any user-specified ε precision for each objective. Instead, the ε precisions are determined internally according to the extent of solutions in the first front. Secondly, it allows some ε-dominated solutions to be selected, rather than being completely removed, provided that they are close to the global optima. Figure 3.9 depicts this idea in more details. First of all, the boundary solutions in the first front are identified and used to construct the 2-D boxes with ε_i as the side length in the *i*-th objective. The value of ε_i is calculated by Eq. (3.15). Note that the boundary solutions must be included in the next population. Therefore, the number of candidate solutions to be selected is *N-2*. In Figure 3.9, the solid circles represent ε-dominating solutions (e.g., *B* and *C*) in the first front; while the dashed circles represent ε-dominated solutions (e.g., *A* and *D*). The grey area indicates the region dominated by ε-dominating solutions. Since the number of non-dominated solutions in the 2-D boxes is larger than *N*-2, they are first compared using the ε-domination concept. The ε-dominating solutions are then included in the next population. If there are remaining spaces, the ε-dominated solutions with smaller Euclidean distances to the global optimum (i.e., the lower-right corner in Figure 3.9) are also chosen.

$$\varepsilon_i = \frac{f_i^{\max} - f_i^{\min}}{N-2} \tag{3.15}$$

Where, ε_i - the side length of each box in the *i*-th objective; $f_i^{\max}$ and $f_i^{\min}$ - maximum and minimum values of the *i*-th objective, respectively; *N* - population size.

As a result, this hybrid replacement strategy ensures that the non-dominated solutions in the first front, which are near to the global optima, are preserved. A good diversity of these solutions is also achieved via the ε-replacement procedure. As it will be shown in Chapter 5, this strategy along with the duplicates handling strategy significantly improves the ε-performance metric (Kollat and Reed 2006) of the GALAXY method.

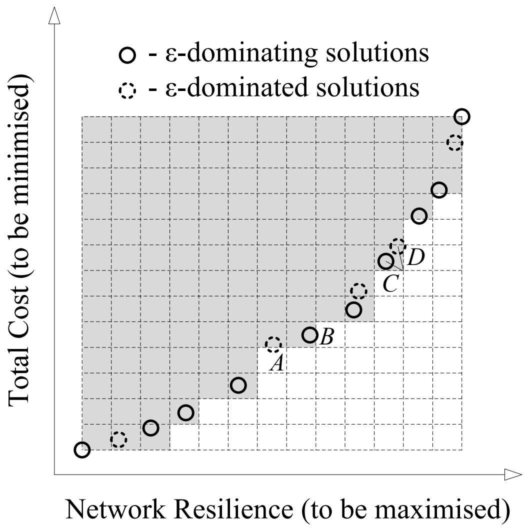


Figure 3.9 Concept of ε-dominated sorting

3.3. Sensitivity Analysis

Since the GALAXY method actually involves only one parameter, namely the probability of mutation for the UM, GM and DC. All the mutation operators are applied at a probability equal to 1/*ND*, which has been proved to be an effective setting (Goldberg 1989) and widely accepted by most researchers. Besides the mutation rate, DC still requires specifying the bounds of mutation rate, because for each gene the mutation probability of DC (denoted as P_{dcm}) samples a uniformly distributed random number within the bounds rather than a prefixed value. Therefore, it is necessary to investigate the sensitivity of the performance of *DC* with regard to the lower and upper bounds of mutation rate (denoted as $P_{dcm}^{\min}$ and $P_{dcm}^{\max}$, respectively). As suggested in Zheng et al. (2013c), a small interval of size $O(1/ND)$ was chosen to form the bounds of mutation rate (see Eq. (3.16)-(3.18)) centred at P_{dcm}. Therefore, in this section the performance of *DC* is tested under different ranges of P_{dcm} from 0.01 to 0.90.

$$\left|P_{dcm}^{\max} - P_{dcm}^{\min}\right| = O(P_{dcm}) \tag{3.16}$$

$$P_{dcm}^{\min} = (1 - range) \times P_{dcm} \quad (3.17)$$

$$P_{dcm}^{\max} = (1 + range) \times P_{dcm} \quad (3.18)$$

The quality of non-dominated solutions obtained by different bounds of mutation rate is evaluated via four performance indicators mentioned in Chapter 2, which are the generational distance (I_{GD}), hypervolume (I_{HV}), additive ε-indicator ($I_{\varepsilon+}$) and ε-performance (I_{EP}). The detailed definition and mathematical formulation of these indicators can be found in Chapter 2 and Chapter 5, respectively. In addition, these performance indicators are normalised to reside between 0 and 1, with 1 representing the ideal (best) value.

Note that, the sensitivity analysis of the *DC* operator is not carried out for all the benchmark problems considered in this thesis, since it would inevitably incur expensive computational overhead. Instead, the tests are implemented on small to intermediate sized design problems and the overall achievements are compared in Table 3.2. The numbers in the table below are presented in the format of 'mean (standard deviation)', which are statistical results across multiple runs (30) on a total of 9 benchmark problems.

Table 3.2 Average performance of *DC* measured by different performance indicators

Range	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
0.01	0.9892 (0.0163)	0.9556 (0.0432)	0.9593 (0.0365)	0.2189 (0.2556)
0.05	0.9892 (0.0164)	0.9557 (0.0439)	0.9596 (0.0366)	0.2197 (0.2601)
0.10	**0.9894 (0.0159)**	0.9555 (0.0435)	0.9591 (0.0366)	**0.2229 (0.2618)**
0.15	0.9891 (0.0163)	0.9553 (0.0440)	0.9590 (0.0367)	0.2169 (0.2551)
0.20	0.9891 (0.0164)	0.9561 (0.0435)	0.9598 (0.0365)	0.2183 (0.2549)
0.25	0.9892 (0.0165)	0.9558 (0.0447)	0.9595 (0.0375)	0.2211 (0.2562)
0.30	0.9893 (0.0162)	0.9554 (0.0435)	0.9593 (0.0362)	0.2161 (0.2527)
0.35	0.9891 (0.0166)	0.9557 (0.0440)	0.9595 (0.0370)	0.2207 (0.2572)
0.40	0.9893 (0.0164)	0.9553 (0.0439)	0.9588 (0.0369)	0.2193 (0.2568)
0.45	0.9890 (0.0170)	0.9552 (0.0444)	0.9593 (0.0372)	0.2218 (0.2586)
0.50	0.9890 (0.0167)	0.9552 (0.0444)	0.9594 (0.0367)	0.2191 (0.2544)
0.60	0.9892 (0.0163)	0.9554 (0.0444)	0.9590 (0.0375)	0.2196 (0.2566)
0.70	0.9890 (0.0170)	**0.9562 (0.0443)**	**0.9603 (0.0371)**	0.2202 (0.2594)
0.80	0.9891 (0.0167)	0.9559 (0.0455)	0.9593 (0.0381)	0.2190 (0.2579)
0.90	0.9890 (0.0168)	0.9550 (0.0444)	0.9588 (0.0370)	0.2173 (0.2532)

Note: The best practice on average in terms of each performance indicator (column-wise) is shown in bold.

The analysis shows that the performance of *DC* is not very sensitive to the bounds of P_{dcm}, as long as the probability of mutation is kept at a reasonable low value (e.g., 1/*ND*). This is a preferred feature for such an operator. By considering the overall performance in terms of convergence, diversity and consistency, a range equal to 0.70 is select to calculate the bounds of P_{dcm} throughout this thesis.

3.4. Summary

This chapter presented a new hybrid algorithm, called GALAXY, for the multi-objective design or rehabilitation of WDSs. In Section 3.2, some essential questions with regard to the characteristics of multi-objective design or rehabilitation of WDSs and the creation of a hybrid algorithm were first raised. Then, a framework for the GALAXY method was proposed based on the Evolutionary Optimisation System followed by the detailed development of each component. Finally, several important strategies implemented in the GALAXY framework were explained. Since the GALAXY method just involves one individual parameter (i.e., the probability bounds of the dither creeping mutation), a sensitivity analysis towards this parameter was carried out in Section 3.3, proving that GALAXY was not significantly affected by this parameter. As a result, the number of user-specified parameters within the GALAXY method was significantly reduced.

4. WATER DISTRIBUTION SYSTEM BENCHMARK ARCHIVE

4.1. Introduction

It is a common practice to verify the capability of a newly-developed algorithm on benchmark problems by comparing it with other algorithms. In the field of water resources engineering, there are quite a few benchmark problems which are available for testing and comparing MOEAs. For example, Reed et al. (2013) presented three many-objective test problems, including the rainfall-runoff model calibration, the groundwater monitoring design and the water supply portfolio planning. These cases introduce a certain degree of difficulty in the landscape of the search space, being deceptive, multi-modal, discontinuous and highly constrained. However, in the domain of WDSs, only a few benchmark networks are frequently used to address the optimal design problems, such as the two-loop network (Alperovits and Shamir 1977), the New York tunnel network (Schaake and Lai 1969) and the Hanoi network (Fujiwara and Khang 1990). These networks are usually small and therefore, it is not easy to distinguish the performance of different algorithms, or to draw conclusions which can benefit (large) real-world cases.

On the other hand, from the viewpoint of multi-objective formulation, there is no universally accepted definition that can facilitate the direct comparison of various algorithms. More specifically, for the two-objective design of WDSs, one objective is generally the total capital cost, whereas the other one varies from case to case (as reviewed in Chapter 2). The unconstrained problem makes all solutions appear feasible so that the algorithm can search at the boundary of the feasible region. In that case, it may be difficult to find a solution within the feasible region. Furthermore, an inappropriate formulation may be criticised by practitioners as the optimal solutions are less useful or even misleading for real cases.

One goal of using optimisation algorithms is to solve more complex problems in the real world. Nevertheless, it is difficult to choose the best method to implement from the existing algorithms in the domain, because there are limited comparison studies of these algorithms for a wide range of problems (especially on large ones). Hence, as mentioned by Farmani et al. (2003a), it may be beneficial for optimisation researchers and practitioners in the water community to build up a library of cases to benchmark the target algorithms against others in a thorough and systematic fashion.

This chapter aims to set out an archive of benchmark problems, termed Water Distribution System Benchmark Archive (WDSBA), to compare the proposed hybrid algorithms with state-of-the-art hybrid frameworks and other MOEAs. A uniform problem formulation is applied throughout this thesis for the two-objective design or extended design of WDSs. Up to twelve problems featuring distinct characteristics and difficulties are included in the WDSBA, ranging from small to large.

4.2. Uniform Problem Formulation

A uniform formulation, i.e., same in terms of objective functions, types of decision variables and constraint violations, is applied to all the benchmark problems in order to compare different algorithms under a standardised framework. Therefore, the complexity among different benchmark problems mainly comes from the size of search space, which depends on the number of pipes in a specific network and the number of diameter (or intervention for rehabilitation cases) options for each pipe. By comparing algorithms on a wide range of benchmark problems, it is possible to draw a general conclusion from the overall performance of each algorithm.

4.2.1. Objective Functions

In the WDSBA, two main objectives are considered, the total capital cost associated with pipes and the network resilience (Prasad and Park 2004). The specific objectives are expressed in Eq. (4.1) and Eq. (4.2) respectively.

$$\min C = \sum_{i=1}^{np} C_u(D_i) \times L_i \tag{4.1}$$

Where, C - total cost (problem dependent monetary unit); np - number of pipes; C_u - unit pipe cost depending on the diameter selected in a specific problem; D_i - diameter of pipe i; L_i - length of pipe i.

$$\max I_n = \frac{\sum_{j=1}^{nn} C_j Q_j (H_j - H_j^{\min})}{(\sum_{k=1}^{nr} Q_k H_k + \sum_{i=1}^{npu} \frac{P_i}{\gamma}) - \sum_{j=1}^{nn} Q_j H_j^{\min}} \tag{4.2}$$

$$C_j = \frac{\sum_{i=1}^{npj} D_i^j}{npj \times \max\{D_i^j\}} \tag{4.3}$$

Where, I_n - network resilience; nn - number of demand nodes; C_j, Q_j, H_j and H_j^{min} - uniformity, demand, actual head and minimum head of node j; nr - number of reservoirs; Q_k and H_k - discharge and actual head of reservoir k; npu - number of pumps; P_i - power of pump i; γ - specific weight of water; npj - number of pipes connected to node j; D_i^j - diameter of pipe i connected to demand node j.

4.2.2. Decision Variables

The decision variables in each problem within the WDSBA are the diameters (or intervention options for rehabilitation cases) of pipes. Due to the availability of pipe sizes on the market (or from manufacturers), the diameter options are of discrete sizes. Thus, the decision variables are of the consecutive integer type, mapping to each available diameter size. The expression of decision variables is shown in Eq. (4.4).

$$DV = \{D_1, D_2, ...D_i, ..., D_{np}\}; \quad D_i \in D \tag{4.4}$$

Where, DV - decision variables; D_i - diameter index of pipe i; np - number of pipes to be optimised; D - set of diameter options of a specific benchmark problem.

4.2.3. Constraints

The constraints in the uniform problem formulation consist of two parts, implicit and explicit constraints. Implicit constraints are the hard constraints that must be fulfilled during the optimisation (simulation) .They are controlled by the laws of mass balance and energy conservation of a network system. This means that at each demand node the water consumption is equal to the difference of the water flowing into and out of that node, and the sum of the head loss due to the friction of pipe walls within each loop is equal to zero. The implicit constraints are expressed in Eq. (4.5) and Eq. (4.6) respectively. These constraints are automatically satisfied by using a hydraulic solver (e.g., EPANET2).

$$Q_i = \sum Q_i^{in} - \sum Q_i^{out}; \quad \forall i \in J \tag{4.5}$$

Where, Q_i - water demand of node i; Q_i^{in} - water flowing into node i; Q_i^{out} - water flowing out of node i; J - set of demand nodes.

$$\sum_{i=1}^{npl} \Delta H_i^l = 0; \quad \forall l \in L \tag{4.6}$$

Where, npl - number of pipes in loop l; ΔH_i^l - head loss in pipe i within loop l; L - set of basic loops in a network system.

The hydraulic head loss in a pipe due to the friction of pipe walls can be computed using different formulae. Two commonly used formulae for WDSs are the Hazen-Williams formula and the Darcy-Weisbach formula. The former is widely used in the United States., whereas the latter is theoretically correct and can be applied to all liquids over all flow regimes. A generalised formula for computing the hydraulic head loss (ΔH) in a pipe is shown in Eq. (4.7). Two specific variants of this formula, i.e. the Hazen-Williams formula and the Darcy-Weisbach formula are given in Eq. (4.8) and (4.9) respectively.

$$\Delta H = Aq^B \tag{4.7}$$

Where, ΔH - head loss in a pipe; A - resistance coefficient; q - flow rate; B - flow exponent.

$$\Delta H = \frac{10.67 q^{1.852} L}{C^{1.852} d^{4.871}} \tag{4.8}$$

$$\Delta H = \frac{8 f(\kappa, d, q) q^2 L}{g \pi^2 d^5} \tag{4.9}$$

Where ΔH - head loss in a pipe (m); q - flow rate (m^3/s); L - pipe length (m); C - Hazen-Williams roughness coefficient (unitless); d - pipe diameter (m); κ - roughness height (m); $f(\kappa,d,q)$ - Darcy friction factor (unitless) depending on $\kappa,\ d,$ and q.

Explicit constraints relate to those performance requirements of a network system. The pressure head requirement at demand nodes and the flow velocity in pipes are commonly considered during the design stage. There is only minimum pressure head requirement of demand nodes in most benchmark problems of WDSBA. However, four out of twelve problems (as shown in Section 4.3) have both minimum and maximum pressure head requirements for the demand nodes, which further limit the search space. At the same time, these four problems also have upper bounds of flow velocity throughout the networks. These explicit constraints are shown in Eq. (4.10) and Eq. (4.11) respectively.

$$H_i^{\min} \leqslant H_i \leqslant H_i^{\max};\ \ \forall i \in J \qquad (4.10)$$

Where, H_i - head at node i; H_i^{min} - minimum head requirement at node i; H_i^{max} - maximum head requirement at node i; J - set of demand nodes.

$$V_i \leqslant V^{\max};\ \ \forall i \in P \qquad (4.11)$$

Where, V_i - flow velocity in pipe i; V^{max} - maximum flow velocity requirement; P - set of pipes.

4.3. Water Distribution System Benchmark Archive

A detailed description of each benchmark network is presented below. The description follows a uniform format, including a brief history of the network, a summary of the network components, design requirement (pressure head and flow velocity if any) and size of search space. In addition, the primary data, i.e., design criteria and diameter options and the associated unit prices are given in this section. Note that the benchmark networks are arranged according to the ascending order of search space.

4.3.1. Description of Benchmark Networks

4.3.1.1. Two Reservoir Network (TRN)

The TRN was first introduced by Gessler (1985) and later modified by Simpson et al. (1994) by adding two large diameter options for new pipes. It represents a benchmark network which considers multiple loading conditions, including one normal scenario and two fire flow scenarios.

The TRN has eight pipes to be considered and the other existing pipes cannot be changed. There are two reservoirs with heads fixed at 365.76 m (left in Figure 4.1) and 371.86 m (right in Figure 4.1) and nine demand nodes. New pipes and cleaned pipes have the same Hazen-Williams roughness coefficient of 120. The minimum pressure of all the nodes under three demand patterns is specified in Table 4.1. The decision variables are the pipe diameters for five new pipes and alternative options, i.e., duplication or cleaning or 'do nothing', for three existing pipes. Each pipe has eight diameter options to choose from. Therefore, the search space is equal to $8^5 x 10^3 \approx 3.28 x 10^7$ discrete combinations, which makes the TRN the smallest combinatorial problem analysed in this thesis. Table 4.2 shows the available diameter options for pipe diameter and the associated unit costs. Figure 4.1 depicts the layout of TRN. The new pipes are shown as thick solid lines; whereas the new duplication pipes are shown as thick dashed lines.

4.3.1.2. Two Loop Network (TLN)

The TLN is arguably the simplest WDS benchmark network for design purposes in the literature. It was first presented by Alperovits and Shamir (1977) as a hypothetical network. The currency unit of cost has been chosen here to be the US dollar.

The TLN contains eight pipes organised in two loops, six demand nodes under a single loading condition and is supplied by a reservoir with a constant head of 210.0 m. As it is a hypothetical network, all the pipes have the same length (1000.0 m) and a Hazen-Williams coefficient of 130. The pressure is set to be at least 30.0 m at all the demand nodes. In this thesis, the limits on minimum and maximum head loss gradient of each pipe proposed in the original paper are ignored. Fourteen diameter options are considered for each pipe, ranging from 25.4 mm (1.0 in.) to 609.6 mm (24.0 in.). Thus, the search space is equal to $14^8 \approx 1.48 \times 10^9$ discrete combinations. Table 4.3 shows the available pipe diameters and the associated unit costs. Figure 4.2 depicts the layout of TLN.

Table 4.1 Minimum pressure at each node under three demand patterns of TRN

Node ID	Loading Conditions					
	Normal Condition		Fire Flow 1		Fire Flow 2	
	Demand (l/s)	H^{min} (m)	Demand (l/s)	H^{min} (m)	Demand (l/s)	H^{min} (m)
2	12.62	28.18	12.62	14.09	12.62	14.09
3	12.62	17.61	12.62	14.09	12.62	14.09
4	0	17.61	0	14.09	0	14.09
6	18.93	35.22	18.93	14.09	18.93	14.09
7	18.93	35.22	82.03	10.57	18.93	14.09
8	18.93	35.22	18.93	14.09	18.93	14.09
9	12.62	35.22	12.62	14.09	12.62	14.09
10	18.93	35.22	18.93	14.09	18.93	14.09
11	18.93	35.22	18.93	14.09	18.93	14.09
12	12.62	35.22	12.62	14.09	50.48	10.57

Note: H^{min} - minimum pressure head requirement.

Table 4.2 Pipe intervention options and associated unit costs of TRN

Diameter (mm)	Unit Cost		Diameter (mm)	Unit Cost	
	New Pipe ($/m)	Cleaned Pipe ($/m)		New Pipe ($/m)	Cleaned Pipe ($/m)
152	49.54	47.57	356	170.93	60.70
203	63.32	51.51	407	194.88	63.00
254	94.82	55.12	458	232.94	-
305	132.87	58.07	509	264.10	-

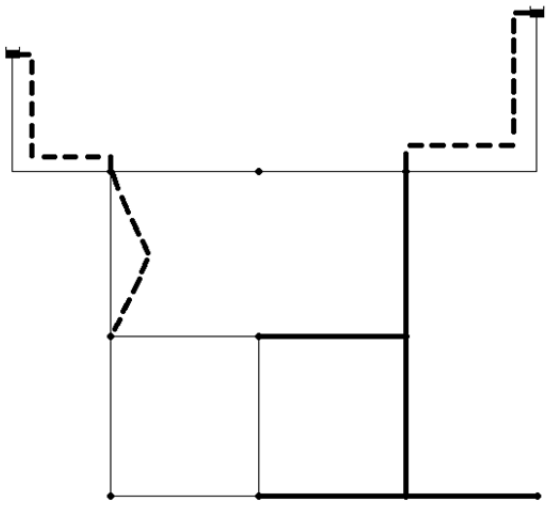


Figure 4.1 Layout of TRN

Table 4.3 Diameter options and associated unit costs of TLN

Diameter (mm/in.)	Unit Cost ($/m)	Diameter (mm/in.)	Unit Cost ($/m)	Diameter (mm/in.)	Unit Cost ($/m)	Diameter (mm/in.)	Unit Cost ($/m)
25.4/1.0	2.0	152.4/6.0	16.0	355.6/14.0	60.0	558.8/22.0	300.0
50.8/2.0	5.0	203.2/8.0	23.0	406.4/16.0	90.0	609.6/24.0	550.0
76.2/3.0	8.0	254.0/10.0	32.0	457.2/18.0	130.0	-	-
101.6/4.0	11.0	304.8/12.0	50.0	508.0/20.0	170.0	-	-

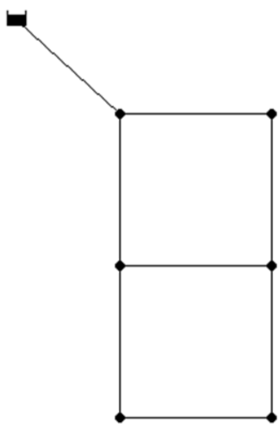


Figure 4.2 Layout of TLN

4.3.1.3. BakRyan Network (BAK)

The BAK, a combination of a design and extended design problem, was first introduced by Lee and Lee (2001) and it represents a water supply system in a city of South Korea.

The BAK has fifty-eight pipes including nine new pipes to be determined, thirty-five demand nodes, and one reservoir with a fixed head of 58.0 m. The Hazen-Williams roughness coefficient for each new pipe is 100. The minimum pressure head above the ground elevation of each node is 15.0 m. Among the new pipes, six of them are parallel. There are eleven available pipe sizes, ranging from 300.0 mm to 1,100.0 mm. Therefore, the search space is equal to $11^9 \approx 2.36 \times 10^9$ discrete combinations. Table 4.4 shows the diameter options and the associated unit costs. Figure 4.3 depicts the layout of BAK. The pipes to be considered are shown in thick solid lines.

Table 4.4 Diameter options and associated unit costs of BAK

Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)
300.0	118.0	450.0	160.0	700.0	242.0	1,000.0	370.0
350.0	129.0	500.0	181.0	800.0	285.0	1,100.0	434.0
400.0	145.0	600.0	214.0	900.0	325.0	-	-

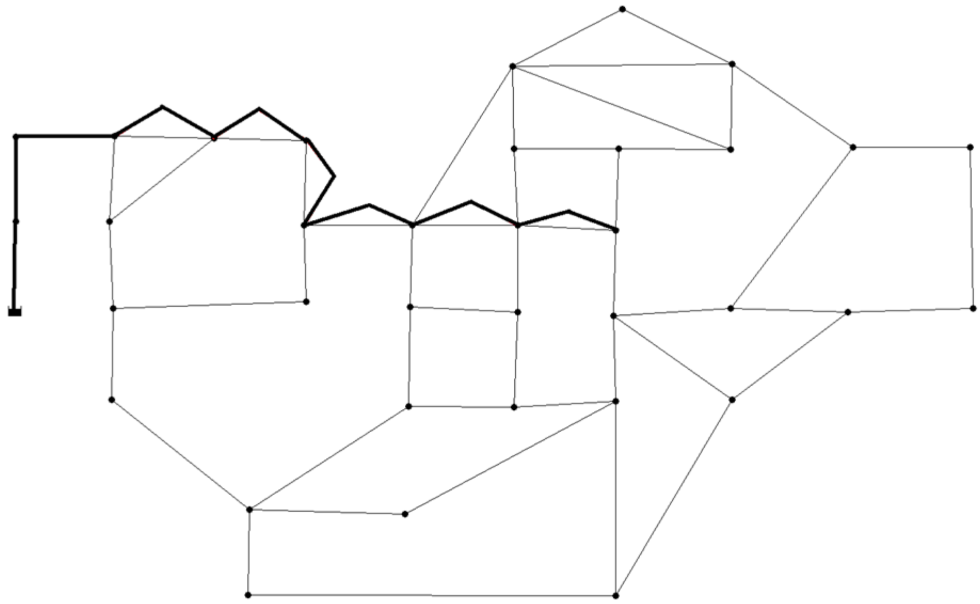


Figure 4.3 Layout of BAK

4.3.1.4. New York Tunnel Network (NYT)

The NYT was first proposed by Schaake and Lai (1969) as a rehabilitation activity undertaken to an existing tunnel system which was the primary water distribution system of the city of New York, USA.

The NYT is comprised of twenty-one pipes organised in two loops, nineteen demand nodes, and one reservoir with a fixed head of 91.44 m (300 ft). All the existing pipes are considered for duplication in order to meet the projected future demand. The Hazen-Williams roughness coefficient for both new and existing pipes is 100. The minimum pressure of all the demand nodes is fixed at 77.72 m (255 ft) except for nodes 16 and 17 that are 79.25 m (260 ft) and 83.15 m (272.8 ft), respectively. A selection of fifteen diameter sizes is available as well as a ‘do nothing’ option. Therefore, the search space is equal to $16^{21} \approx 1.93 \times 10^{25}$ discrete combinations. Table 4.5 shows the diameter options and the associated unit costs. Figure 4.4 depicts the layout of NYT. The new duplication pipes are shown as thick solid lines.

Table 4.5 Diameter options and associated unit costs of NYT

Diameter (mm/in.)	Unit Cost ($/m)	Diameter (mm/in.)	Unit Cost ($/m)	Diameter (mm/in.)	Unit Cost ($/m)
914.4/36.0	307.05	2438.4/96.0	1036.09	3962.4/156.0	1891.70
1219.2/48.0	438.65	2743.2/108.0	1199.02	4267.2/168.0	2073.79
1524.0/60.0	578.48	3048.0/120.0	1366.34	4572.0/180.0	2258.99
1828.8/72.0	725.23	3352.8/132.0	1537.76	4876.8/192.0	2447.21
2133.6/84.0	877.99	3657.6/144.0	1712.96	5181.6/204.0	2638.25

4.3.1.5. Blacksburg Network (BLA)

The BLA was first presented by Sherali et al. (2001) as a newly expanded subdivision of the water distribution system in the town of Blacksburg, Virginia. The network was then slightly modified by Bragalli et al. (2008) since the data was incomplete in the original paper and this modified network has been used throughout the thesis.

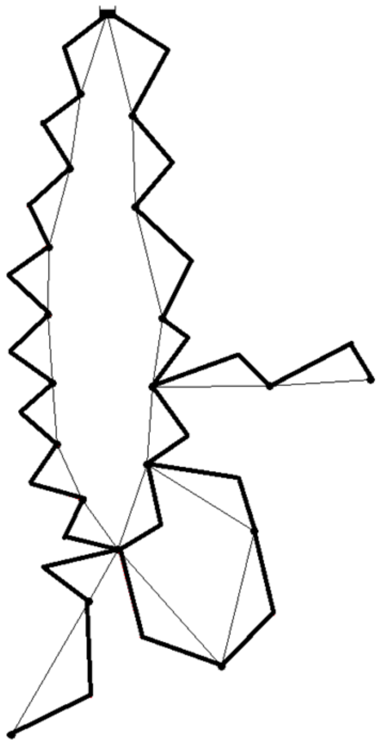


Figure 4.4 Layout of NYT

The BLA consists of thirty-five pipes of which twelve have fixed diameters, one reservoir with a fixed head of 715.56 m[2], and 30 demand nodes. A universal Hazen-Williams coefficient of 120 is applied to all the pipes under consideration. The pressure requirement of each node is limited within a specified range under the single loading condition. The minimum pressure head for each node is 30.0 m, while the maximum pressure head varies from node to node and these values are provided in Table 4.6. In addition, there is also a constraint on the maximum flow velocity (2.0 m/s) for all the pipes. The total number of decision variables for the BLA problem is twenty-three. According to Bragalli et al. (2008), there are fourteen[3] diameter options resulting in the search space equal to $14^{23} \approx 2.30 \times 10^{26}$ discrete combinations. Table 4.7 shows the commercially available pipe sizes and the associated unit costs. Figure 4.5 depicts the layout of BLA.

Table 4.6 Maximum pressure head requirement at each node of BLA

N_I	H^{max} (m)	N_I	H^{max} (m)	N_I	H^{max} (m)	N_I	H^{max} (m)	N_I	H^{max} (m)	N_I	H^{max} (m)
1	62.99	6	66.65	11	63.60	16	60.55	21	62.84	26	75.64
2	65.73	7	67.26	12	72.44	17	72.74	22	62.08	27	74.88
3	69.09	8	67.26	13	64.36	18	62.08	23	58.27	28	75.94
4	59.18	9	72.59	14	62.38	19	60.40	24	51.71	29	69.39
5	62.84	10	69.09	15	61.92	20	63.29	25	70.00	30	68.48

Note: N_I - node index which is a consecutive number starting from one to the total number of nodes in the network; H^{max} - maximum pressure head requirement.

[2] This number is much higher than the original elevation of 659.28 m (2163 ft) in Sherali et al. (2001).

[3] This number was firstly introduced in Bragalli et al. (2008) and thereafter modified by adding another option (pipe elimination) in Raad's PhD thesis. Herein, the option of pipe elimination is ignored as it will cause some end nodes disconnected from the water source.

Table 4.7 Diameter options and associated unit costs of BLA

Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)
25.40	0.52	152.40	18.90	355.60	102.89	558.80	254.08
50.80	2.10	203.20	33.60	406.40	134.39	609.60	302.37
76.20	4.72	254.00	52.50	457.20	170.09	-	-
101.60	8.40	304.80	75.59	508.00	209.98	-	-

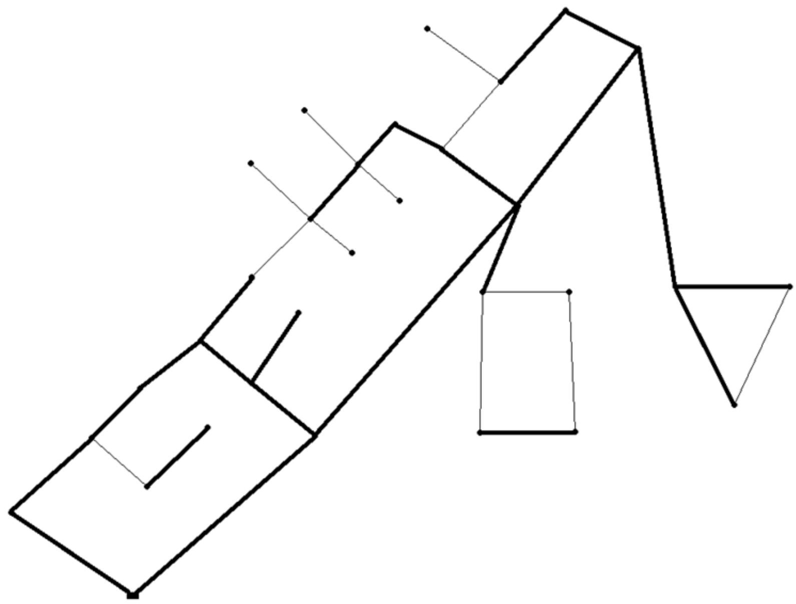


Figure 4.5 Layout of BLA

4.3.1.6. Hanoi Network (HAN)

The HAN was first used by Fujiwara and Khang (1990) as a design problem and it represents a water distribution system in Hanoi, the capital of Vietnam. The Hanoi network has a large region of infeasible solutions in the landscape of decision variables, thus making it difficult to explore the search space (Dong et al. 2012).

The HAN consists of thirty-four pipes organised in three loops, thirty-one demand nodes and one reservoir with a fixed head of 100.0 m. The Hazen-Williams roughness coefficient for all pipes is 130. The minimum head above the ground elevation of each node is 30.0 m. There are six commercially available pipe sizes, ranging from 304.8 mm (12.0 in.) to 1016.0 mm (40.0 in.).

Therefore the search space is equal to $6^{34} \approx 2.87 \times 10^{26}$ discrete combinations. Table 4.8 shows the diameter options and the associated unit costs. Figure 4.6 depicts the layout of HAN.

Table 4.8 Diameter options and associated unit costs of HAN

Diameter (mm/in.)	Unit Cost ($/m)	Diameter (mm/in.)	Unit Cost ($/m)	Diameter (mm/in.)	Unit Cost ($/m)
304.8/12.0	45.73	508.0/20.0	98.39	762.0/30.0	180.75
406.4/16.0	70.40	609.6/24.0	129.33	1016.0/40.0	278.28

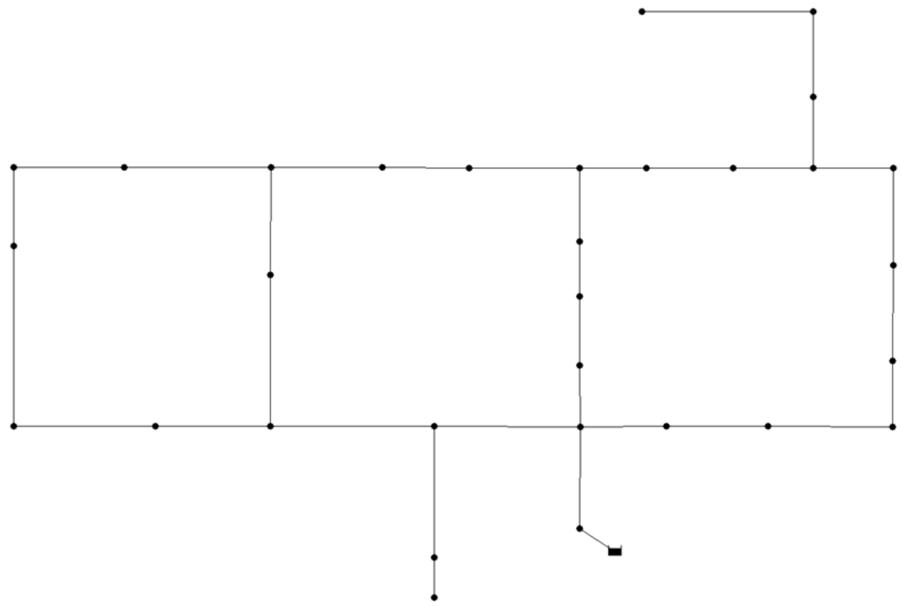


Figure 4.6 Layout of HAN

4.3.1.7. GoYang Network (GOY)

The GOY was initially presented by Kim et al. (1994) and it represents a water supply system in a city of South Korea.

The GOY has thirty pipes, twenty-two demand nodes, and one constant pump of 4.52 kW linking to one reservoir with a constant head of 71.0 m. The Hazen-Williams roughness coefficient for each new pipe is 100. The minimum pressure head above the ground elevation of each node is 15.0 m. There are eight commercially available pipe sizes, ranging from 80.0 mm to 350.0 mm. Thus, the search space is equal to $8^{30} \approx 1.24 \times 10^{27}$ discrete combinations. Table 4.9

shows the diameter options and the associated unit costs. Figure 4.7 depicts the layout of GOY.

Table 4.9 Diameter options and associated unit costs of GOY

Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)	Diameter (mm)	Unit Cost ($/m)
80.0	37.890	125.0	40.563	200.0	47.624	300.0	62.109
100.0	38.933	150.0	42.554	250.0	54.125	350.0	71.524

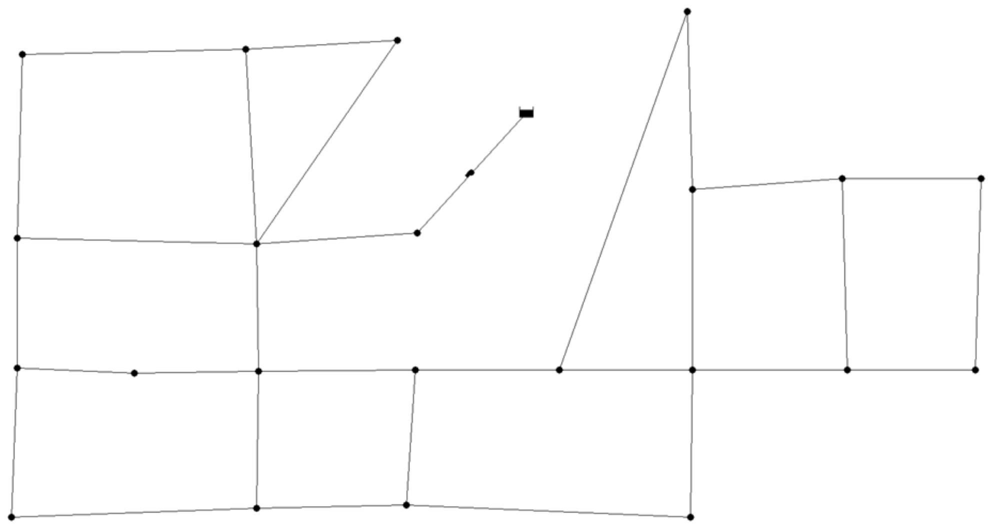


Figure 4.7 Layout of GOY

4.3.1.8. Fossolo Network (FOS)

The FOS was initially introduced by Bragalli et al. (2008) as a single neighbourhood of Bologna, called Fossolo in Northern Italy. Three instances were considered for this network. However, only one of them is adopted in this thesis as it represents the most complicated one in terms of the size of search space.

The FOS has fifty-eight pipes, thirty-six demand nodes, and one reservoir with a fixed head of 121.0 m. The material for all the pipes is polyethylene. Due to the characteristic of polyethylene, a relatively high roughness coefficient of 150 is applied to all the pipes. The minimum pressure head of all the demand nodes is maintained at 40.0 m, while the maximum pressure head of each node is specified in Table 4.10. In addition, the flow velocity in each pipe is kept at less than or equal to 1.0 m/s. There are twenty-two sizes in total to choose from. Hence, the search space is equal to $22^{58} \approx 7.25 \times 10^{77}$ discrete combinations. Table 4.11 shows the available diameter options and the associated unit costs. Figure 4.8 depicts the layout of FOS.

Table 4.10 Maximum pressure head requirement at each node of FOS

N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)
1	55.85	7	53.10	13	59.10	19	58.10	25	56.60	31	56.60
2	56.60	8	54.50	14	58.40	20	58.17	26	57.60	32	56.80
3	57.65	9	55.00	15	57.50	21	58.20	27	57.10	33	56.40
4	58.50	10	56.83	16	56.70	22	57.10	28	55.35	34	56.30
5	59.76	11	57.30	17	55.50	23	56.80	29	56.50	35	55.57
6	55.60	12	58.36	18	56.90	24	53.50	30	56.90	36	55.10

Note: N_I - node index which is a consecutive number starting from one to the total number of nodes in the network; H_{max} - maximum pressure head requirement.

4.3.1.9. Pescara Network (PES)

The PES was first presented by Bragalli et al. (2008) as a simplified real WDS in Pescara, a medium-size city in Italy.

The PES has ninety-nine pipes, sixty-eight demand nodes, and three reservoirs with fixed heads between 53.08 m and 57.00 m. The pipe material is cast iron. A uniform Hazen-Williams roughness coefficient of 130 is applied to all pipes. The minimum pressure head of all the demand nodes is maintained at 20.0 m, while the maximum pressure head of each node is specified in Table 4.12. In addition, the flow velocity in each pipe is kept at less than or equal to 2.0 m/s. There are 13 pipe sizes, ranging from 100.0 mm to 800.0 mm. Thus the search space is equal to $13^{99} \approx 1.91 \times 10^{110}$ discrete combinations. Table 4.13 shows the available diameter options and the associated unit costs. Figure 4.9 depicts the layout of PES.

Table 4.11 Diameter options and associated unit costs of FOS

Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)
16.00	0.38	61.40	4.44	147.20	24.78	290.60	99.58
20.40	0.56	73.60	6.45	163.60	30.55	327.40	126.48
26.00	0.88	90.00	9.59	184.00	38.71	368.20	160.29
32.60	1.35	102.20	11.98	204.60	47.63	409.20	197.71
40.80	2.02	114.60	14.93	229.20	59.70	-	-
51.40	3.21	130.80	19.61	257.80	75.61	-	-

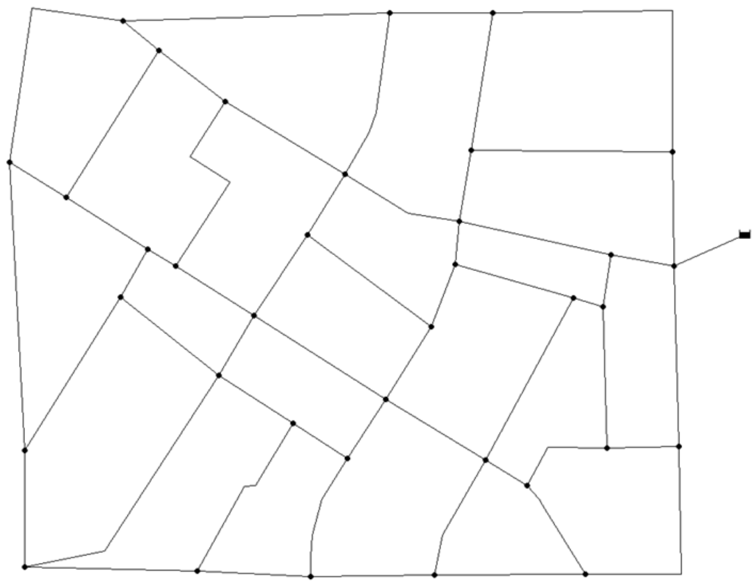


Figure 4.8 Layout of FOS

4.3.1.10. Modena Network (MOD)

The MOD was also introduced by Bragalli et al. (2008) as a simplified real WDS in Modena, a medium-sized city in Italy.

The MOD has 317 pipes, 268 demand nodes, and four reservoirs with fixed heads between 72.0 m and 74.5 m. The materials and sizes of commercially available pipes are the same as PES (see Table 4.13). A uniform Hazen-Williams roughness coefficient of 130 is applied to all pipes. The minimum pressure head of all the demand nodes is maintained at 20.0 m. Since the maximum pressure head at each node of MOD would take too much space, it is provided in Table A.1 (See Appendix A). In addition, the flow velocity in each pipe is kept at less than or equal to 2.0 m/s. There are thirteen pipe sizes (same as PES), so the search space is equal to $13^{317} \approx 1.32 \times 10^{353}$ discrete combinations. Figure 4.10 depicts the layout of MOD.

Table 4.12 Maximum pressure head requirement at each node of PES

N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)	N_I	H_{max} (m)
1	54.1	13	53.8	25	53.2	37	52.8	49	55.2	61	54.9
2	52.0	14	37.8	26	52.7	38	54.1	50	55.4	62	54.9
3	53.5	15	53.0	27	53.4	39	54.1	51	53.7	63	53.7
4	53.2	16	53.5	28	53.8	40	28.5	52	54.5	64	54.9
5	54.8	17	54.2	29	54.2	41	29.7	53	54.2	65	55.5
6	50.0	18	54.3	30	55.2	42	55.9	54	53.8	66	37.8
7	50.5	19	55.2	31	53.2	43	54.9	55	53.4	67	54.9
8	51.8	20	54.9	32	54.1	44	54.2	56	53.2	68	55.5
9	52.7	21	55.0	33	53.3	45	53.7	57	53.9	-	-
10	51.8	22	38.3	34	54.9	46	55.2	58	54.0	-	-
11	29.0	23	53.8	35	50.3	47	55.2	59	52.8	-	-
12	53.8	24	55.4	36	50.3	48	55.4	60	52.8	-	-

Note: N_I - node index which is a consecutive number starting from one to the total number of nodes in the network; H_{max} - maximum pressure head requirement.

Table 4.13 Diameter options and associated unit costs of PES

Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)
100.0	27.7	250.0	75.0	450.0	169.3	800.0	391.1
125.0	38.0	300.0	92.4	500.0	191.5	-	-
150.0	40.5	350.0	123.1	600.0	246.0	-	-
200.0	55.4	400.0	141.9	700.0	319.6	-	-

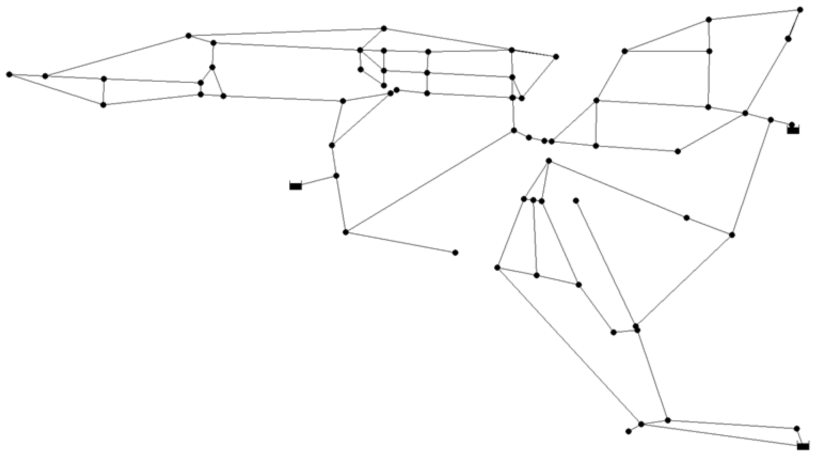


Figure 4.9 Layout of PES

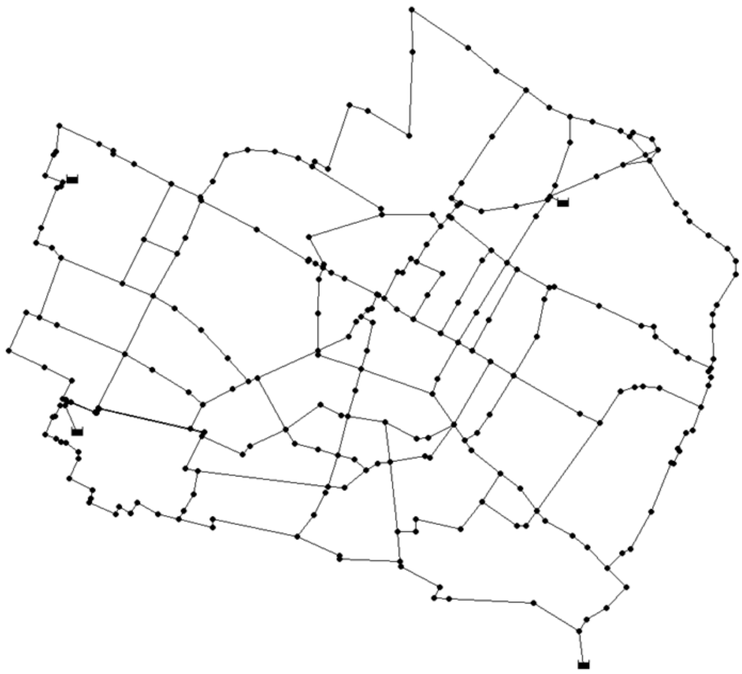


Figure 4.10 Layout of MOD

4.3.1.11. Balerma Irrigation Network (BIN)

The BIN was initially investigated by Reca and Martínez (2006) and it represents an adaptation of the existing irrigation system in the Sol-Poniente irrigation district, which is located in Balerma, province of Almería, Spain. The distinguishing feature of this network is that all nodes have the same demand of 5.55 l/s across the network.

The BIN has 454 relatively short pipes, 443 demand nodes (hydrants), and four reservoirs with fixed heads between 112.0 m and 127.0 m. The material of pipes is polyvinyl chloride (PVC). The Darcy-Weisbach roughness coefficient of 0.0025 mm is applied to all the pipes. The minimum pressure head above the ground elevation is 20.0 m for all the demand nodes. There are a total of ten commercially available sizes, ranging from 113.0 mm to 581.8 mm. Therefore, the search space is equal to10^{454}=$1.00\text{x}10^{455}$ discrete combinations. Table 4.14 shows the available diameter options and the associated unit costs. Figure 4.11 depicts the layout of BIN.

Figure 4.11 Layout of BIN

Table 4.14 Diameter options and associated unit costs of BIN

Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)	Diameter (mm)	Unit Cost (€/m)
113.0	7.22	162.8	14.84	285.0	45.39	581.8	215.85
126.6	9.10	180.8	18.38	361.8	76.32	-	-
144.6	11.92	226.2	28.60	452.2	124.64	-	-

4.3.1.12. Exeter Network (EXN)

The EXN was set up by the Centre for Water Systems (CWS) at the University of Exeter as a realistic and challenging duplication problem (Farmani et al. 2004). The network serves a population of approximately 400,000 and needs to be reinforced to meet the projected demand in 2020. It consists of a great number of small pipes and few transmission mains, with a large head-loss range at the extremities of the system. Originally, this model was built as an extended period simulation model. It is simplified here as per Farmani et al. (2005a) by replacing two tanks with fixed head reservoirs and by turning the EPS model into a steady-state model, considering the peak demand of system over twenty-four hours.

The EXN has 3,032 pipes including 567 pipes considered for duplication, five valves, 1,891 junction nodes and seven water sources. Two major reservoirs (node 3001 and 3002) supply water to the system at fixed heads of 58.4 m and 62.4 m respectively. The system is also fed by its neighbour systems via node 3003 to 3007 at fixed rates. Three non-return valves (also known as check valves) are connected to node 3001 and 3002 to control the flow direction into and out of the system. One pressure reducing valve is located in the downstream of node 3004 to maintain the downstream pressure below 58.4 m. One throttle control valve is also in the link downstream of node 3004 to control the flow and pressure of the system.

The minimum pressure requirement of demand nodes is 20.0 m. There are ten available discrete pipe sizes and one extra option of 'do nothing'. Therefore, the search space is equal to $11^{567} \approx 2.95 \times 10^{590}$ discrete combinations, which makes EXN the largest combinatorial problem analysed in this thesis. The unit cost for duplicating the existing pipe depends on both the diameter selected and the road type. Table 4.15 shows the pipe diameters, the corresponding roughness heights (following Darcy-Weisbach formula) and unit costs. The location of major roads is specified in Table 4.16 in terms of pipe ID. Figure 4.12 depicts the layout of EXN.

It is worth noting that the EXN network used in this thesis is a slightly different from the one which can be obtained from the following web address http://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/expansion/exnet.html. The modifications are summarised as follows: (1) Node 1610 has been removed to avoid isolated junction; (2) 20 PM in the section of “TIMES” of input file has been changed to 8 PM; and (3) Since it is impossible for junction 1107 to meet the minimum pressure requirement, it is ignored in the calculation of constraint violation. In addition, it would be too complicated to highlight all the pipes considered for duplication, these pipes are tagged in the input file of EPANET2 model.

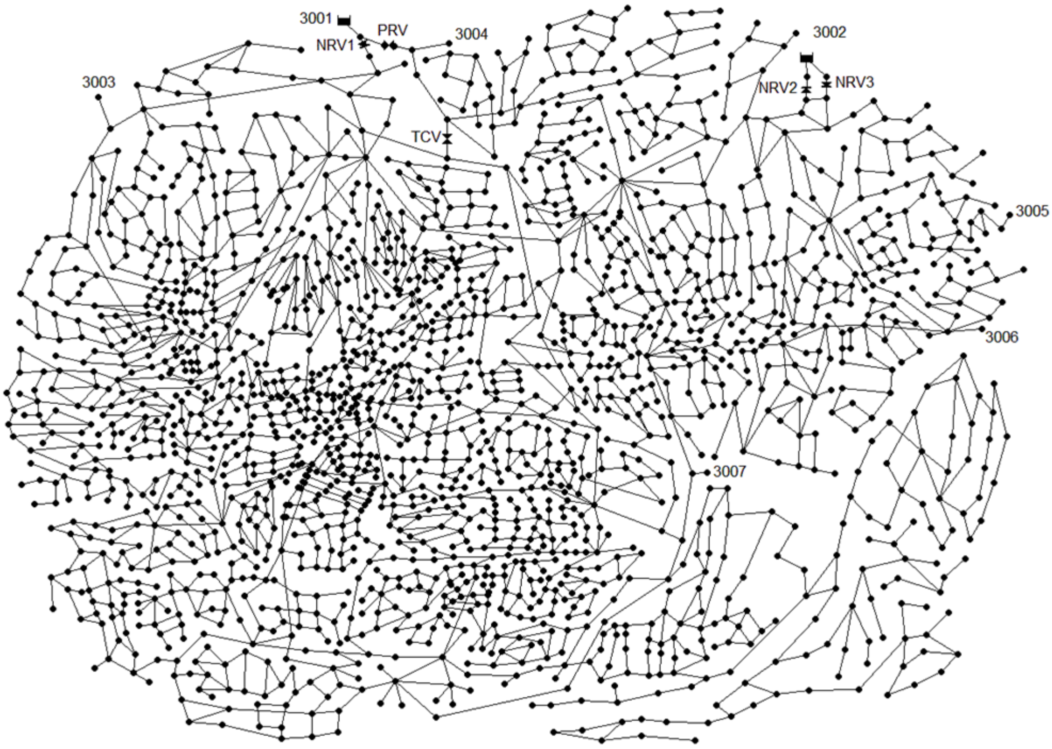


Figure 4.12 Layout of EXN

Table 4.15 Pipe intervention options, roughness coefficients and associated unit costs of EXN

Diameter (mm)	Roughness (mm)	Unit Cost (£/m)		Diameter (mm)	Roughness (mm)	Unit Cost (£/m)	
		Minor Road	Major Road			Minor Road	Major Road
110	0.03	85	100	400	0.23	250	290
159	0.065	95	120	500	0.3	310	340
200	0.1	115	140	600	0.35	370	410
250	0.13	150	190	750	0.43	450	500
300	0.17	200	240	900	0.5	580	625

Table 4.16 Location of major road in terms of pipe ID

Road No.	Pipe ID	Road No.	Pipe ID	Road No.	Pipe ID	Road No.	Pipe ID
1	5016	12	2681	23	5066	34	2065
2	4899	13	2803	24	2063	35	3917
3	2791	14	2784	25	5239	36	3937
4	3172	15	5085	26	5117	37	3823
5	3121	16	2579	27	3513	38	3792
6	3154	17	2651	28	2493	39	3834
7	3147	18	3055	29	2297	40	2963
8	2759	19	4057	30	5063	41	3690
9	2821	20	2790	31	5165	-	-
10	2752	21	2619	32	2369	-	-
11	2769	22	2087	33	4072	-	-

4.3.2. Classification and Features

Table 4.17 summarises all the benchmark networks used in this thesis. The details include the number of component elements, i.e., junction, reservoir, tank, pipe, pump and valve, as well as the formula of hydraulic head loss.

The benchmark networks in the WDSBA can be classified into different categories. For instance, according to the number of water sources and demand patterns, they can be categorised as single source or multiple sources network and single loading condition or multiple loading conditions, respectively. Another possibility is to classify the networks based on the type of hydraulic simulation, i.e., steady-state simulation or extended period simulation. However, all the networks in the WDSBA are steady-state models. Table 4.18 shows the classification of these networks from two aforementioned viewpoints.

Table 4.17 Number of each component of each benchmark network

Network	No. of Each Element						Formula
	Junction	Reservoir	Tank	Pipe	Pump	Valve	
TRN	10	2	0	17(8)	0	0	*H-W*
TLN	6	1	0	8	0	0	*H-W*
BAK	35	1	0	58(9)	0	0	*H-W*
NYT	19	1	0	42(21)	0	0	*H-W*
BLA	30	1	0	35(23)	0	0	*H-W*
HAN	31	1	0	34	0	0	*H-W*
GOY	22	1	0	30	1	0	*H-W*
FOS	36	1	0	58	0	0	*H-W*
PES	68	3	0	99	0	0	*H-W*
MOD	268	4	0	317	0	0	*H-W*
BIN	443	4	0	454	0	0	*D-W*
EXN	1891	2	0	3032(567)	0	2	*D-W*

Note: *H-W* - Hazen-Williams formula; *D-W* - Darcy-Weisbach formula. The number inside the parentheses denotes the number of pipes considered for optimisation. Otherwise, it means that all the pipes are considered for optimisation.

Table 4.18 Classification of each network used in this thesis

Network	*SS/MS*	*SL/ML*	Network	*SS/MS*	*SL/ML*
TRN	*MS(2)*	*ML(3)*	GOY	*SS*	*SL*
TLN	*SS*	*SL*	FOS	*SS*	*SL*
BAK	*SS*	*SL*	PES	*MS(3)*	*SL*
NYT	*SS*	*SL*	MOD	*MS(4)*	*SL*
BLA	*SS*	*SL*	BIN	*MS(4)*	*SL*
HAN	*SS*	*SL*	EXN	*MS(7)*	*SL*

Note: *SS* - single source; *MS* - multiple sources; *SL* - single loading condition; *ML* - multiple loading conditions. The number inside the parentheses after *MS* or *ML* denotes the number of sources or loading conditions respectively.

The features of each benchmark problem in the WDSBA are summarised in Table 4.19, including the number of decision variables and diameter options, the size of search space and design criteria. Note that these benchmark problems are categorised into four groups, ranging from small to large. The source code of benchmark problems in *C* language can be downloaded from www.exeter.ac.uk/benchmarks-pareto-fronts.

Table 4.19 Summary of the formulation of benchmark problems in WDSBA

Problem		No. of		Search Space	Design Criteria			
		DVs	Options		H^{min}	H^{max}	V^{max}	*SL*/*ML*
Small Problem	TRN	8	8*	3.28×10^{7}	Yes	No	No	*ML*
	TLN	8	14	1.48×10^{9}	Yes	No	No	*SL*
	BAK	9	11	2.36×10^{9}	Yes	No	No	*SL*
Medium Problem	NYT	21	16	1.93×10^{25}	Yes	No	No	*SL*
	BLA	23	15	2.30×10^{26}	Yes	Yes	Yes	*SL*
	HAN	34	6	2.87×10^{26}	Yes	No	No	*SL*
	GOY	30	8	1.24×10^{27}	Yes	No	No	*SL*
Intermediate Problem	FOS	58	22	7.25×10^{77}	Yes	Yes	Yes	*SL*
	PES	99	13	1.91×10^{110}	Yes	Yes	Yes	*SL*
Large Problem	MOD	317	13	1.32×10^{353}	Yes	Yes	Yes	*SL*
	BIN	454	10	1.00×10^{455}	Yes	No	No	*SL*
	EXN	567	11	2.95×10^{590}	Yes	No	No	*SL*

Note: *DV* - decision variables; H^{min} - minimum pressure head requirement; H^{max} - maximum pressure head requirement; V^{max} - maximum flow velocity; *SL* - single loading condition; *ML* - multiple loading conditions. *For the TRN problem, three of eight pipes are existing ones which have three options including 'do nothing', cleaning or duplication. Therefore, the number of design options of existing pipes is ten.

4.4. Summary

In this chapter, a Water Distribution System Benchmark Archive is set up based on twelve networks collected from the literature. A uniform problem formulation, i.e., two-objective optimal design or rehabilitation of WDSs, is applied to all the benchmark networks in the WDSBA. Then, each network is introduced following a uniform format in ascending order of search space. The main contribution of this chapter lies in providing a wide range of problems on which various optimisation methods can be benchmarked in a systematic way. A comparative study based on the WDSBA will give more insight into multi-objective optimisation using evolutionary algorithms.

5. EXPERIMENTS AND COMPARISONS ON BENCHMARK PROBLEMS

5.1. Introduction

This chapter applies the proposed GALAXY method and four other state-of-the-art MOEAs, including two representative hybrid algorithms and two baseline MOEAs, to solve the benchmark design or rehabilitation problems in the WDSBA.

In Section 5.2, the experiment is set up. Firstly, several suitable quantitative indicators are chosen from the literature for comparing the ultimate and dynamic performances of these algorithms. In addition, a tool based on an empirical attainment function (EAF), which was developed by López-Ibáñez et al. (2010), is employed to facilitate an intuitive comparison between the GALAXY method and its competitors in the objective space. Then, the computational budgets for various cases are justified using the baseline NSGA-II. Lastly, the individual parameters of each algorithm are determined.

In Section 5.3, results obtained by various algorithms are compared and discussed for the selected cases. The performances of different algorithms are evaluated by numerical indicators and the EAF tool from both ultimate and dynamic perspectives. In addition, the hybrid algorithms are assessed from the viewpoint of dynamic variations of their search operators, which facilitates a deep understanding of the internal evolution of each hybrid method.

In Section 5.4, this chapter is summarised and concluded.

5.2. Experimental Setup

This section describes the setup of experiments in which the GALAXY method is compared with four state-of-the-art MOEAs in the literature, from the perspectives of ultimate and dynamic performance. Among these four MOEAs, two of them are hybrid optimisation algorithms (i.e., AMALGAM and Borg as introduced in Chapter 2) and the others are baseline MOEAs including NSGA-II

(Deb et al. 2002) and ε-MOEA (Deb et al. 2005). The reasons for choosing these four MOEAs are as follows. NSGA-II has been widely used as a benchmark MOEA in water engineering (Farmani et al. 2005b; Kollat and Reed 2006; Raad et al. 2009), and it serves as the prototype of AMALGAM and GALAXY. ε-MOEA was introduced after NSGA-II featuring the ε-dominance concept, which is able to maintain the convergence and diversity simultaneously. ε-MOEA uses the same search operators as NSGA-II, i.e., SBX and PM, but is based on a steady-state algorithmic framework. Borg was developed based on ε-MOEA and more advanced strategies were involved, such as adaptive population sizing, randomised restart, and time continuation (Hadka and Reed 2013).

To evaluate the ultimate and dynamic performance of each algorithm, a combination use of numerical indicators and the EAF graphical tool are employed to facilitate an unbiased comparison and analysis. These MOEAs are applied to solve the benchmark problems in WDSBA, and the computational budget for each design problem is determined to ensure that all the algorithms converge properly. In addition, before implementing each algorithm, their individual parameters are set appropriately.

5.2.1. Performance Indicators

5.2.1.1. Generational Distance (I_{GD})

This indicator measures an average distance of the non-dominated solutions in the Pareto approximation front (denoted as *AF*) from those in the Pareto reference front (denoted as *RF*). More specifically, the Euclidean distance in the objective space from each solution in the *AF* to the closest one in the *RF* is considered (see Eq. (5.1)). I_{GD} purely measures the extent of convergence to the *RF*. Figure 5.1 illustrates the concept of I_{GD}. Without loss of generality, both objectives are assumed to be minimised hereinafter.

$$I_{GD} = \frac{\sum_{i=1}^{|AF|} \min(d_i^j)}{|AF|} \forall i \in AF, \forall j \in RF \tag{5.1}$$

Where, d_i^j - the Euclidean distance from solution *i* in the *AF* to the solution *j* in the *RF*; $|AF|$ - the cardinality of *AF*.

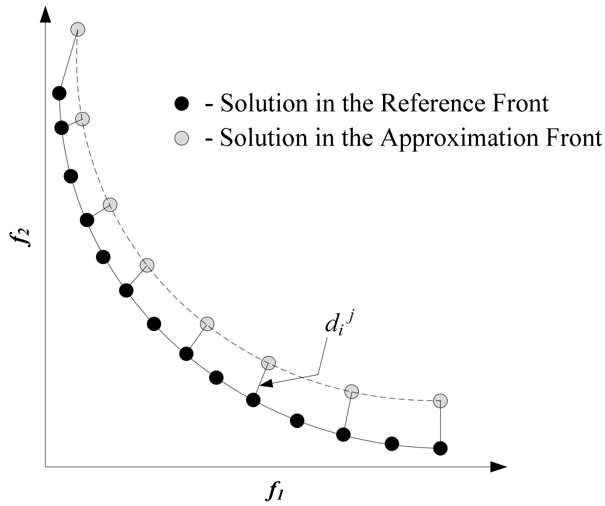


Figure 5.1 Illustration of concept of I_{GD}

5.2.1.2. Unary Hypervolume (I_{HV})

This metric calculates the ratio of the volume of hypercube (in the objective space) dominated by solutions in the *AF* to that by solutions in the *RF*. Each hypercube is constructed with a reference point (normally a vector of worst objective values) and a solution as the diagonal corners. I_{HV} represents a combined measure of convergence and diversity. A greater I_{HV} value indicates better performance. Figure 5.2 illustrates the concept of I_{HV}. The solutions in the *AF* are depicted as grey solid circles, and the solutions in the *RF* are depicted as dark solid circles. The grey area indicates the hypervolume dominated by the solutions in the *AF*; accordingly, the area encompassed by the dark solid circles and the reference point indicates the hypervolume dominated by the solutions in the *RF*.

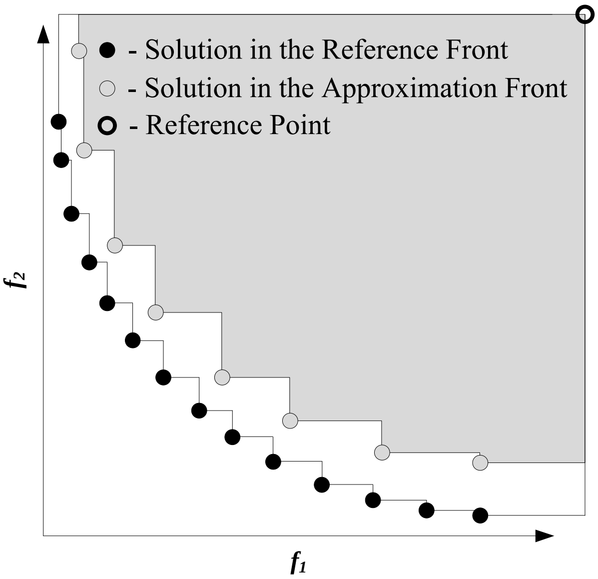


Figure 5.2 Illustration of concept of I_{HV}

5.2.1.3. Unary Additive ε -Indicator ($I_{\varepsilon+}$)

This metric seeks the minimum distance that the *AF* must be translated in order to completely dominate the entire *RF*. That is, any solution in the *RF* is ε-dominated by at least one solution in the *AF*. Unlike I_{HV} which tends to be insensitive to the gaps between consecutive solutions, $I_{\varepsilon+}$ is able to identify the objective vectors with poor proximity. As illustrated in the figure below, the dashed, empty circles represent the missing points in the *AF*. Therefore, $I_{\varepsilon+}$ can easily capture the gap between *AF* and *RF*, indicating the inconsistency (uniform convergence to the *RF*) of the solutions in the *AF.* Hadka and Reed (2012) asserted that $I_{\varepsilon+}$ can report a metric two to three times worse than I_{GD} and I_{HV} in such a case. In other words, it represents not only the convergence of the *AF*, but also the consistency of the non-dominated solutions found, which is another desired feature of an MOEA. Figure 5.3 illustrates the concept of $I_{\varepsilon+}$. Given an *AF* and an *RF*, the $I_{\varepsilon+}$ can be computed following the Eq. 5.2-5.4 (adapted from (Zitzler et al., 2003)).

$$\varepsilon_{z^1,z^2} = \max_{1 \le i \le n}(\left| z_i^1 - z_i^2 \right|) \quad \forall z^1 \in AF, z^2 \in RF \tag{5.2}$$

$$\varepsilon_{z^2} = \min_{z^1 \in AF}(\varepsilon_{z^1,z^2}) \quad \forall z^2 \in RF \tag{5.3}$$

$$I_{\varepsilon+} = \max_{z^2 \in RF}(\varepsilon_{z^2}) \tag{5.4}$$

Where, *n* - number of objectives; ε_{z^1,z^2} - the maximum differences among *n* objectives of two selected solutions, one from the *AF* and the other from the *RF*; ε_{z^2} - the maximum differences among *n* objectives of the nearest solution in the *AF* to a given solution z^2 in the *RF*.

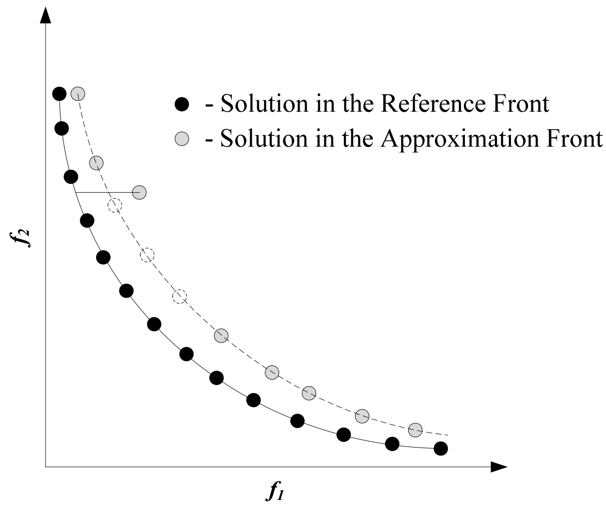


Figure 5.3 Illustration of concept of $I_{\varepsilon+}$ (adapted from (Hadka and Reed 2012))

5.2.1.4. ε-Performance (I_{EP})

As both Borg and ε-MOEA were created based on the concept of ε-dominance concept (Laumanns et al., 2002), the number of ε-non-dominated solutions in the final archive is not necessarily equal to the population size. In other words, the reported best solutions have been filtered using a grid with user-specified ε precisions. When comparing them with those found by non-ε-dominance based algorithms (i.e., AMALGAM and NSGA-II), the calculation of aforementioned indicators (such as I_{HV}) may be affected and result in a biased interpretation. Therefore, the ε-performance metric is also included to alleviate the issue.

This metric accounts for the proportion of solutions that are discovered within a user-specified ε distance from the reference set (Kollat and Reed 2005). Firstly, a series of boxes are built with side lengths equal to ε_i for the *i*-th objective and each solution in the *RF* being located in the centre of the box. Then, the number of solutions in the approximation set that fall within the boxes is counted. Note that only the closest one (in terms of Euclidean distance) in the approximation set is considered if multiple solutions match the criterion. Finally, the ratio of this number to the cardinality of the reference set represents the ε-performance of this approximation set. In principle, this indicator measures the quality of non-dominated solutions in terms of convergence in an absolute sense. Figure 5.4 provides an illustration of calculating I_{EP}.

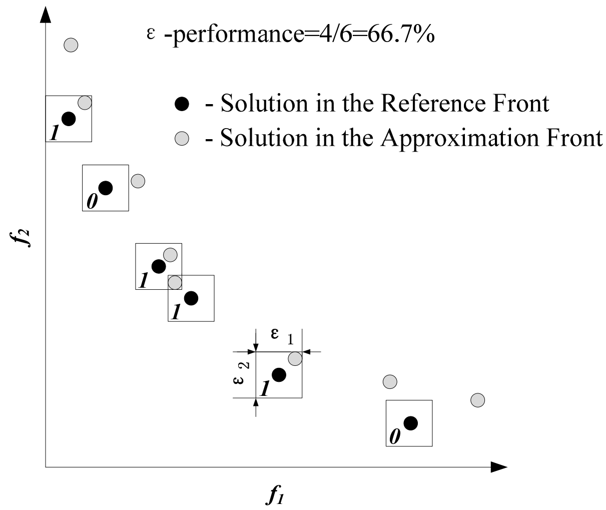


Figure 5.4 Illustration of calculation of I_{EP} (adapted from (Kollat and Reed 2005))

5.2.1.5. Normalisation of Performance Indicators

To deal with a wide variety of values measured by different performance indicators, all the aforementioned indicators are normalised to reside within the range [0, 1] if necessary, with 1 representing the ideal value. Thus, I_{GD} and $I_{\varepsilon+}$ are normalised by using Eq. (5.5) and Eq. (5.6), respectively. The ranges of I_{HV} and I_{EP} are intrinsically between 0 and 1. On the other hand, the qualified solutions (in the *AF*) considered for calculating the aforementioned metrics must be feasible and belong to the first rank. In other words, the inferior and/or infeasible solutions are omitted from the evaluation.

$$norm(I_{GD}) = \max(1 - I_{GD}, 0) \tag{5.5}$$

$$norm(I_{\varepsilon+}) = \max(1 - I_{\varepsilon+}, 0) \tag{5.6}$$

To compute all the aforementioned performance indicators, a reference set is required for each benchmark problem as the best approximation to the Pareto-optimal front. To this end, five state-of-the-art MOEAs with their recommended parameter settings were applied to solve the benchmark problems in the WDSBA, given extensive computational budgets depending on the size of search space (Wang et al. 2014a). Three different population sizes were implemented for each MOEA with respect to the scale of each problem. The true Pareto-optimal fronts for small problems were achieved via full enumeration within the solution space; whereas the best-known Pareto fronts for the other problems were obtained by filtering the aggregated approximation sets found by all the methods.

5.2.1.6. EAF Tool

The EAF based graphical approach (López-Ibáñez et al. 2010) is employed as it provides a visual and intuitive way to compare different algorithms for two-objective optimisation. The EAF can describe the probabilistic distribution (in the objective space) of the outcomes obtained by an algorithm via multiple runs. Moreover, it can provide useful information, such as where and how much one algorithm differs from another one. Such an example is provided in Figure 5.5, assuming that both objectives are to be minimised. Generally speaking, by combining the solutions finally obtained, the performances of two algorithms are plotted side by side and four types of information are demonstrated: (1) best attainment surface obtained by both algorithms (lower solid lines); (2) worst attainment surface obtained by both algorithms (upper solid lines); (3) median attainment surface of each algorithm (dashed lines); and (4) differences of EAFs between two algorithms. Such differences are encoded using discrete levels of grey colour: the darker an area is plotted, the larger the difference exists there.

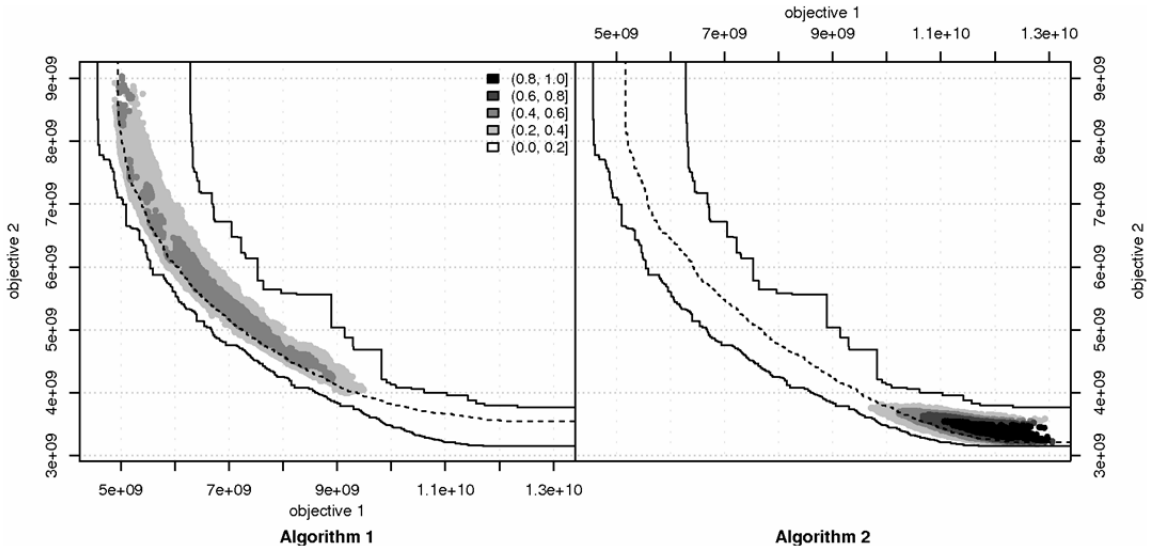


Figure 5.5 An example of EAF plot (adopted from (López-Ibáñez et al. 2010))

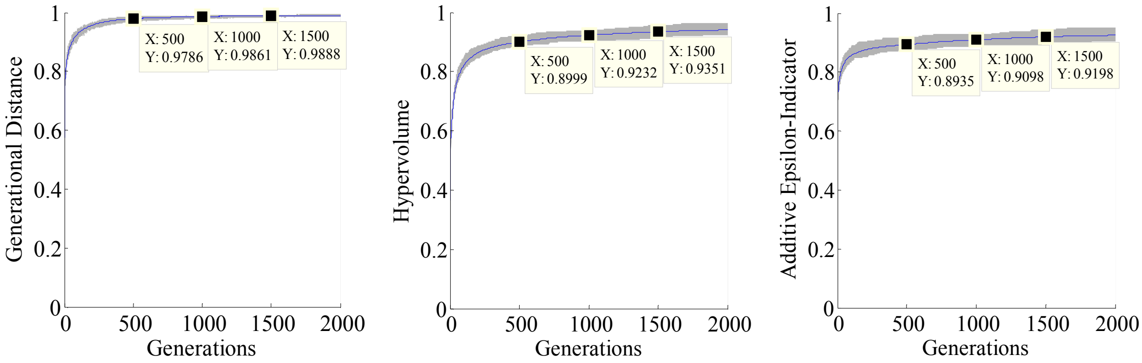
5.2.2. Computational Budget

As one of the most popular algorithms in the second generation MOEAs (Coello 2006), the NSGA-II has been broadly used as the baseline algorithm in comparative studies. In order to determine what a sufficient computational budget in terms of the number of function evaluations (NFEs) is, the NSGA-II was applied extensively to solve some representative benchmark problems in Chapter 4 with different random seeds. Note that not all the algorithms were considered to determine the suitable computational budgets for various benchmark problems. This is because that the others algorithms were developed after NSGA-II and they were proved to achieve better performance on a wide range of test problems. The aforementioned indicators were monitored to assess the dynamic performance of NSGA-II, thus picking a suitable budget which is sufficient to ensure the convergence of NSGA-II. In other words, a budget was determined such that the improvement of the NSGA-II's convergence was less than a threshold (0.5% herein), implying that more computation was not necessary. Regarding the selection of benchmark problems, they were first categorised into four groups (from small to large networks) according to the size of search space, which is believed to be the major source of difficulty. Then, the problems with the largest search space in the first three groups (excluding the large network group) were solved by the

NSGA-II. Note that, the budget was not verified for the last group because the computational overhead was too high for the available computing resources. The trial runs and data post-processing for large networks would take exceptionally long time and massive disk space due to the duration of simulation and the analysis of dynamic performance, respectively.

Given above, the BAK, GOY and PES were selected as the representative benchmark problems, and each was solved thirty times independently by the NSGA-II. A population size of 100 was kept and the number of generations was set to 2,000, which was equivalent to 200,000 NFEs. Through some trial runs, it was found that the distribution indices of SBX and PM had a significant impact on the performance of the real-coded NSGA-II. Thus, these indices were kept to be the same for SBX and PM and various different values were tested to identify the best index. Using the same procedure as mentioned previously, an index of 1 for SBX and PM turned out to return the best approximation sets on average according to various performance indicators (see Appendix C). Therefore, this value was adopted throughout the thesis. In addition, the crossover rate and the mutation rate were fixed at 0.9 and the inverse of the number of decision variables respectively, because these are the most recommended values in the literature (Deb et al. 2005; Vrugt and Robinson 2007).

Figure 5.6 illustrates the dynamic performance of the NSGA-II (both indices equal to one) using the full budget (i.e., after 2,000 generations) for the PES problem. Three intermediate points, i.e., at generation=500, 1,000, 1,500, are annotated to show the mean value of a specific metric over 30 runs. These values are then compared to the final achievement using the full budget. As indicated in Table 5.1, a proper computational budget is determined by confirming that no significant improvement in terms of I_{GD} (convergence indicator) can be achieved subsequently. The threshold here is chosen to be 0.5%, which is believed to be small enough from a practical perspective. The figures for the other two problems (i.e., BAK and GOY) are not shown here for brevity (see Appendix C).



(a) I_{GD} (b) I_{HV} (c) $I_{\varepsilon+}$

Figure 5.6 Dynamic performance of the NSGA-II for the PES problem

Table 5.1 Convergence test of the NSGA-II on benchmark problems

Problem	Gen	Performance Indicator						Improvement (%)		
		Low Budget			Full Budget					
		I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{GD}	I_{HV}	$I_{\varepsilon+}$
BAK	250	0.9999	0.9943	0.9974	0.9999	0.9991	0.9982	0.0	0.5	0.1
GOY	500	0.9964	0.9588	0.9560	0.9974	0.9659	0.9612	0.1	0.7	0.5
PES	1,000	0.9861	0.9232	0.9098	0.9900	0.9412	0.9253	0.4	1.9	1.7

Note: Gen - number of generations.

As mentioned in Chapter 2, I_{GD}, I_{HV} and $I_{\varepsilon+}$ can be used to evaluate different aspects of an MOEA for a particular problem. I_{GD} focuses only on the convergence aspect while I_{HV} is able to measure both convergence and diversity in a combined sense. $I_{\varepsilon+}$ can not only reflect the proximity of an approximation set but also implicitly assess the consistency (gaps between consecutive solutions) of an approximation set towards the Pareto-optimal front.

Note that the improvements in I_{GD} (see Table 5.1) for the three problems are less than 0.5%, indicating that the NSGA-II has converged after the selected generations. A relatively large improvement in terms of I_{HV} and $I_{\varepsilon+}$ can be observed for the PES problem mainly due to the partial order of the *Selection* and *Replacement* procedures in the NSGA-II. That means convergence is of higher priority compared with diversity within the non-dominated sorting. So, the NSGA-II tries to improve the diversity after convergence.

In short, it can be asserted that 25,000 and 50,000 NFEs with a population of 100 are enough for the small and medium sized problems, respectively. A total number of 100,000 NFEs with a population of 100 are required for the intermediate sized problems. In contrast, for large problems, the population size and the number of generations are doubled compared to the budget for the intermediate sized problems. This is equivalent to 400,000 NFEs in order to cope with the huge search space. Note that there is no guarantee that any algorithm can converge given this budget. However, an algorithm that can perform well with limited NFEs is preferred when dealing with real-world design problems, for which high computational budget may not be affordable.

In addition, thirty independent runs are implemented for each benchmark problems using different random seeds. Here, it is worth mentioning that for each run the initial population of different algorithms are kept identical, thereby making them start the evolution from the same position in search space. This treatment is aimed at eliminating the impact of the quality of the initial population on the algorithm’s achievement.

5.2.3. Algorithmic Setup

It is well established that MOEAs suffer from the parameterisation issue (Kollat and Reed 2006; Hadka and Reed 2012). In other words, parameter settings play a crucial role in the performance of MOEAs considered. This issue can be even more complex when employing hybrid algorithms (e.g., AMALGAM and Borg) because they normally introduce additional parameters to be specified before application. As a result, it is of great importance to properly choose each parameter in a comparative study. In this chapter, these parameters are divided into two groups, i.e., general parameters and specific parameters. General parameters include those that are common for various algorithms, like the population size (N) and the number of function evaluations (NFEs); whereas the specific parameters involve those only employed by algorithms themselves. For general parameters, they are kept the same across different algorithms for various benchmark problems; but for specific parameters, they are determined based on a careful literature survey as well as by conducting a sensitivity analysis.

Since the GALAXY method actually involves only one specific parameter, namely the range of the probability of dither creeping mutation, it is determined through the sensitivity analysis given in Section 3.3 in Chapter 3. More precisely, this range is set to be 0.7; note that DC is not sensitive to the setting of this parameter. In addition, the probability of three mutation operators (i.e., UM, GM and DC) is set according to the most recommended value in the literature, which is equal to the inverse of the number of decision variables (i.e., 1/*ND*). Therefore, it is necessary to highlight that the GALAXY method stands out from the other hybrid methods (i.e., AMALGAM and Borg) as it is dedicated for solving discrete problems and effectively eliminates all the specific parameters of search operators. This is regarded as a desired feature of GALAXY by providing a very user-friendly tool for the two-objective optimal design or rehabilitation of WDSs.

In contrast, the AMALGAM and Borg involve 11 and 15 individual parameters (see Table 5.2) respectively, which can greatly impact on their performance. In this thesis, the individual parameters of Borg and AMALGAM are not fine-tuned for two main reasons. Firstly, Borg features adaptive population sizing and “time continuation” strategy (involving several connected runs triggered by automatic restart). In fact, one of the main advantages of Borg is to eliminate the need for parameterisation, resulting in a highly reliable and efficient MOEA. Secondly, the primary control parameters in the AMALGAM (see Table 3.1 in Chapter 3) are not fixed by default. Instead, these parameters are randomly sampled from the high-performance ranges recommended in relevant papers (Parsopoulos and Vrahatis 2002; Gelman et al. 2003; Hu et al. 2003; Iorio and Li 2005). Hence, AMALGAM is also expected to reduce the issue of parameterisation to some extent. As shown in Reed (2013), Borg and AMALGAM were demonstrated to be the top performing methods among 10 benchmark MOEAs for a suite of water resources applications with real-valued and discrete variables (including rainfall-runoff calibration, long-term groundwater monitoring and risk-based water supply portfolio planning). The control maps for these applications also highlighted the relatively large areas of “sweet spots” (Goldberg 2002) of Borg and AMALGAM with regard to their parameterisation, indicating that they are expected to return satisfactory results under the default settings.

Table 5.2 Parameter settings for each algorithm

<table>
<tr><th rowspan="2">Algorithm</th><th colspan="2">General Parameter (problem-dependent)</th><th rowspan="2" colspan="2">Specific Parameters</th></tr>
<tr><th>NFE</th><th>N</th></tr>
<tr><td>GALAXY</td><td rowspan="12">25,000-
400,000</td><td rowspan="2">100-200</td><td colspan="2">probability of mutation: 1/ND</td></tr>
<tr><td>AMALGAM</td><td colspan="2">see Table 3.1 in Chapter 3</td></tr>
<tr><td rowspan="7">Borg</td><td rowspan="7">100-200;
initial popSize: N
min popSize: N
max popSize: 10,000
population ratio: 4
selection ratio: 0.02</td><td>SBX</td><td>rate: 0.9
distribution index: 1</td></tr>
<tr><td>DE</td><td>crossover rate: 0.6
step size: 0.6</td></tr>
<tr><td>UM</td><td>rate: 1/ND</td></tr>
<tr><td>SPX</td><td>No. of parents: 10
expansion rate: $\sqrt{10+1}$</td></tr>
<tr><td>PCX</td><td>No. of parents: 10
Eta: 0.1
Zeta: 0.1</td></tr>
<tr><td>UNDX</td><td>No. of parents: 10
Zeta: 0.5
Eta: 0.35/$\sqrt{ND}$</td></tr>
<tr><td>PM</td><td>rate: 1/ND
distribution index: 1</td></tr>
<tr><td rowspan="2">NSGA-II</td><td rowspan="3">100-200</td><td>SBX</td><td>rate: 0.9
distribution index: 1</td></tr>
<tr><td>PM</td><td>rate: 1/ND
distribution index: 1</td></tr>
<tr><td>ε-MOEA</td><td colspan="2">same as NSGA-II</td></tr>
</table>

Note: popSize - population size (*N*). DEs in GALAXY, AMALGAM and Borg are mutually different variants. In addition, PM in Borg is not employed individually as in the others. Instead, offspring produced by SBX, DE, SPX, PCX and UNDX are subsequently mutated using PM.

On the other hand, as discussed in Section 5.2.2, the distribution indices of SBX and PM have a major impact on their search ability. So, the sensitivity analysis base on these indices was carried out to find the best settings. Since all the algorithms except the GALAXY method employ SBX and PM as search operators, only NSGA-II was implemented in the sensitivity analysis, and the best settings (both equal to one) are applied to these MOEAs. The complete specifications of individual parameters of each method are provided in Table 5.2.

In addition, as Borg and ε -MOEA both require user-defined ε_i precision for the *i*-th objective, they are determined by a trial-and-error approach and provided in Table 5.3. It is worth mentioning that the precision settings provided in the table below for each benchmark problem is also used to calculate the I_{EP}.

Table 5.3 ε precision settings for each benchmark problem

Benchmark Problem	1st Objective: *Cost* Minimisation (in Millions Unit)			2nd Objective: I_n Maximisation (-)		
	$Cost^{best}$	$Cost^{worst}$	ε_{cost}	I_n^{worst}	I_n^{best}	ε_{In}
TRN	0.399	5.525	0.01	0	1	0.001
TLN	0.016	4.400	0.01	0	1	0.001
BAK	0.378	1.389	0.001	0	1	0.001
NYT	0	294.156	1	0	1	0.001
BLA	0.058	1.299	0.001	0	1	0.001
HAN	1.803	10.970	0.1	0	1	0.001
GOY	0.175	0.330	0.001	0	1	0.001
FOS	0.003	1.662	0.001	0	1	0.001
PES	1.346	19.004	0.01	0	1	0.001
MOD	1.989	28.083	0.01	0	1	0.001
BIN	0.724	21.642	0.01	0	1	0.001
EXN	0	97.501	0.01	0	1	0.001

Note: The $Cost^{best}$ and $Cost^{worst}$ for each case are obtained by setting the decision variables (i.e., pipe diameters) to the minimum and maximum commercially available sizes, respectively. For NYT and EXN cases, the $Cost^{best}$ is 0 because these are rehabilitation problems, including “do-nothing” option for each pipe. I_n^{worst} and I_n^{best} are ideal cases, which are determined by the definition of I_n.

5.3. Results and Discussion

In this section, the ultimate and dynamic performances of the GALAXY method are compared with those obtained by its competitors (i.e., two hybrid algorithms and two baseline MOEAs). For the sake of brevity, four case studies in the WDSBA are selected for comparison purposes. Recall that the benchmark problems in the WDSBA are divided into four groups according to the size of search space; therefore, four representative cases are chosen from each of these groups to verify the performance of the GALAXY method. More specifically, BAK, HAN, PES, and EXN problems are examined in this section and the reasons are twofold: (1) BAK, PES and EXN represent the cases with the largest search space in the small, intermediate and large groups, respectively; and (2) HAN in the medium group has an extremely small feasible region which makes it difficult to locate the PF in a limited *NFE*.

Here, it is necessary to clarify that the solutions used to compute the performance indicators (either ultimate or dynamic) are those with the highest rank in the population. In particular, for the Pareto-dominance based MOEAs (i.e., GALAXY, AMALGAM and NSGA-II), the solutions in the first front are chosen for computing various indicators; while for the ε -dominance based MOEAs (i.e., Borg and ε -MOEA) the solutions in the archive are selected.

5.3.1. Ultimate Performance

The results obtained by five algorithms for the four benchmark problems are shown in Table 5.4-5.7. A statistical analysis is provided across thirty independent runs for each performance indicator. The average, maximum, minimum and standard deviation of each indicator are compared among algorithms and the best value is shown in bold. In addition, the comparisons between GALAXY and the other MOEAs using the EAF tool are illustrated in Figure 5.7-5.10.

For the BAK problem, all the algorithms were able to achieve a similar level of achievement in terms of I_{GD}, I_{HV}, and $I_{\varepsilon+}$, as revealed in Table 5.4. As it can be seen from this table, GALAXY slightly dominated the other MOEAs by attaining the majority of the best values in the statistical analysis. Moreover, GALAXY did outperform the others in terms of I_{EP} by covering nearly 75% of the true PF on average. In contrast, the other MOEAs were only able to identify about 60% of the true PF. This superiority of GALAXY can be attributed to the duplicates handling strategy and the hybrid replacement strategy, which ensures that the global best non-dominated solutions in between the boundary solutions are never been discarded.

Table 5.4 Ultimate performance of various MOEAs for the BAK problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	**1.0000**	0.9999	**1.0000**	0.9999	**1.0000**
	Max.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	Min.	**1.0000**	0.9999	0.9999	0.9998	0.9996
	Std.	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0001
I_{HV}	Avg.	**0.9953**	0.9943	0.9933	0.9943	0.9915
	Max.	**0.9999**	0.9998	0.9998	0.9997	0.9971
	Min.	**0.9931**	0.9900	0.9901	0.9901	0.9869
	Std.	0.0032	0.0036	**0.0024**	0.0034	0.0026
$I_{\varepsilon+}$	Avg.	**0.9977**	0.9973	0.9973	0.9974	0.9954
	Max.	0.9985	0.9987	**0.9990**	0.9986	0.9973
	Min.	**0.9973**	0.9961	0.9961	0.9961	0.9932
	Std.	**0.0005**	0.0007	0.0006	0.0007	0.0017
I_{EP}	Avg.	**0.7408**	0.5930	0.6225	0.5916	0.6046
	Max.	**0.7482**	0.6547	0.6403	0.6331	0.6331
	Min.	**0.7338**	0.5468	0.5971	0.5252	0.5827
	Std.	**0.0040**	0.0269	0.0075	0.0237	0.0161

From the comparisons of EAF (see Figure 5.7), all the methods except the ε-MOEA were able to achieve very close and stable Pareto fronts in multiple runs. The ε-MOEA failed to locate the solutions in the region of high network resilience, which was probably due to the usage of ε-dominance concept. In contrast, Borg was able to overcome this deficiency via more advanced techniques, e.g., multi-operator search and adaptive population sizing.

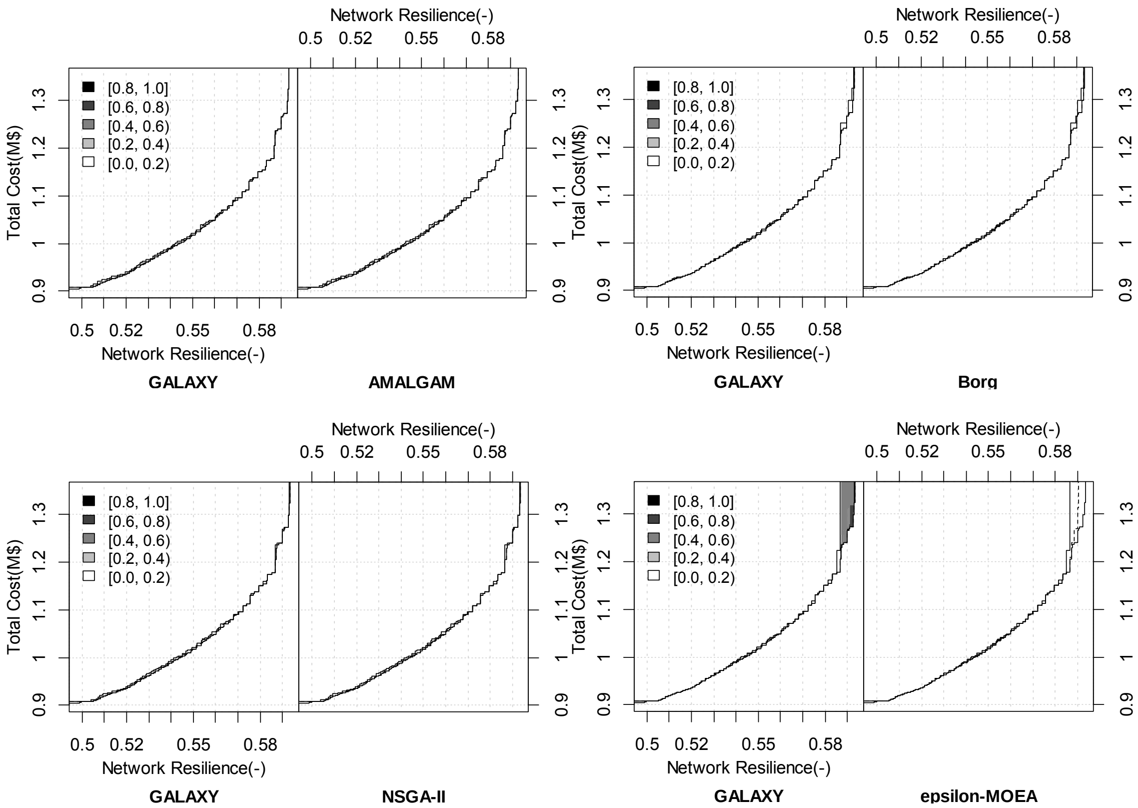


Figure 5.7 Comparison of GALAXY with the other MOEAs for the BAK problem using the EAF tool

For the HAN problem, again, GALAXY dominated the other MOEAs by achieving most of the best values in the statistical analysis. It can be seen from Table 5.5 that all the algorithms converged very well (see I_{GD}), but actually their performances were quite different according to the other indicators. In particular, only GALAXY and NSGA-II were able to obtain the I_{HV} and $I_{\varepsilon+}$ larger than 0.96 and 0.98 respectively on averages. GALAXY was superior to the others in terms of I_{EP} by covering nearly 39% of the best-known PF.

From the comparisons of EAF (see Figure 5.8), it is clear that GALAXY could consistently find better solutions in the boundary areas, which in turn contributed better results in terms of I_{HV} and $I_{\varepsilon+}$. Both the ε-dominance based MOEAs found difficulties in discovering the boundary solutions, which was probably caused by removing non-dominated solutions which were ε-dominated, thus losing useful genes which were indispensable for finding boundary solutions.

Table 5.5 Ultimate performance of various MOEAs for the HAN problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	**0.9989**	0.9984	0.9987	0.9986	0.9988
	Max.	**0.9996**	0.9995	**0.9996**	0.9994	0.9995
	Min.	0.9968	0.9966	0.9967	0.9970	**0.9977**
	Std.	0.0006	0.0006	0.0006	0.0007	**0.0005**
I_{HV}	Avg.	**0.9649**	0.9245	0.9215	0.9620	0.9184
	Max.	**0.9878**	0.9702	0.9694	0.9870	0.9432
	Min.	**0.9400**	0.8751	0.8953	0.9259	0.8778
	Std.	**0.0157**	0.0250	0.0165	0.0170	**0.0157**
$I_{\varepsilon+}$	Avg.	0.9800	0.9593	0.9595	**0.9809**	0.9603
	Max.	0.9927	0.9851	0.9892	**0.9936**	0.9753
	Min.	**0.9652**	0.9324	0.9452	0.9585	0.9370
	Std.	**0.0092**	0.0136	0.0096	0.0098	**0.0092**
I_{EP}	Avg.	**0.3879**	0.2401	0.1886	0.2737	0.1827
	Max.	**0.4870**	0.3670	0.2643	0.4574	0.2261
	Min.	**0.2835**	0.1235	0.0939	0.1130	0.1322
	Std.	0.0590	0.0611	0.0474	0.0936	**0.0262**

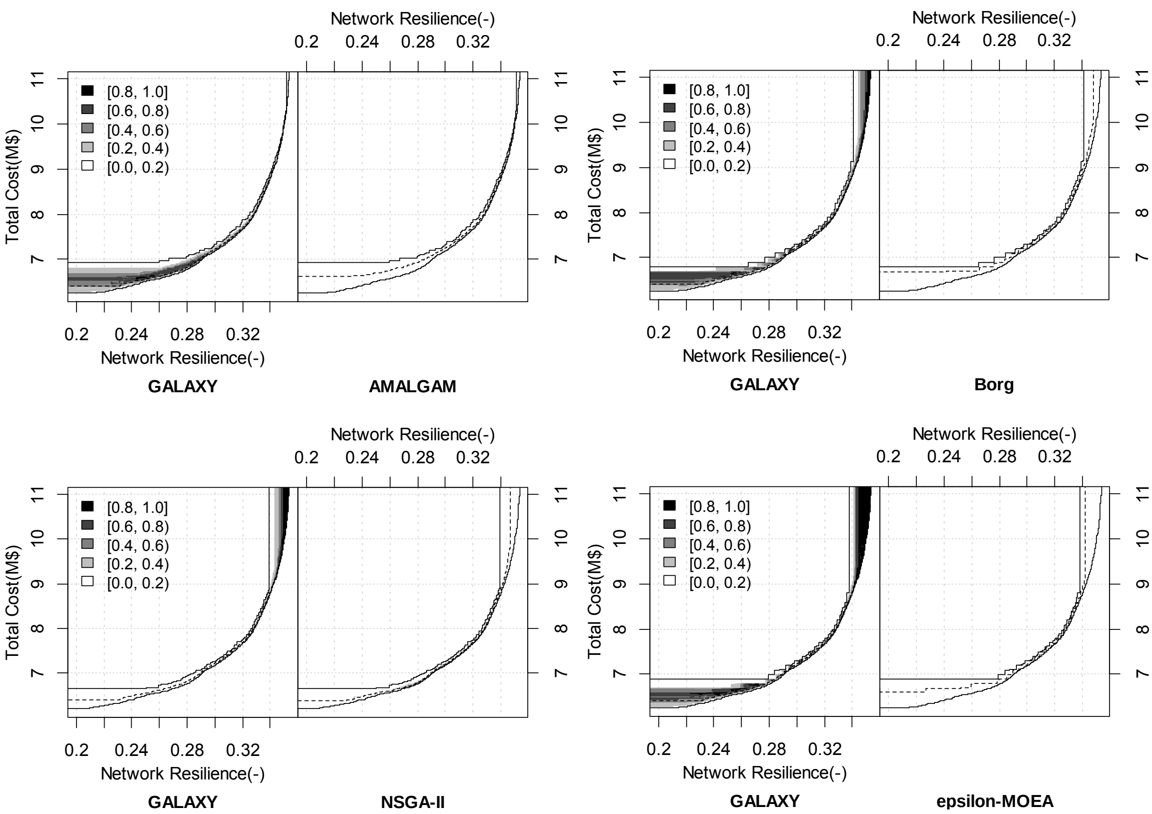


Figure 5.8 Comparison of GALAXY with the other MOEAs for the HAN problem using the EAF tool

For the PES problem, a similar observation to the HAN problem was obtained in Table 5.6. ε-MOEA demonstrated the best overall convergence compared with the other MOEAs but the performances in terms of the other indicators were much worse. In contrast, GALAXY consistently performed very well in terms of all the indicators. In addition, it is worth highlighting its merit in terms of I_{EP} that all the other MOEAs completely failed to approach any non-dominated solutions on the best-known PF.

Table 5.6 Ultimate performance of various MOEAs for the PES problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	0.9859	0.9610	0.9699	0.9860	**0.9881**
	Max.	0.9901	0.9835	0.9803	0.9900	**0.9942**
	Min.	0.9765	0.9506	0.9584	**0.9791**	0.9614
	Std.	**0.0031**	0.0063	0.0057	**0.0031**	0.0058
I_{HV}	Avg.	**0.9661**	0.9342	0.8795	0.9205	0.8958
	Max.	**0.9759**	0.9593	0.8949	0.9360	0.9106
	Min.	**0.9546**	0.9139	0.8600	0.8996	0.8743
	Std.	**0.0049**	0.0115	0.0080	0.0084	0.0082
$I_{\varepsilon+}$	Avg.	**0.9532**	0.9517	0.8869	0.9098	0.8885
	Max.	0.9656	**0.9658**	0.9090	0.9297	0.9026
	Min.	**0.9407**	0.9107	0.8585	0.8850	0.8629
	Std.	**0.0071**	0.0103	0.0113	0.0101	0.0092
I_{EP}	Avg.	**0.0059**	0.0000	0.0000	0.0000	0.0000
	Max.	**0.0224**	0.0000	0.0000	0.0000	0.0000
	Min.	**0.0000**	0.0000	0.0000	0.0000	0.0000
	Std.	**0.0092**	0.0000	0.0000	0.0000	0.0000

From the comparisons of EAF (see Figure 5.9), it can be clearly seen that GALAXY was capable of discovering boundary solutions in regions of high and low network resilience, which the other MOEAs were unable to reach. However, Borg, NSGA-II and ε-MOEA successfully identified solution of better quality than GALAXY around the ‘knee point’ of the best-known PF.

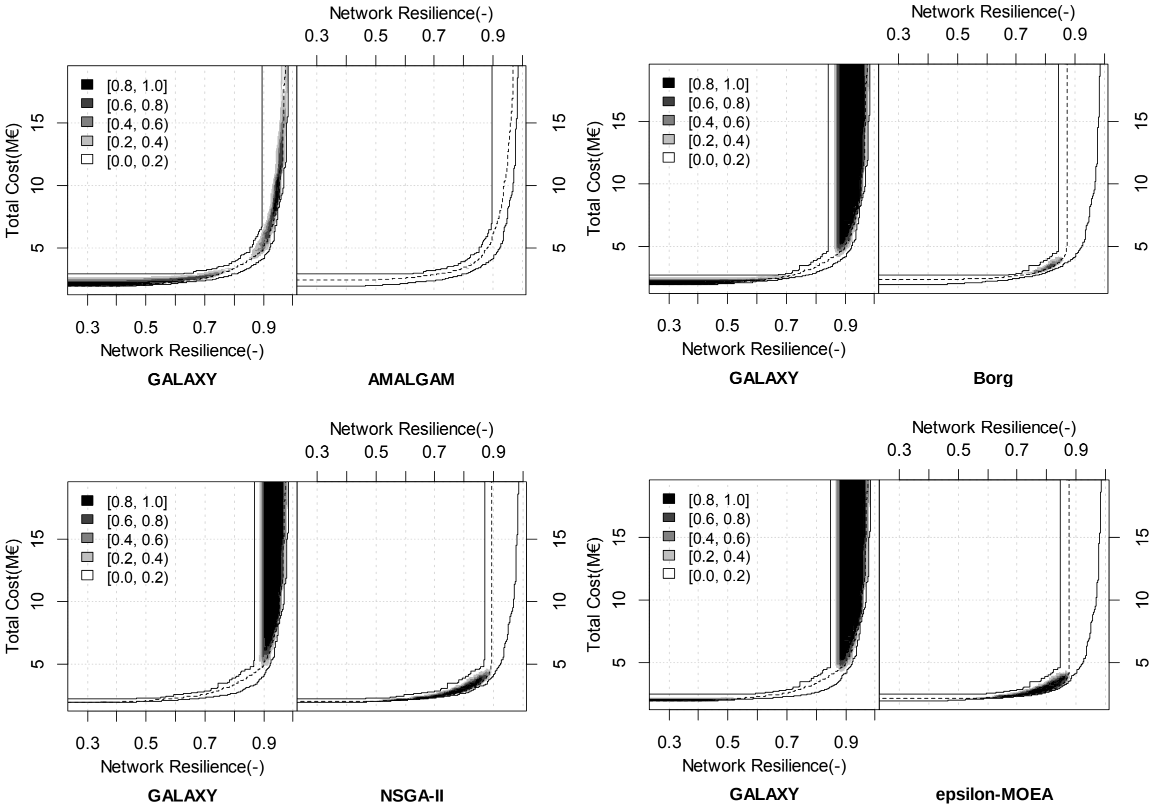


Figure 5.9 Comparison of GALAXY with the other MOEAs for the PES problem using the EAF tool

For the EXN problem, as can be seen from Table 5.7, Borg turned out to be the top performing algorithm by achieving a similar level of convergence compared to the other MOEAs but with evidently better results in terms of I_{HV} and $I_{\varepsilon+}$. None of the methods could discover any solutions which were near to the best-known PF (I_{EP} is equal to zero), and this fact can be attributed to the computational budget (i.e., NFEs) for the EXN problem was not sufficient at all. So, none of the algorithms converged. GALAXY was completely dominated by the other MOEAs, and its failure can be attributed to its struggle against the infeasible solutions for more generations (See Figure 5.14) compared to the other MOEAs, which in turn significantly hindered the search in the feasible region. This point will be further explained in Section 5.3.2.

From the comparisons of EAF (see Figure 5.10), GALAXY was able to find some better solutions than Borg in the region of high network resilience, although it was convincingly outperformed by the other MOEAs. It is also worth noting that the MOEAs moved towards different directions in the objective space. GALAXY, AMALGAM and NSGA-II tended to converge towards the region of high network resilience, whereas the Borg and ε-MOEA attempted to converge towards the region of low network resilience.

Table 5.7 Ultimate performance of various MOEAs for the EXN problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	0.9359	**0.9624**	0.9621	0.9591	0.9597
	Max.	0.9460	**0.9700**	0.9686	0.9639	0.9664
	Min.	0.9262	**0.9566**	0.9514	0.9547	0.9511
	Std.	0.0050	0.0031	0.0032	**0.0026**	0.0037
I_{HV}	Avg.	0.7822	0.8371	**0.8553**	0.8371	0.8346
	Max.	0.8024	0.8503	**0.8695**	0.8589	0.8445
	Min.	0.7628	**0.8286**	**0.8286**	0.8213	0.8160
	Std.	0.0101	**0.0054**	0.0096	0.0080	0.0063
$I_{\varepsilon+}$	Avg.	0.8524	0.8773	**0.9240**	0.8866	0.8972
	Max.	0.8696	0.8868	**0.9407**	0.8982	0.9090
	Min.	0.8367	0.8672	**0.9051**	0.8712	0.8776
	Std.	0.0092	0.0056	0.0084	0.0062	**0.0054**
I_{EP}	Avg.	0.0000	0.0000	0.0000	0.0000	0.0000
	Max.	0.0000	0.0000	0.0000	0.0000	0.0000
	Min.	0.0000	0.0000	0.0000	0.0000	0.0000
	Std.	0.0000	0.0000	0.0000	0.0000	0.0000

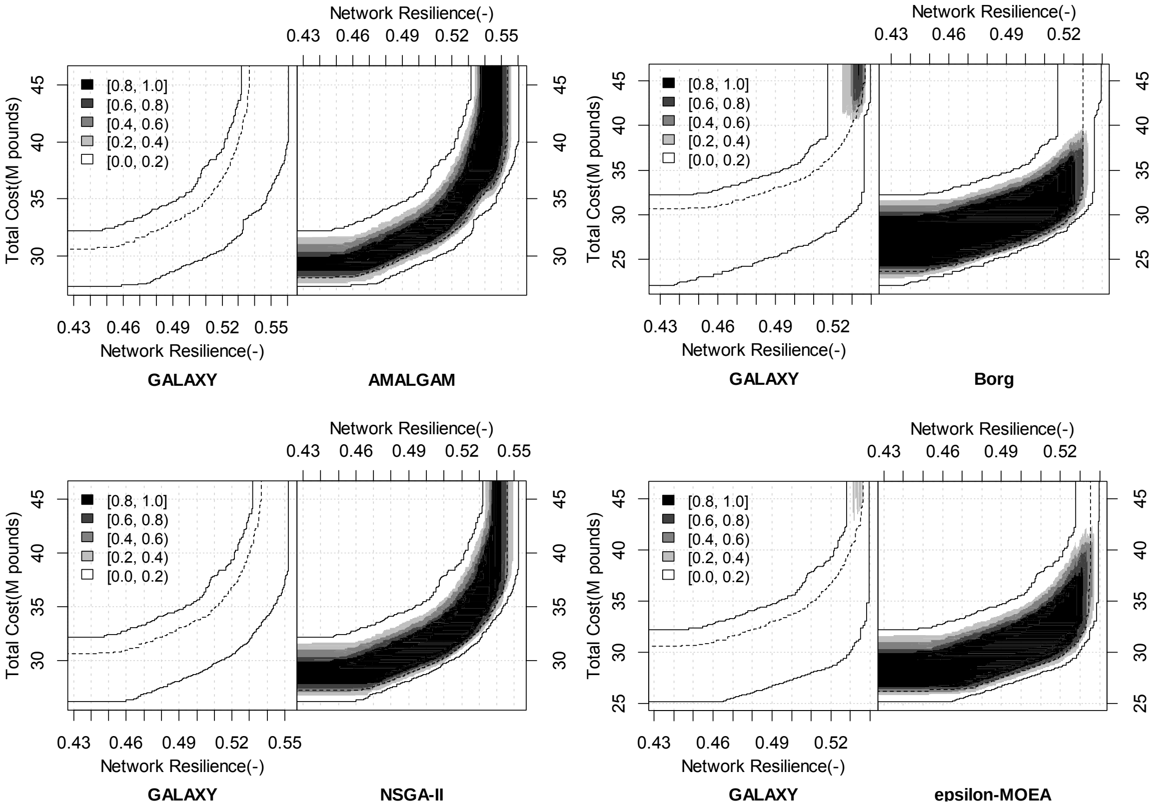


Figure 5.10 Comparison of GALAXY with the other MOEAs for the EXN problem using the EAF tool

5.3.2. Dynamic Performance

In this section, the dynamic performances of each MOEA across generations in terms of four performance indicators (see Section 5.2.1) are illustrated and compared in Figures 5.11-5.14. In each figure, the solid line denotes the average performance according to an indicator (on the left vertical axis), whereas the dashed line denotes the standard deviation of an indicator through thirty runs (on the right vertical axis). Additionally, the grey area indicates the range of variations of an indicator during multiple runs. To facilitate the intuitive comparisons, the performances of MOEAs are plotted individually and organised in a stacked fashion. So, each row represents the dynamic performances of a specific MOEA in terms of four indicators; whereas each column represents the dynamic performances of various MOEAs in terms of a specific indicator.

Generally speaking, the dynamic performances in terms of I_{GD}, I_{HV} and $I_{\varepsilon+}$ for different MOEAs turned out to be similar; that is, a significant improvement was achieved in the early stages of a run (usually less than one fifth of total generations) and then consistent but minor gains were maintained until the end of the search. It is worth pointing out that the concept of 'generations' for the ε-dominance based MOEAs (i.e., Borg and ε-MOEA) differs from that of the other MOEAs that are based on the Pareto-dominance concept. Unlike the generational framework, the ε-dominance based MOEAs feature a steady-state framework and update the external archive and the population in real time (i.e., after the creation of a new individual). However, to perform the comparison with the Pareto-dominance based MOEAs, the archive solutions at the interval of a 'generation' are selected to calculate various indicators.

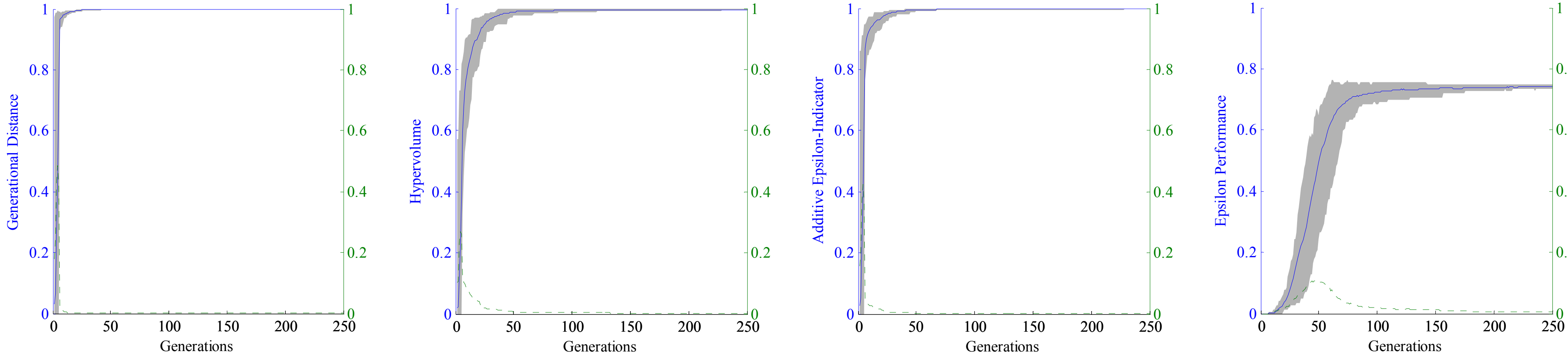
For the BAK problem, it can be observed from Figure 5.11 that all the MOEAs demonstrated good performances as mentioned in Section 5.3.1. But GALAXY was able to reach a much higher I_{EP} compared to the other MOEAs. The AMALGAM and NSGA-II stagnated in terms of I_{EP} after around 50 generations; in contrast, the other MOEAs consistently improved I_{EP} across generations, thus benefiting from the ε-dominance based replacement.

For the HAN problem, it can be observed from Figure 5.12 that GALAXY consistently achieved better performances than the other MOEAs in terms of all the indicators. It is interesting to see that the AMALGAM and NSGA-II were able to exceed Borg and ε-MOEA in terms of I_{EP}. In addition, all the MOEAs struggled to move out of the infeasible region at the first few generations.

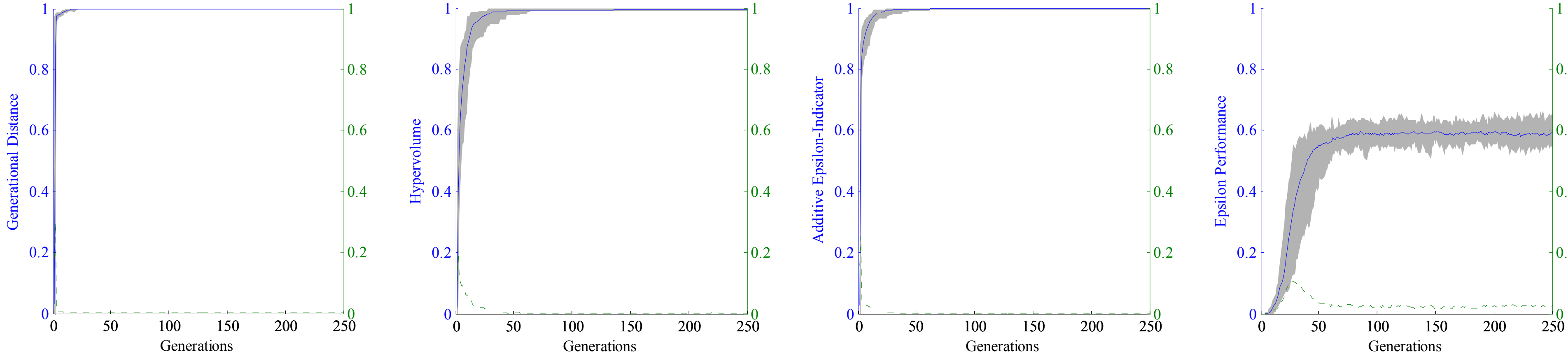
For the PES problem, it can be seen from Figure 5.13 that the performances of various MOEAs look very similar and GALAXY again turned out to be the topmost performer by consistently obtaining better results in terms of each indicator. Note that GALAXY was the only algorithm which successfully discovered near-optimal solutions on the best-known PF at around 200 generations. This exceptional progress can be attributed to the inclusion of the TF operator, being extremely efficient in the first few generations in approaching the boundary solutions in the region of high network resilience.

For the EXN problem, it is evident in Figure 5.14 that GALAXY took much more effort than the other MOEAs to pass through the infeasible region, thus resulting in a relatively poor achievement at the end of the search. If it were able to move out of the infeasible region much sooner, GALAXY would probably achieve very similar performances, if not better. This expectation was hinted by the fact that GALAXY demonstrated relatively faster improvements especially for I_{HV}. On the other hand, it is implied that the quality of initial population greatly impacts on the performance of an MOEA for solving a very complex design problem, which contains a large portion of infeasible solutions. Borg undoubtedly beat the other MOEAs by effectively finding feasible solutions at the early stage for such a complex design problems. As it will be discussed in Section 5.3.3, this superior performance benefited from the usage of various multi-parent search operators (i.e., SPX, PCX and UNDX). The dynamic performances of all the MOEAs in terms of I_{EP} were not shown here because they completely failed on this aspect.

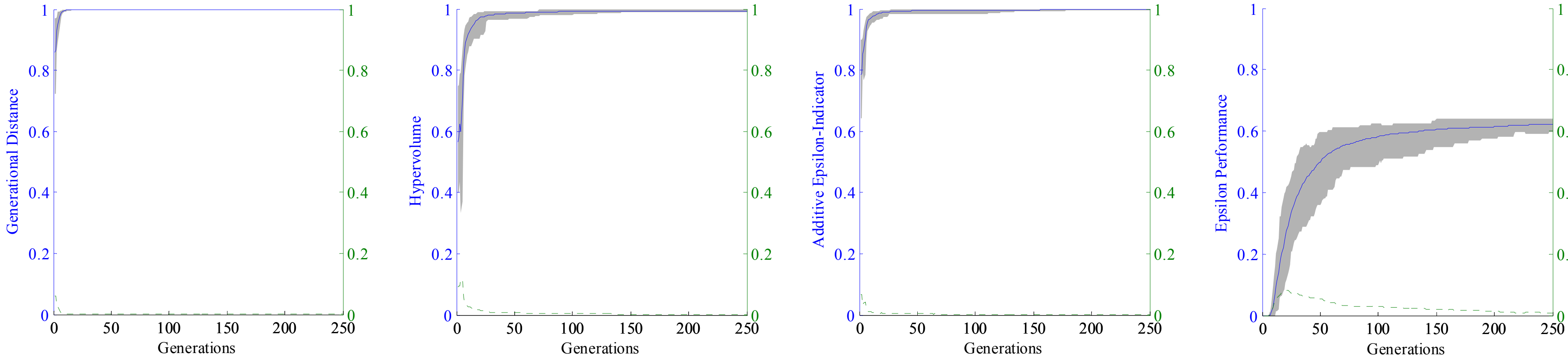
GALAXY



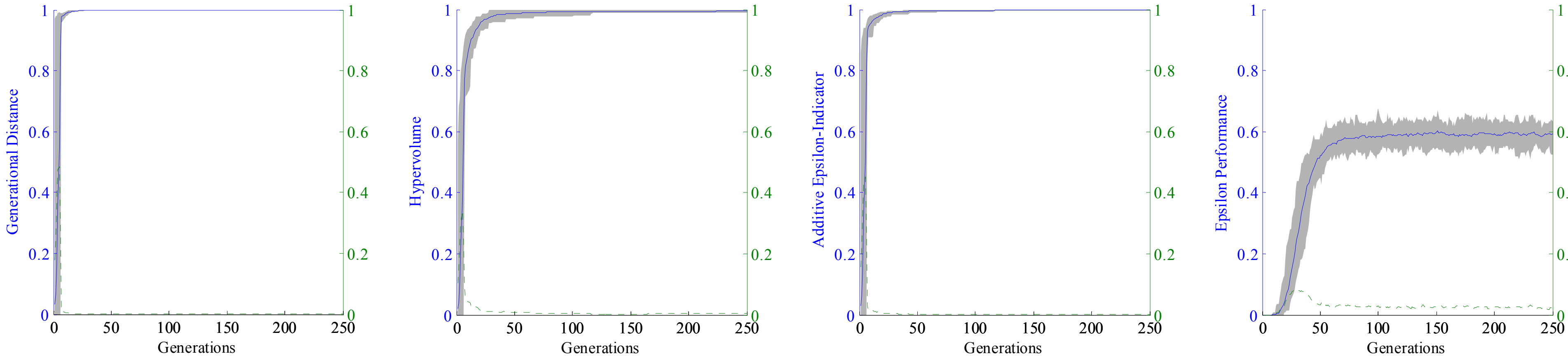
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0
50
100
150
200
250
0
0.2
0.4
0.6
0.8
1

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0
50
100
150
200
250
0
0.2
0.4
0.6
0.8
1

ε-MOEA

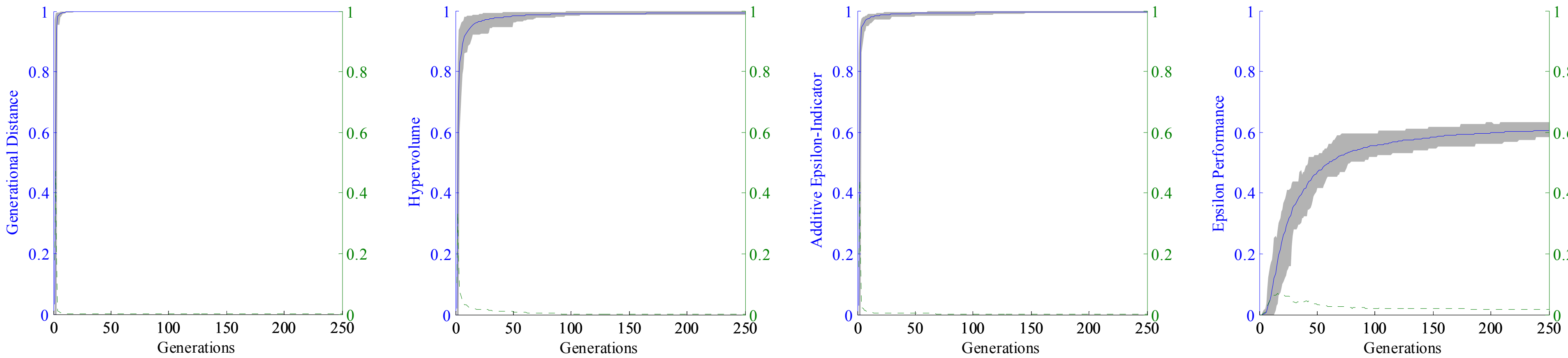
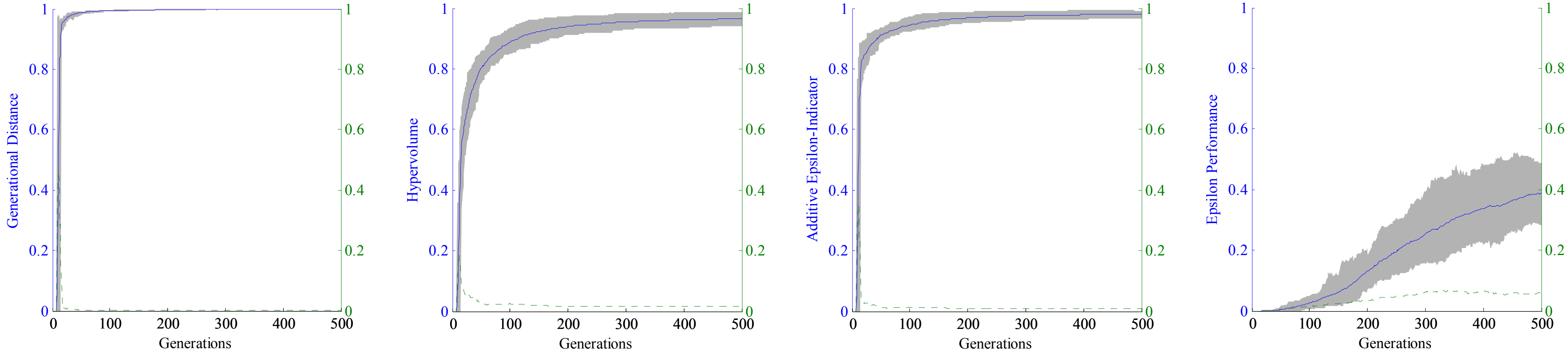
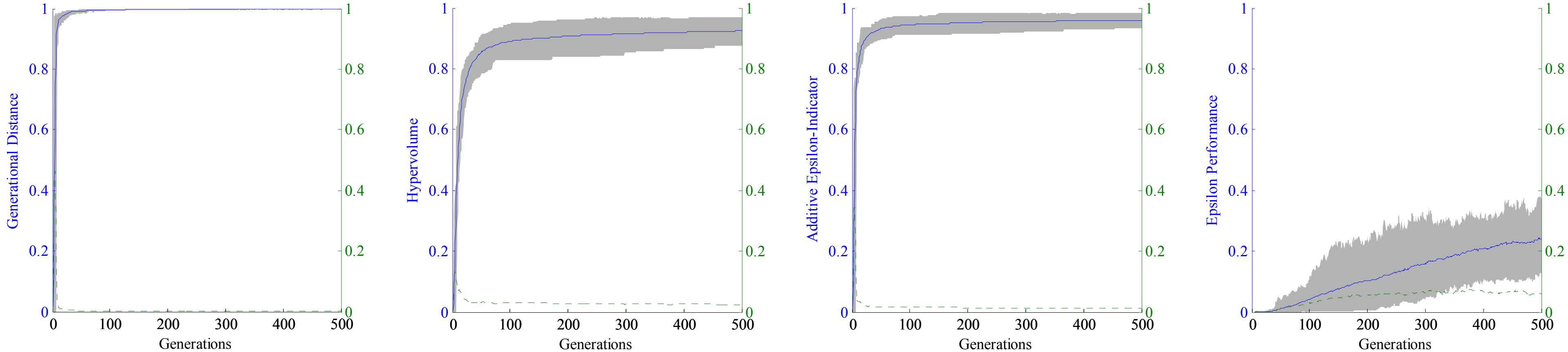


Figure 5.11 Dynamic performances of various MOEAs for the BAK problem

GALAXY

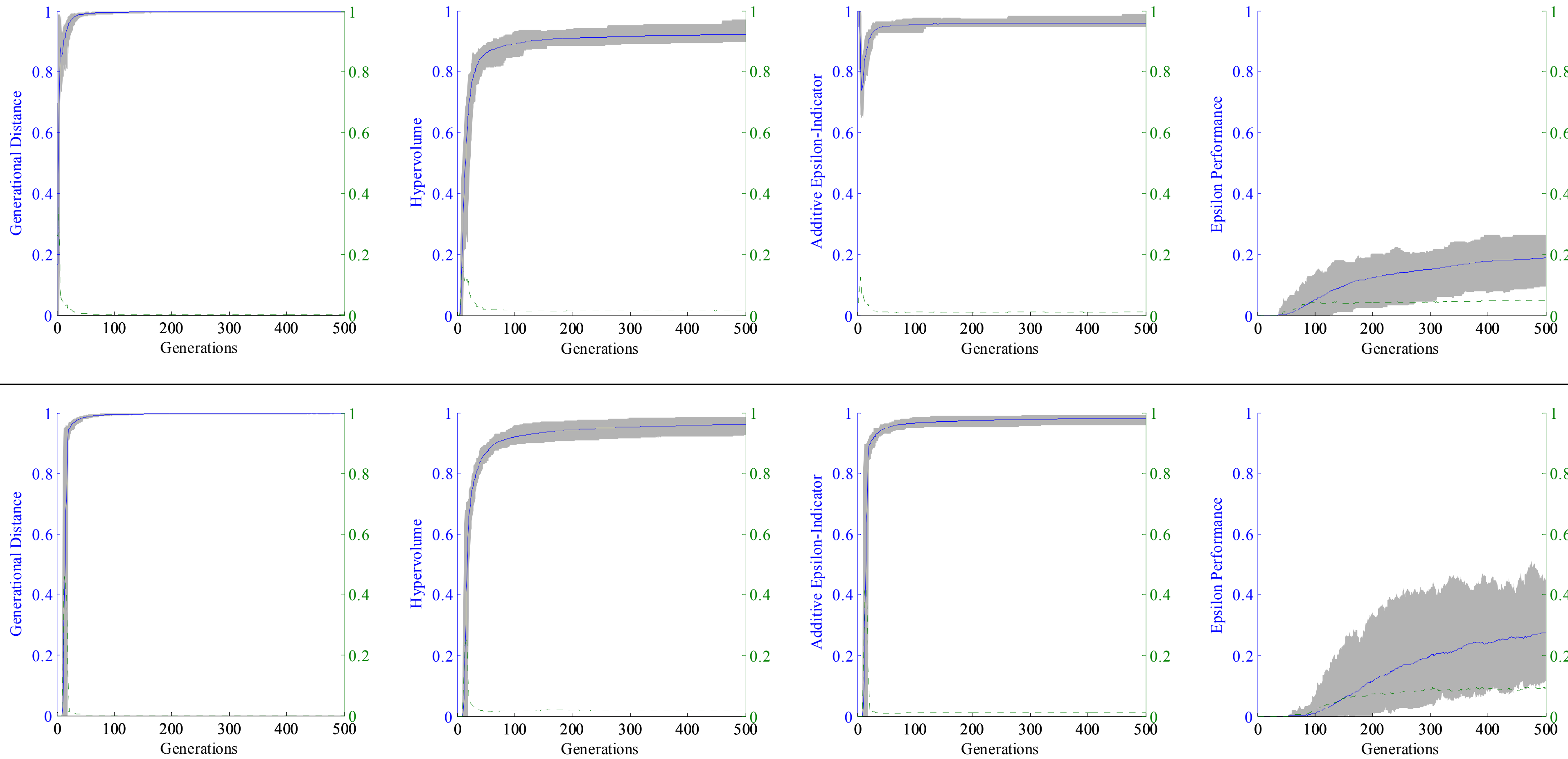


AMALGAM



Borg

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 100 200 300 400 500
0 0.2 0.4 0.6 0.8 1

ε-MOEA

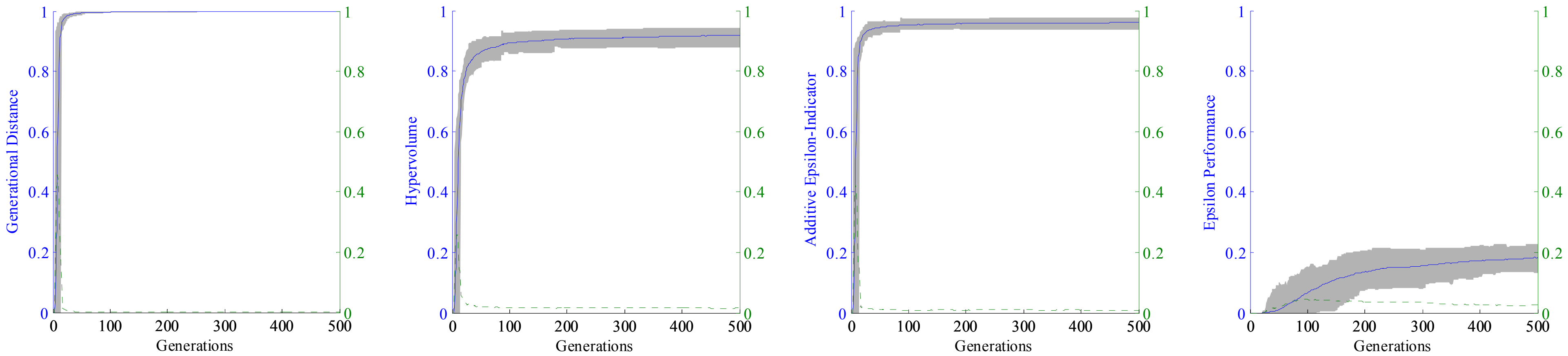
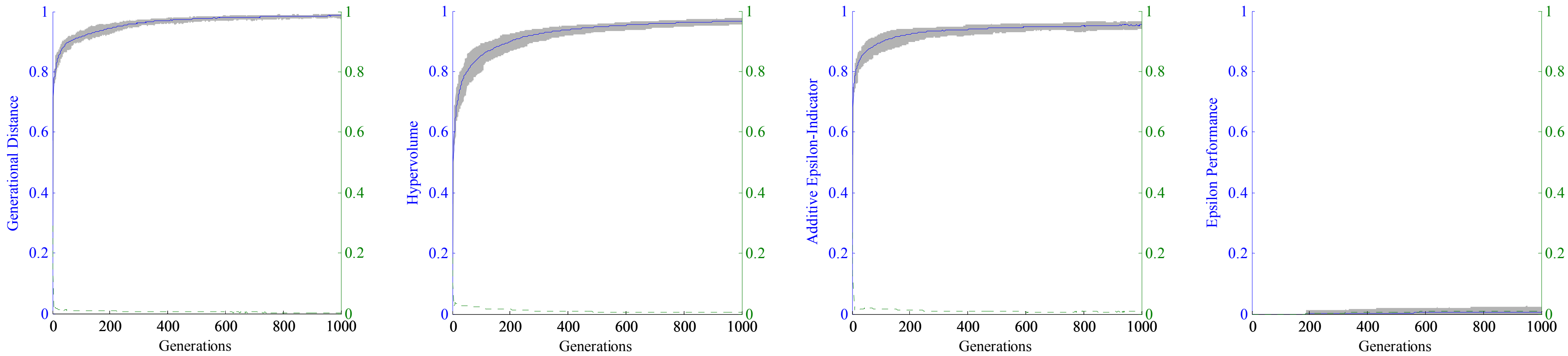
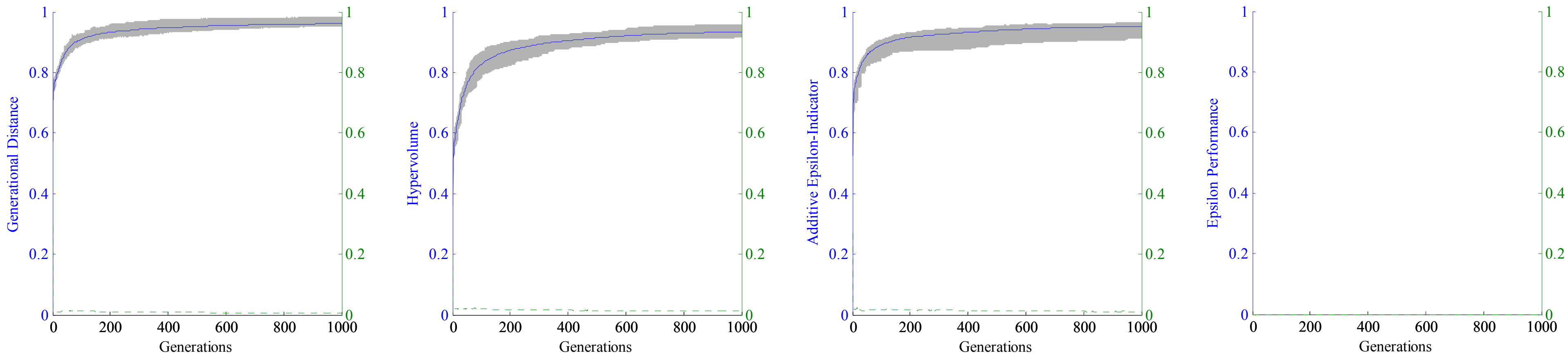


Figure 5.12 Dynamic performances of various MOEAs for the HAN problem

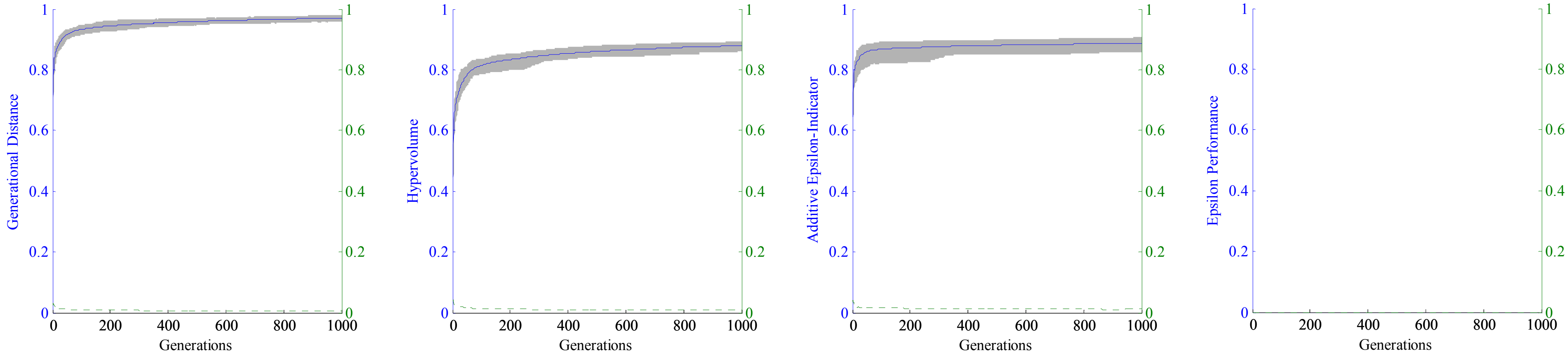
GALAXY



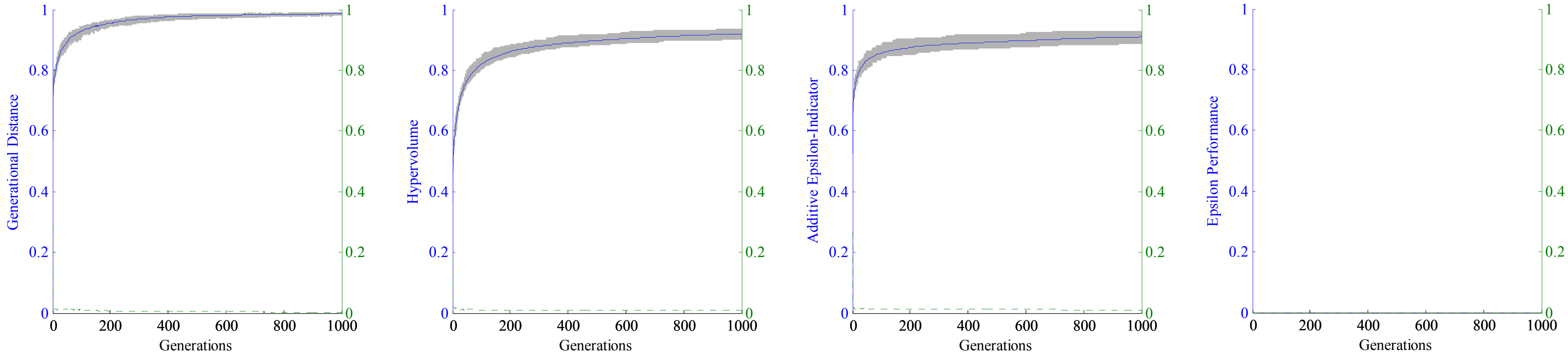
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
Generations
Generations
Generations

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
Generations
Generations
Generations

ε-MOEA

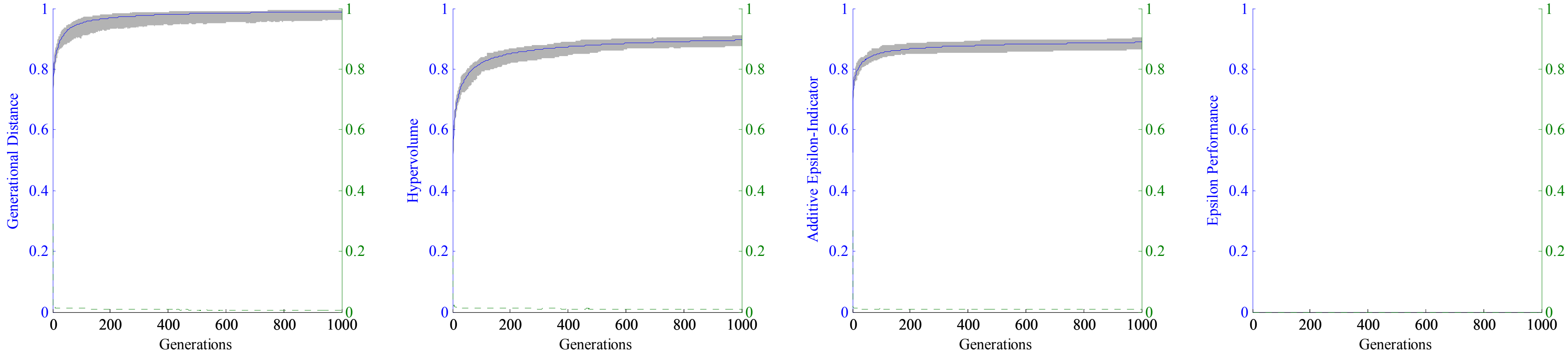
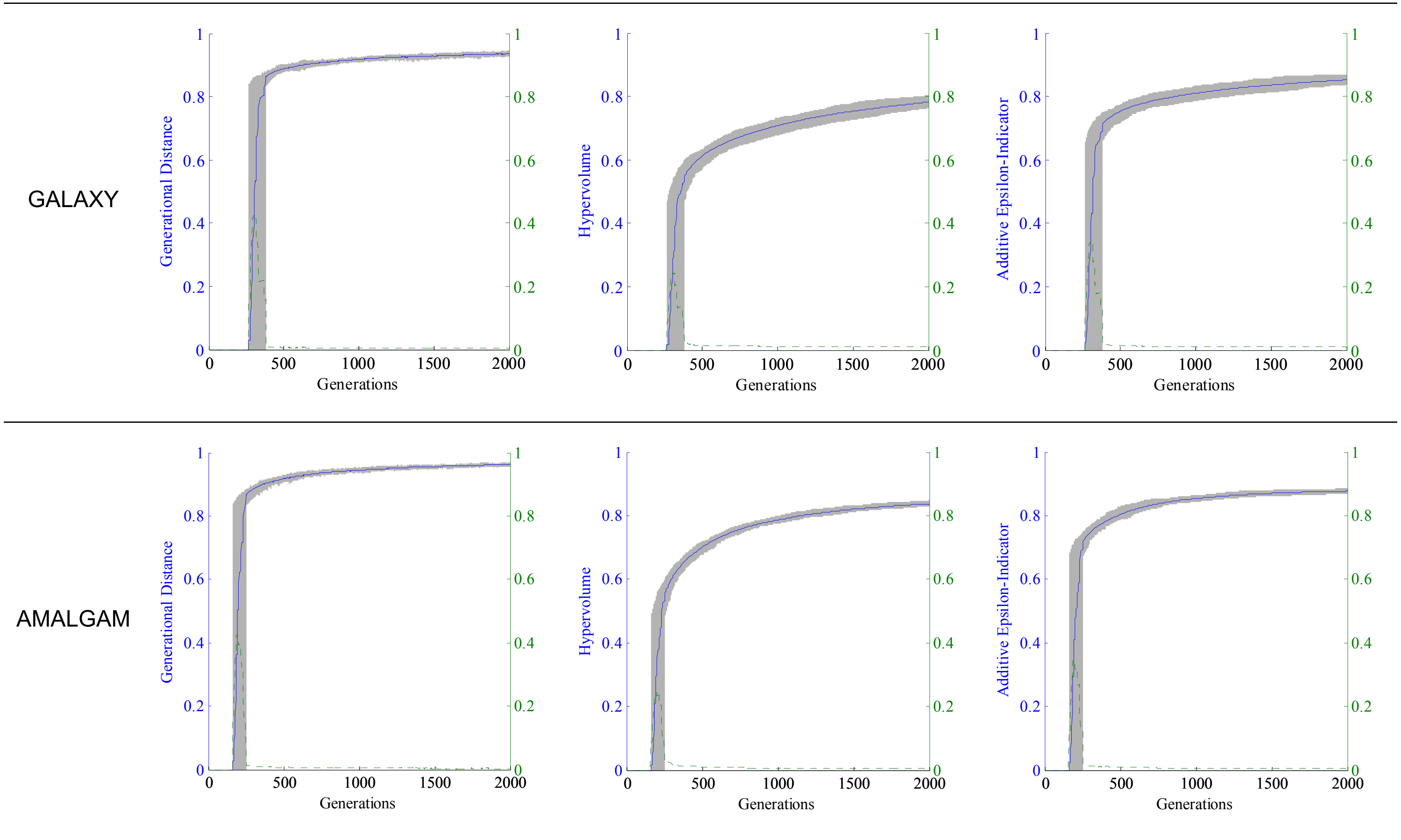
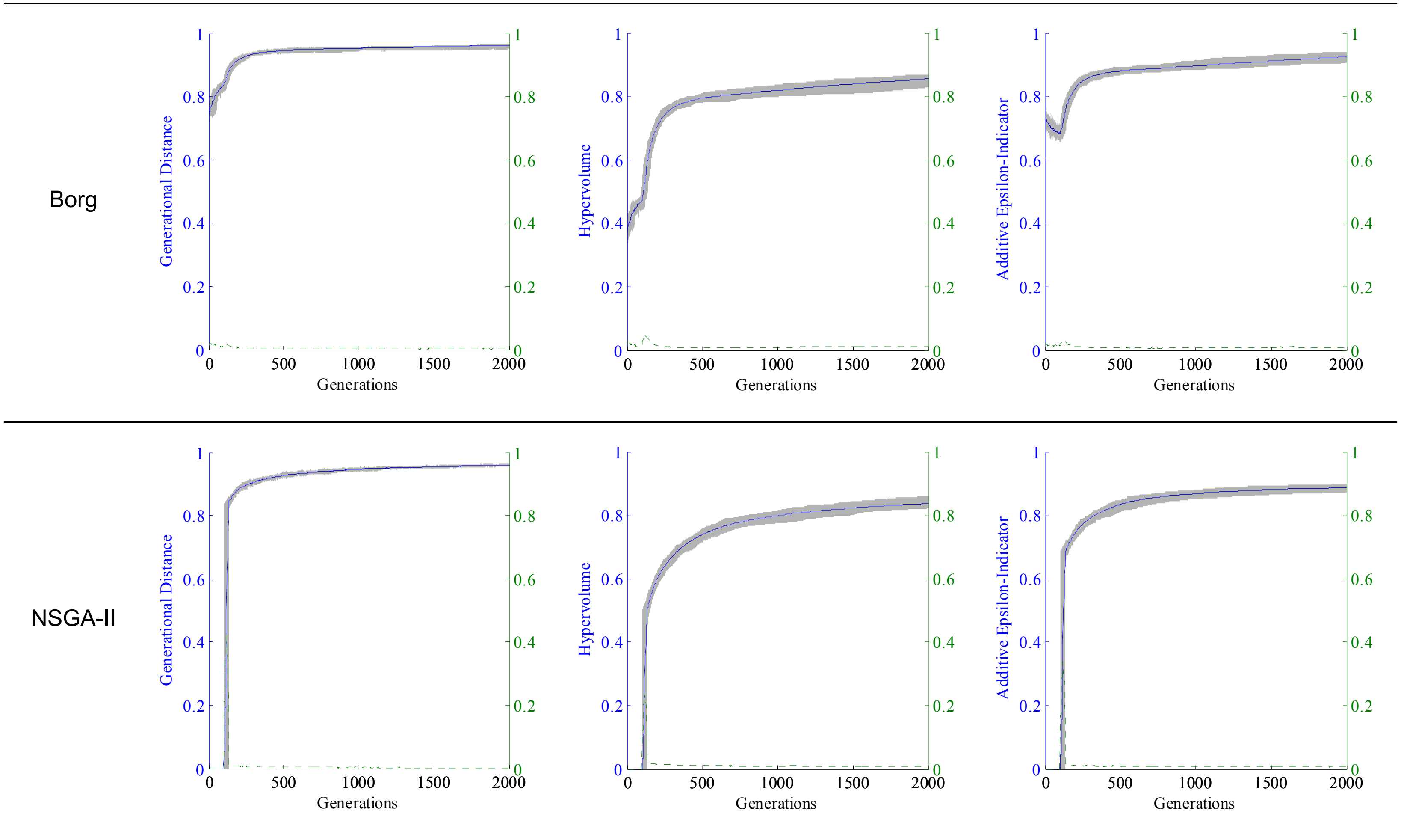


Figure 5.13 Dynamic performances of various MOEAs for the PES problem


GALAXY
AMALGAM
Generational Distance
Hypervolume
Additive Epsilon-Indicator
Generations
0
500
1000
1500
2000
0
0.2
0.4
0.6
0.8
1


Borg
NSGA-II
Generational Distance
Hypervolume
Additive Epsilon-Indicator
Generations
0
500
1000
1500
2000
0.2
0.4
0.6
0.8
1

ε-MOEA

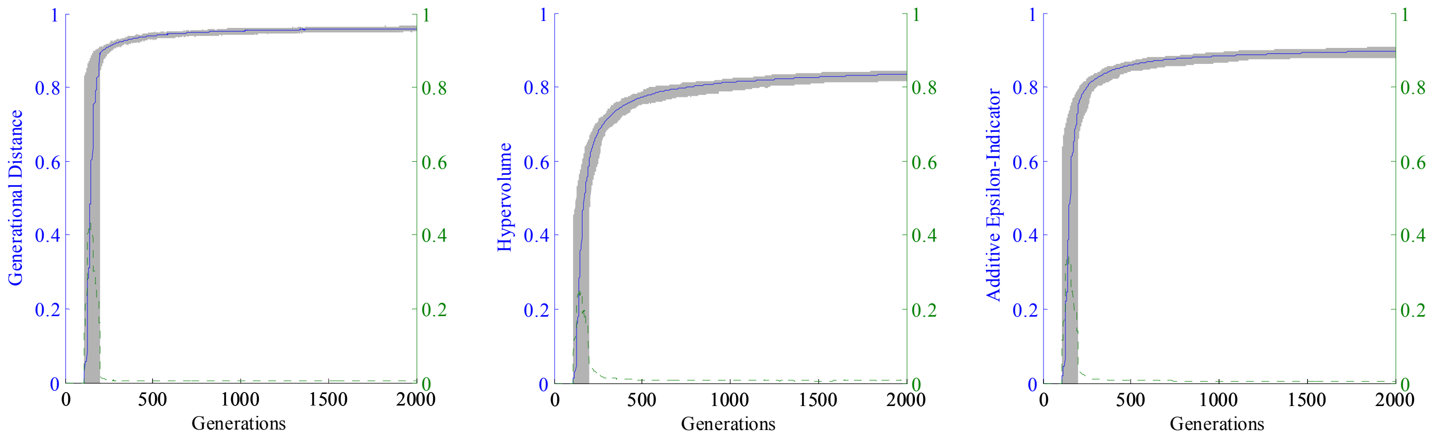


Figure 5.14 Dynamic performances of various MOEAs for the EXN problem

5.3.3. Dynamic Variations of Search Operators

In this section, the dynamic variations of multiple search operators within the hybrid algorithms are illustrated and further analysed. Figures 5.15, 5.16, 5.17, and 5.18 show the corresponding variations of search operators within hybrid algorithms for BAK, HAN, PES, and EXN, respectively. In each figure, the solid line denotes the average probability that each operator is employed for generating the offspring, and the grey area indicates the range of variations of this probability during thirty runs. These figures are particularly useful in understanding the behaviour of a hybrid method for a specific design problem. It is also worth mentioning that the adaptation of the portfolios of multiple operators in Borg differs from GALAXY and AMALGAM due to its steady-state structure. In Borg, each operator is selected for generating the offspring with a probability which is equal to its contribution to the members in the ε-box dominance archive. This approach favours operators yielding high quality individuals in terms of both convergence and diversity. In contrast, GALAXY and AMALGAM, which do not use external archives, count the successful reproduction rate of each operator and adaptively distribute the portfolios for the next generation (see Section 3.2.4 in Chapter 3).

For the selected cases, GALAXY consistently exhibited the desired characteristics. More specifically, the first three search operators (i.e., TF, DE and SBXI) with relatively strong leaping ability in the global sense made the contribution mainly during the first few generations (less than 25% of total generations). In contrast, the remaining operators (i.e., UM, GM and DC) with relatively strong leaping ability in the local sense gradually increased their employability towards the end of the search, indicating that it progressively moved from exploration to exploitation. This point is particularly evident for the DC operator, which consistently improved its probability for reproduction. However, for the EXN problem (see Figure 5.18), the TF operator did not seem very efficient, leading to low progress of GALAXY in the region of infeasible solutions.

For AMALGAM, the PSO was not useful most of the time. The AMS operator worked very well for the BAK (see Figure 5.15) and HAN (see Figure 5.16) problems, accounting for about 40% and more than 50% probability, respectively. But it failed to perform effectively for the PES (see Figure 5.17) and EXN (see Figure 5.18) problems. GA and DE constantly contributed many high quality solutions to the next population for the first three cases. For the EXN problem, AMS helped AMALGAM to pass through the infeasible region; afterwards, its influence subsided as PSO so that it made no contribution to the population. Thus, GA became the dominating operator and AMALGAM behaved very similar to NSGA-II but with smaller effective population size, because nearly 7.5% of population size were occupied by ineffective operators (PSO and AMS). Therefore, AMALGAM's performance was expected to be worse than that of NSGA-II.

For Borg, the dynamic variations of search operators were different from case to case. SBX and SPX made major contributions to the ε-box dominance archive for the BAK and HAN problems. PCX also performed reasonably well for the HAN problem, and it became the dominant operator for the PES problem by taking up nearly 90% probability. For the EXN problem, SBX, PCX and UM were major operators, and UM steadily increased its probability up to nearly 90% from about 500 generations. Although DE and the other two multi-parent operators (i.e., SPX and UNDX) generally performed poorly, they indeed made important contributions in improving the feasibility of the ε-box dominance archive.

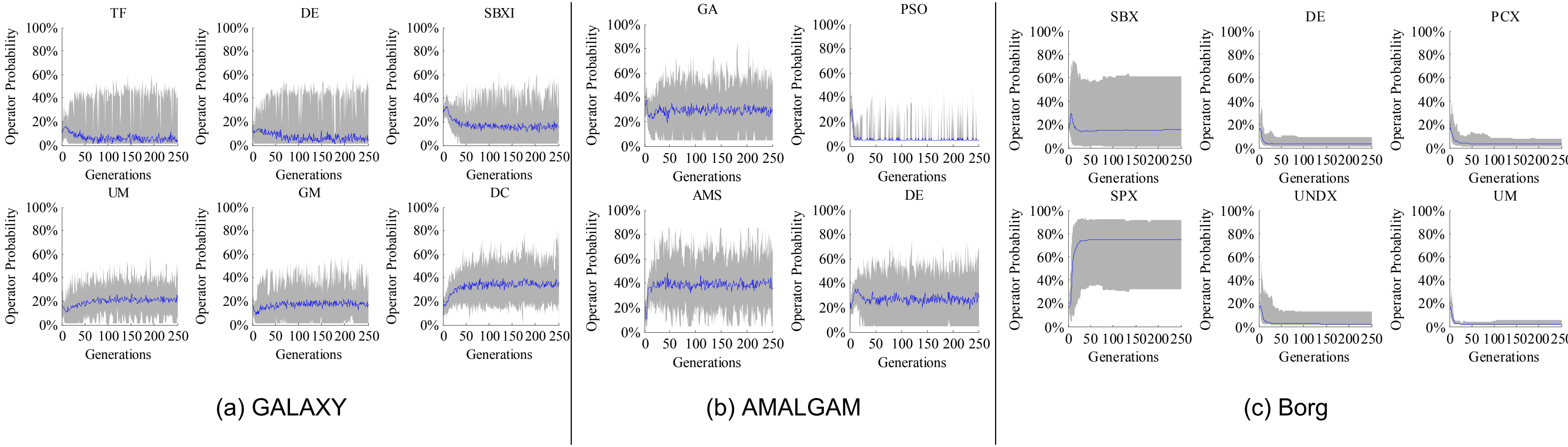
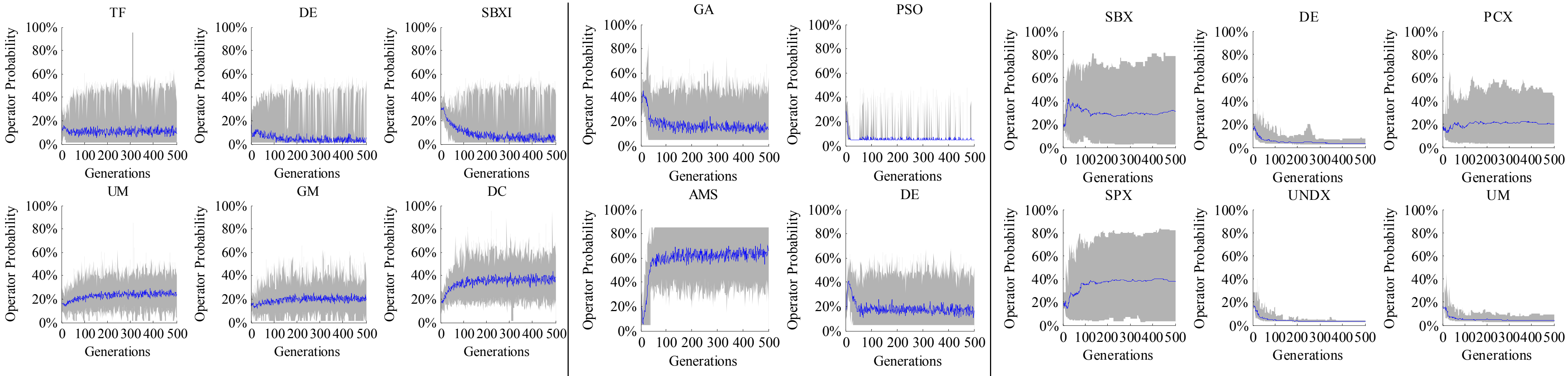


Figure 5.15 Dynamic performances of search operators of hybrid algorithms for the BAK problem



(a) GALAXY (b) AMALGAM (c) Borg

Figure 5.16 Dynamic performances of search operators of hybrid algorithms for the HAN problem

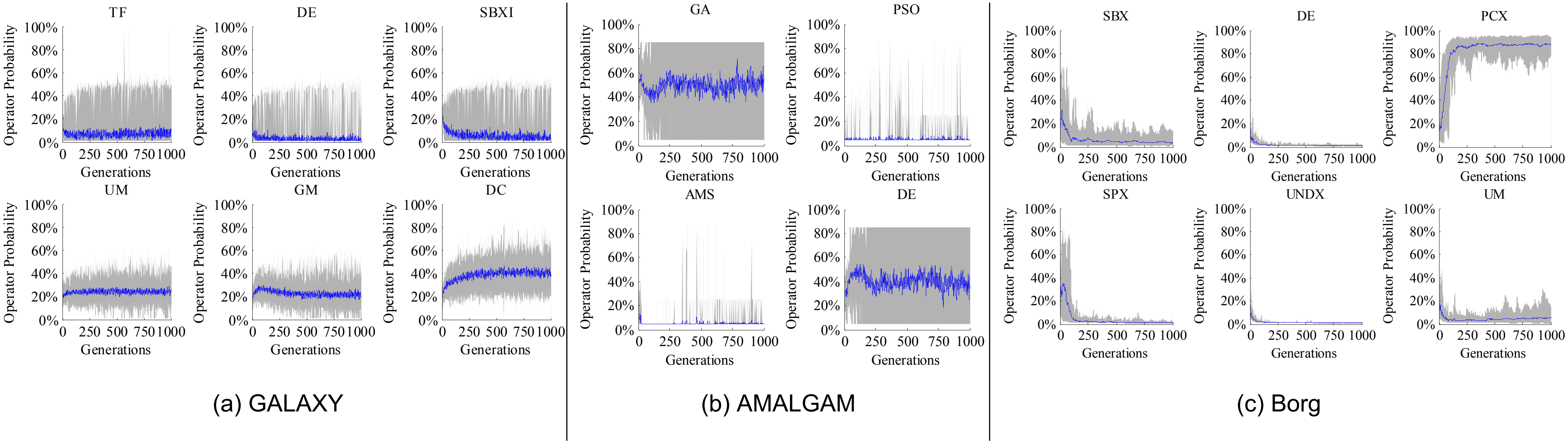


Figure 5.17 Dynamic performances of search operators of hybrid algorithms for the PES problem

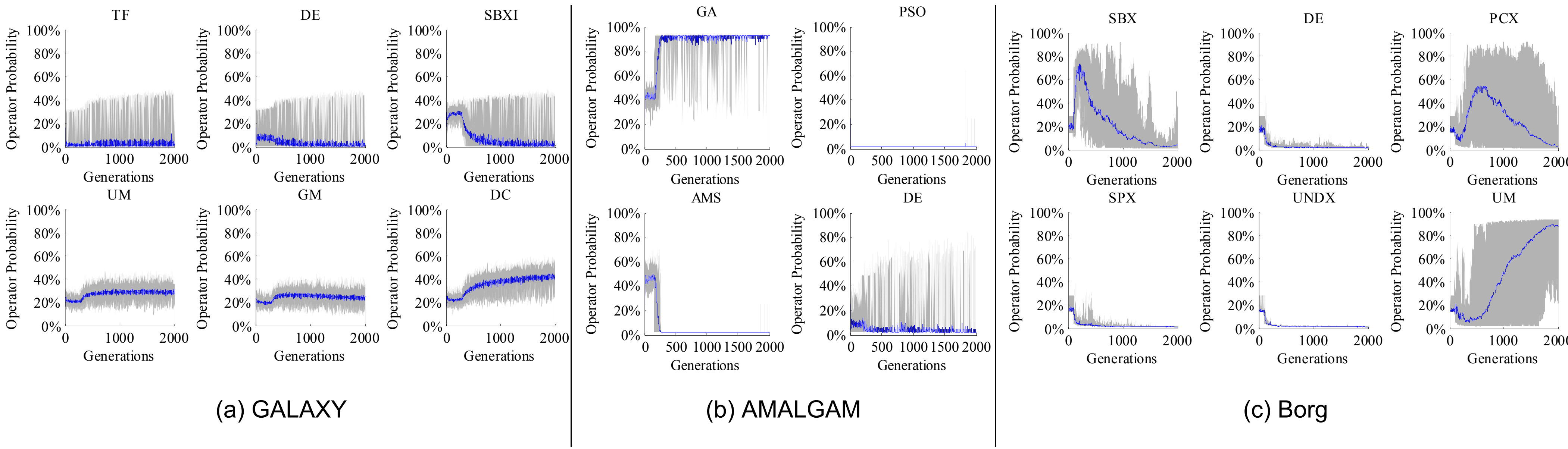


Figure 5.18 Dynamic performances of search operators of hybrid algorithms for the EXN problem

5.3.4. Results of the Other Benchmark Problems

Due to the limited space, results for the other benchmark problems (tables and figures) are presented in Appendix D. A brief summary of these results is given below. Generally speaking, GALAXY consistently outperformed the other MOEAs for the majority of these problems. It achieved very similar I_{GD} to the others, if not better, but attained much better I_{HV}, $I_{\varepsilon+}$ and I_{EP} values.

However, for the NYT problem the hybrid MOEAs were outperformed by the baseline MOEAs (i.e., NSGA-II and ε -MOEA), although the differences between GALAXY and the best values of I_{GD}, I_{HV} and $I_{\varepsilon+}$ were not substantial. The ε-MOEA successfully covered around 24% solutions of the best-known PF focusing around the 'knee point', whereas the others were only able to locate less than 14% of these solutions. Borg turned out to be the worst algorithm in terms of all the indicators. Recall that NYT and EXN are extended design problems, in which a subset of pipes are considered for duplication. Therefore, the quality of a solution (in terms of objective function values) is also affected by the conditions of existing pipes in the system. This feature differentiates the extended design problems from pure design problems, in which all the pipe sizes need to be optimised.

On the other hand, according to the theoretical assumption made in this thesis, the search operators play a crucial role in the performance of hybrid algorithms. As it can be seen from Figure D.19 and Figure 5.18, SBX and PCX turned out to be very effective in dealing with extended design problems. This can partially explain why the baseline MOEAs (i.e., NSGA-II and ε-MOEA) were superior to hybrid algorithms for the NYT problem, and Borg became the best algorithm for the EXN problem. More specifically, for the NYT problem SBX was the best operator so that the baseline MOEAs which only contains SBX and PM converged much quicker compared to hybrid algorithms. While for the EXN problem, PCX and SBX appeared to be more efficient to overcome the influence of existing pipes, thus leading to the best overall performance of Borg. GALAXY, in contrast, did not involve the 'standard' SBX operator as well as other multi-parent operators (e.g., PCX), and thus it showed relatively poor performance on the extended design problems.

GALAXY was also slightly outperformed by AMALGAM according to I_{HV} and $I_{\varepsilon+}$ for the GOY problem. NSGA-II marginally outperformed the others in terms of I_{EP} but it failed to cover the boundary solutions in the region of high network resilience. The success of AMALGAM for the GOY problem can be attributed to the merit of AMS, which effectively explored the Pareto distribution and found high quality solutions.

It is worth noting that for intermediate and large problems, GALAXY was constantly superior to the other MOEAs according to various indicators. Surprisingly, none of the algorithms were able to locate solutions close to the best-known PF for the MOD problem, but GALAXY successfully identified a few boundary solutions for the BIN problem, whose search space is much larger than that of MOD. This again emphasised the importance of incorporating operators that have better exploration capability in the global sense (e.g., TF) into the hybrid framework.

Regarding the dynamic variations of search operators of hybrid algorithms, GALAXY consistently demonstrated the desired behaviour that operators with different leaping capabilities were utilised effectively and efficiently for exploration and exploitation. For AMALGAM, GA and DE were shown to be effective for a wide range of cases. However, PSO was found to have difficulty in finding high quality solutions except for the TLN, GOY and FOS problems. AMS performed very well for cases with less than 60 decision variables (i.e., from the TRN to FOS problems). However, it completely failed for the remainder of problems with large search spaces. For Borg, SBX, PCX and SPX were the dominant operators across a number of benchmark problems. UM together with PCX became the leading operators especially for large problems.

As it also can be seen from the related figures, the dynamic performance of GALAXY and the portfolios of its search operators were relatively stable across the benchmark problems. It always showed less, if not the least, variations given different initial populations. This is an extra advantage of GALAXY whose behaviour is robust with respect to the initial status and random seed.

5.4. Summary and Conclusions

This chapter applied the GALAXY method as well as the other four MOEAs to solve the benchmark problems in the WDSBA. In Section 5.2, four performance indicators and the EAF tool were first described for measuring and comparing the ultimate and dynamic performances of various algorithms. Then, the computational budgets for cases with different complexity were justified using the baseline NSGA-II. To facilitate a fair comparison, the parameterisation of each MOEA was carefully dealt with and selected. Results obtained for some representative cases were compared and discussed in Section 5.3. The other results will be presented in Appendix D due to the limited space.

The main conclusions of this chapter are as follows:

- GALAXY generally outperformed the other MOEAs in terms of both ultimate and dynamic performances on the majority of benchmark problems according to various performance indicators and the EAF tool. The algorithm was able to effectively capture boundary solutions, thus maintaining a wide spread of solutions across the objective space. The search operators within the GALAXY framework behaved as expected, gradually moving search from exploration to exploitation by adapting their portfolios. In addition, its performance was more stable compared to its competitors, which is another favourable characteristic in practice.
- However, the other MOEAs performed better than GALAXY for the EXN problem, and this was due to its struggle with infeasible solutions at early stages of the search. Borg, which successfully found feasible solutions in the first few generations, exhibited the best achievement. This may be attributed to the inclusion of many multi-parent search operators in its framework. However, it should be noted that all the algorithms were stopped before they sufficiently converged for the EXN problem, which

implied that GALAXY still had a chance to beat these MOEAs given more computational budget.

- Another important conclusion discovered in this chapter is that many performance indicators should be considered to assess an MOEA from different aspects, e.g., convergence, diversity and consistency. A combined analysis of four numerical indicators and the EAF tool employed in this Chapter provided a complementary way to evaluate an algorithm’s performance comprehensively.

6. ANYTOWN REHABILITATION PROBLEM

6.1. Introduction

In this chapter, the proposed hybrid algorithm GALAXY, as well as the other MOEAs considered in Chapter 5, are applied to solve the complex, multi-objective rehabilitation problem of the Anytown network (Walski et al. 1987). In Section 6.2, a concise history of this problem is given, followed by a clear problem description and formulation adopted in this thesis presented in Section 6.3. In Section 6.4, the experiment is set up by justifying the computational budget and parameter settings of each algorithm. All the methods are used to solve the problem multiple times independently and the results obtained are compared and discussed in Section 6.5. Finally, the chapter is summarised and relevant conclusions are drawn in Section 6.6.

6.2. A Concise History of the Anytown Rehabilitation Problem

In preparation for the first session of the 'Battle of the Network Models' (Walski et al. 1987), the Anytown problem was established as a rehabilitation and operation problem to reinforce a hypothetical USA water network to meet the projected demand in 2005. It represented a challenging benchmark problem, which closely resembled the features commonly found in many real systems. The competition required the participants to find the least-cost design of new pipes, tanks, and pumps under multiple loading conditions, including average daily flow, instantaneous peak flow and three fire flow scenarios. For more details about this battle, readers are referred to Walski et al. (1987).

Figure 6.1 shows the configuration of the Anytown network. Water is supplied from a clear well (node 40) with a fixed head of 10 ft (3.048 m) via three identical parallel pumps. Existing pipes located in the city region are shown in thicker solid lines, whereas the other solid lines represent the existing pipes in the residential region. The dashed lines denote new pipes under consideration. Two existing tanks are erected near node 14 and 17, each of which is linked by a short pipe, known as a riser, with a fixed length of 101 ft (30.785 m). The specific design requirements for this network will be presented subsequently in Section 6.3.

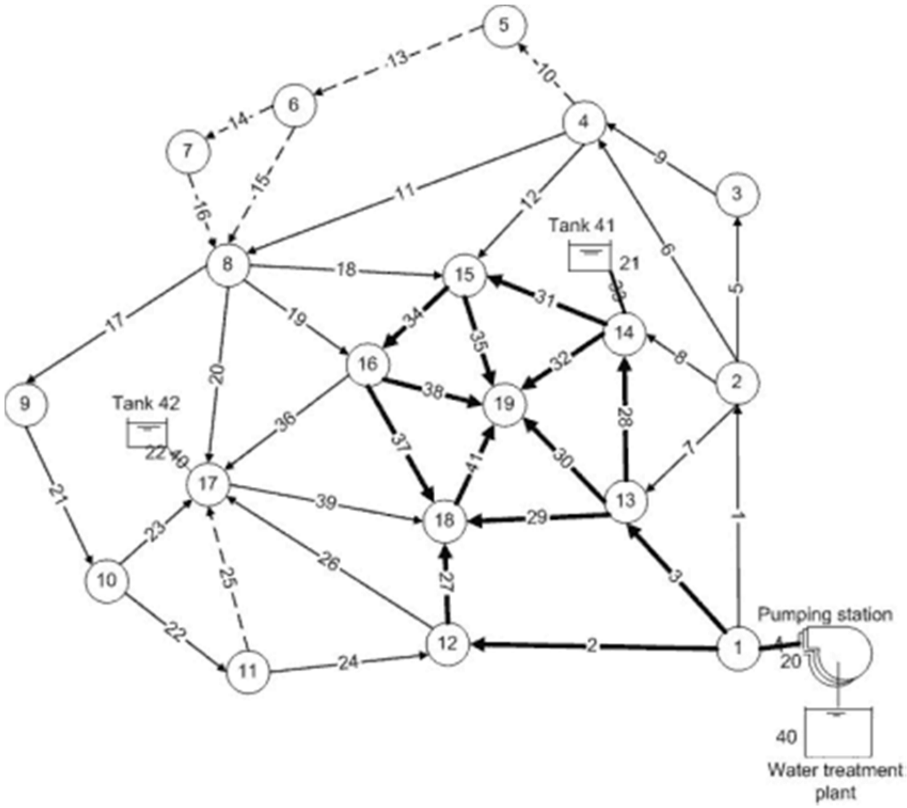


Figure 6.1 Configuration of the Anytown network (adopted from http://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/expansion/anytown.php)

Compared to other benchmark problems archived in Chapter 4, the Anytown problem gives reasonable consideration to both design and operations of a network system. This includes the introduction of a wide range of loadings, the design of new pipes, alternative options for existing pipes (duplication, or cleaning and lining), storage tank location and sizing as well as pump scheduling. In addition, the cost of pipe components not only depends on the diameter selected but also on the area (i.e., city or residential region) where the installation is going to take place. The capital and operational cost, which occur at different stages of the life-time of the system, are aggregated by considering the present worth of the operational cost based on an interest rate of 12% and an amortization period of 20 years.

In the ‘Battle of the Network Models’, although various methods in conjunction with manual calculations and engineering judgement were proposed to solve this complex problem, most solutions submitted by participants could not fully meet the design criteria, especially in the filling of tanks at off-peak loadings (Walski et al. 1987). Additionally, tank location and sizing were mainly determined subjectively rather than by incorporating them into an optimisation model. However, they all claimed that using optimisation models significantly improved the quality of their solutions. Several important insights into this problem were drawn from this battle, that is, tank location, tank sizing and pump operation dramatically affected pipe sizing, and thus these factors needed to be carefully considered during optimisation.

Later on, many researchers made efforts to examine this challenging problem in different ways. Table 6.1 summarises some work from the literature, followed by a brief survey of these publications.

Murphy et al. (1994) solved the Anytown problem using the Genetic Algorithm (GA) technique, handling pipe, pump and tank variables simultaneously in the optimisation model. The objective was to obtain a design with minimum capital and energy costs. To deal with infeasible solutions, a penalty cost, which was a function of the distance from feasibility, was added if a solution had constraint violations (pressure and/or tank level deficiency). Although a different interpretation was used from those presented in the ‘Battle of the Network Models’, they reported a much cheaper solution with a total cost of $11.335 million, which was about 8% less than the cheapest one obtained in the Battle. It should be noted that a GA was coupled with a steady-state hydraulic model, as it was too computationally expensive at the time to run a simulation that represented the dynamic operation over an extended period.

Walters et al. (1999) formulated the Anytown problem from a multi-objective perspective and applied an expanded Structured Messy Genetic Algorithm (SMGA) to include non-pipe decision variables (i.e., pumping installations and storage tanks). They managed to find even cheaper solutions compared to all the identified solutions in previously published papers. However, this may be due to the different interpretation of the problem. Specifically, they allowed new tanks to exceed the upper limit of the given standard sizes, i.e., 1 million gallons (4546.09 m^3), which made it possible to use the pumps in a more efficient way. It should be noted that no new pumping stations were considered, but pumps with identical characteristics to the existing ones could be installed in the current pumping station. In addition, the optimisation model was coupled with an extended period hydraulic model with a coarse simulation step (6 hours per time step). This was also limited by the unacceptable computational overhead for a more accurate simulation. However, a full simulation using a one hour time step was conducted to check the feasibility of the final solutions.

Table 6.1 Optimisation of the Anytown problem in the literature

Papers	Objectives	Optimisation Algorithms
(Murphy et al. 1994)	min(*Cost*+*Penalty*)	Genetic Algorithm
(Walters et al. 1999)	min(*Cost*) max([1]*Benefit*)	SMGA
(Farmani et al. 2005b)	min(*Cost*) max(minimum I_r)	NSGA-II
(Vamvakeridou-Lyroudia et al. 2005)	min(*Cost*) max(*Fuzzy Benefit*)	Genetic Algorithm + fuzzy reasoning
(Farmani et al. 2006)	min(*Cost*) max(minimum I_r) min(maximum *Water Age*)	NSGA-II
(Vamvakeridou-Lyroudia et al. 2006)	min(*Cost*) max(*Fuzzy Benefit*)	Genetic Algorithm + fuzzy reasoning
(Prasad and Tanyimboh 2008)	min(*Cost*) max(*Flow Entropy*)	NSGA-II
(Olsson et al. 2009)	min(*Cost*) min([2]*Shortfall*)	UMDA hBOA CSM NSGA-II
(Fu et al. 2012a)	min(*Cost*) min([2]*Shortfall*)	ε-NSGA-II + sensitivity analysis
(Wang et al. 2014b)	min(*Cost*) max(minimum I_n)	AMALGAM MOHO NSGA-II

Note: [1]Benefit was defined as a weighted sum of various deficiency measures with respect to the design criteria. [2]Shortfall was defined as a weighted sum of nodal pressure deficit and total tank level discrepancy. I_r denotes resilience index. I_n denotes network resilience.

Although the aforementioned work found solutions that could work at the maximum daily demand, these designs lacked adequate capacities to fill tanks at off-peak loadings. Therefore, Farmani et al. (2005b) dealt with the average day demand, the peak day demand, as well as three fire flow demands for the first time using the NSGA-II. The problem was formulated to minimise the total cost and to maximise the resilience index (Todini 2000). More accurate hydraulic simulations (i.e., one hour per time step) were considered throughout all the loading conditions. Due to the nature of the Anytown problem (discrete and highly constrained), they finally obtained 8 non-dominated solutions, which satisfied all the design requirements. Another contribution of this work was that it provided a clear definition of the decision variables, objective functions and constraints. It should be noted that the problem formulation differed slightly from that used by Walters et al. (1999) in that no new pumps were considered and only up to two cylindrical tanks could be added. Moreover, tank characteristics were treated as independent variables which resulted in accelerating convergence of the algorithm in the feasible region. Later on, Farmani et al. (2006) applied the NSGA-II to solve the three-objective design of the Anytown problem, taking the minimisation of water age (a surrogate measure of water quality) as the additional objective into account.

Vamvakeridou-Lyroudia et al. (2005) presented a practical way to solve the optimal design of the Anytown problem. GA was combined with fuzzy reasoning to search the solution space, aimed at minimising the costs and maximising a benefit/quality function. They successfully identified a solution which satisfied multiple criteria specified with reduced cost. However, it should be noted that their interpretation was different from those presented previously, especially for the treatment of storage tank variables. Subsequently, Vamvakeridou-Lyroudia et al. (2006) developed a multi-level, multi-criteria decision making process to solve the Anytown problem in a hierarchical way, which transformed the optimisation into an ‘engineer friendly’ decision support system.

Prasad and Tanyimboh (2008) solved the modified two-objective Anytown problem by proposing a tank design procedure and a surrogate measure for network reliability, called flow entropy (Tanyimboh and Templeman 1993). They assumed new tanks to be cylindrical and described their size implicitly. Besides the location of a tank, its volume, diameter-to-height ratio, minimum operational elevation, and fraction of minimum volume were regarded as independent variables. The flow entropy measures the uniformity of pipe flow rates and by maximising this indicator, a robust network under stressed conditions can be maintained. An extended period simulation was performed during optimisation with a hydraulic time step equal to 3 hours. The results demonstrated that their optimisation model was capable of exploring the search space effectively. However, it should be noted that this formulation was different from the original as well as the aforementioned versions by others researchers.

The Anytown rehabilitation problem was also handled in Olsson et al. (2009), Fu et al. (2012a) and Wang et al. (2014b). The main differences between these works lie in the problem formulation and methodologies (algorithms) employed. Olsson et al. (2009) tried to solve a two-objective, unconstrained optimisation problem, aiming at minimising the cost and the weighted sum of pressure deficiency and tank level discrepancy. They applied three genetic algorithms which used probabilistic methods to explore building blocks and compared their performance with NSGA-II. Fu et al. (2012a) formulated the Anytown problem in the sense of many-objective, including up to six criteria being considered concurrently. An enhanced version of NSGA-II, called ε-NSGA-II (Kollat and Reed 2006), was used to explore the high-dimensional space in an effective and efficient way. Wang et al. (2014b) focused on the two-objective, constrained formulation, but the second objective was substituted with the network resilience (Prasad and Park 2004), which is capable of alleviating some drawbacks of the resilience index (Todini 2000). They employed two hybrid evolutionary algorithms as well as NSGA-II.

In short, there exist various ways to formulate the Anytown benchmark problem and many algorithms have been applied to find optimal solutions. Nevertheless, these solutions obtained in the aforementioned papers cannot be directly compared, since the problem was interpreted from different viewpoints (e.g., number of objectives, variable representation and constraint handling). In other words, there is a lack of clear and uniform definition to benchmark different algorithms. The intention in the subsequent part of this chapter is to establish a detailed formulation, which is aimed at providing a constrained, multi-objective formulation (definition) of Anytown rehabilitation problem.

6.3. Problem Formulation

In this section, a brief problem description is given first. This is followed by the mathematical formulation of the Anytown problem adopted in this chapter.

6.3.1. Problem Description

The rehabilitation of Anytown network involves many possible options, including duplication of existing pipes, addition of new pipes, construction of new tanks and scheduling of the existing pump station. The candidate (feasible) solutions must satisfy various criteria with respect to the service standards and operational requirements. In addition, it is worth noting that Imperial units have been used in the subsequent parts of this chapter because of the original formulation of the Anytown rehabilitation problem.

Design Options

<u>Pipes</u>

For each existing pipe, there are three options, i.e., duplication, cleaning and lining or “do nothing”; whereas for a new pipe, it can either be added by choosing a suitable diameter or simply discarded. A total of 10 diameter options are commercially available and the associated unit price (as shown in Table 6.2) is determined by both the location (in city or residential area) and the purpose of construction (i.e., addition, duplication or cleaning and lining). Generally speaking, duplicating an existing pipe involves more disruption and reconstruction of pavement roads; hence, it incurs more expenditure than adding a new pipe or cleaning and lining an existing pipe. In addition, the excavation undertaken in a city area has a more significant impact than in a residential area, thus it is also considered to be more expensive. The roughness coefficients of existing pipes are projected C values for the year 2005 following the Hazen-Williams equation. Specifically, a cleaned and lined existing pipe has a C value of 125, whereas the C value of a new pipe is set to 130 by default. Old risers are not considered during optimisation, but new risers need to be specified if new tanks are built in the network system.

Table 6.2 Pipe intervention options and associated costs

Pipe diameter	New pipes	Duplicating existing pipes		Clean and line existing pipes	
		City	Residential	City	Residential
(in)	($/ft)	($/ft)	($/ft)	($/ft)	($/ft)
6	12.8	26.2	14.2	17	12
8	17.8	27.8	19.8	17	12
10	22.5	34.1	25.1	17	12
12	29.2	41.4	32.4	17	13
14	36.2	50.2	40.2	18.2	14.2
16	43.6	58.5	48.5	19.8	15.5
18	51.5	66.2	57.2	21.6	17.1
20	60.1	76.8	66.8	23.5	20.2
24	77	109.2	85.5	30.1	-
30	105.5	142.5	116.1	41.3	-

Note: Since the diameters of existing pipes in the residential region are less than or equal to 20 inch, the cleaning and lining option for the diameter larger than this size is not applicable.

Pumps

Three identical pumps are utilised in the system and a five-point characteristic curve (flow-head or flow-efficiency) can be derived from the data given in Table 6.3. A constant energy tariff that is equal to $0.12/kWh is applied throughout 24 hours. The operational cost of the pumping station is considered for a horizon of 20 years (for amortization purposes) and the present value of energy cost is based on an interest rate of 12%.

Table 6.3 Pump characteristics

Discharge (gpm)	Pump head (ft)	Efficiency (%) (wire to water)
0	300	0
2000	292	50
4000	270	65
6000	230	55
8000	181	40

Tanks

As in Farmani et al. (2005b), the option of building up to two new tanks in places which are not directly linked to the existing tanks or water source is considered. As a result, there are 16 possible locations for building new tanks, i.e., except at node 1, 14 and 17 in Figure 6.1. Tank capital costs are considered as a function of volume, and the standard sizes and associated costs are given in Table 6.4. The cost of a tank with an intermediate size is interpolated linearly. Tanks are not permitted to be smaller or larger than the acceptable limits (i.e., smaller than 50,000 Imperial gallons or greater than 1 million Imperial gallons). This is due to the financial consideration that a too small or large tank is not cost-efficient.

Table 6.4 Capital cost of new tank

Tank Volume (Imperial gallons)	Capital Cost ($)
50,000	115,000
100,000	145,000
250,000	325,000
500,000	425,000
1,000,000	600,000

Operational Requirement

Nodal Demand

To provide a more reliable solution, five different loading conditions are considered concurrently, i.e., average day demand, instantaneous peak demand as well as three fire flow demands. Each of them specifies a typical pattern of water use and accounts for the performance of network under different scenarios. The base demand of each node under various loading conditions is given in Table 6.5. The instantaneous peak demand is 1.8 times the average day demand; whereas each fire flow demand needs to supply the peak day flow which is 1.3 times the average day flow, as well as the flow required for fire fighting at the corresponding nodes. Demand variation (pattern) is specified in Table 6.6.

Table 6.5 Base nodal demand under five loading conditions

Node ID	Loading Conditions (unit: gpm)				
	Average day (24 hours)	Instantaneous peak (snapshot)	Fire Flow 1 (2 hours)	Fire Flow 2 (2 hours)	Fire Flow 3 (2 hours)
2	200	360	260	260	260
3	200	360	260	260	260
4	600	1080	780	780	780
5	600	1080	780	**1500**	780
6	600	1080	780	**1500**	780
7	600	1080	780	**1500**	780
8	400	720	520	520	520
9	400	720	520	520	520
10	400	720	520	520	520
11	400	720	520	520	**1000**
12	500	900	650	650	650
13	500	900	650	650	650
14	500	900	650	650	650
15	500	900	650	650	650
16	400	720	520	520	520
17	1000	1800	1300	1300	**1000**
18	500	900	650	650	650
19	1000	1800	**2500**	1300	1300

Note: It is assumed that only the required flows specified in the table above are needed under each fire flow condition. The fire-fighting places (boldfaced in the corresponding columns) are at node 19, node 5-7, and node 11 and 17 respectively.

Nodal Pressure

The pressure requirement of each node in the network varies according to different loading conditions. For the average day demand and the instantaneous peak demand, a minimum of 40 psi must be provided at each node. However, in the fire flow scenarios, the pressure at each node should be maintained above 20 psi.

Table 6.6 Water demand pattern throughout a day

Period	Demand Factor
6 p.m.-9 p.m.	1.0
9 p.m.-12 p.m.	0.9
12 p.m.-3 a.m.	0.7
3 a.m.-6 a.m.	0.6
6 a.m.-9 a.m.	1.2
9 a.m.-12 a.m.	1.3
12 a.m.-3 p.m.	1.2
3 p.m.-6 p.m.	1.1

Tank Operation

The two existing tanks have the same base elevation at 215 ft, and the minimum normal day level and the maximum level at 225 ft and 250 ft, respectively. This is equivalent to an effective capacity of 156,250 Imperial gallons for each tank. The water below the minimum normal day level is retained for emergency needs, giving a volume of 62,500 Imperial gallons for each tank. Tanks are required to be operated from the minimum normal day levels (at 6 p.m.) whatever loading condition is considered. Under average day demand, the capacity of each tank should be fully utilised, that is, it is required to reach the maximum and minimum levels at least once during a 24-hour simulation. Moreover, each tank needs to meet its minimum normal day level at the end of simulation. The retained capacity (water under the minimum normal day level) can only be used under fire flow demands.

The newly-built tanks should also be operated as specified for the existing tanks. In evaluation of fire flow scenarios, one pump is forced to be out of service which adds a further stressed condition to the system.

6.3.2. Problem Formulation

As introduced in Chapter 4, two objectives are considered in the problem formulation, i.e., minimising the total cost and maximising the network resilience. However, due to the existence of tank and pump components in the network, the main differences between the formulation of the Anytown problem from the one presented in Chapter 4 lie in the definitions of objectives, decision variables and constraints due to the operation of the Anytown network.

Objectives

The first objective is to minimise the total cost (see Eq. (6.1)), which consists of capital costs and operational costs. The capital cost includes the expenditure on pipes (i.e., new, duplicate or cleaning and lining ones) and new tanks (see Eq. (6.2)), whereas the operational cost is considered for an amortization period of 20 years with the interest rate fixed at 12%. A daily operational cost of the pumping station is computed based on a constant energy price which is equal to \$0.12/kWh. Note that the present value of the operational cost (see Eq. (6.3)) is used in the first objective.

$$\min C_{Total} = C_{Capital} + PV\,(C_{Operational}) \tag{6.1}$$

$$C_{Capital} = C_{Pipe}^{New} + C_{Pipe}^{Duplication} + C_{Pipe}^{Clean} + C_{Tank}^{New} \tag{6.2}$$

Where, C_{Total} - total costs; $C_{Capital}$ - capital costs; $C_{Operational}$ - operational costs; PV - the present value of operational costs in a horizon of 20 years; C_{Pipe}^{New} - the capital costs of new pipes (including risers for new tanks if any); $C_{Pipe}^{Duplication}$ - the capital costs of duplication pipes; C_{Pipe}^{Clean} - the capital costs of existing pipes which are to be cleaned and lined; C_{Tank}^{New} - the capital costs of new tanks if any.

$$PV\,(C_{Operational}) = 365 \times C_{Operational}^{Daily} \times \frac{[(1+0.12)^{20}-1]}{0.12 \times (1+0.12)^{20}} \tag{6.3}$$

Where, $C_{Operational}^{Daily}$ - the daily operational costs; PV - the present value of the operational costs.

The second objective is to maximise the minimum network resilience (I_n) during a 24-hour simulation under the average day flow condition. Eq. (6.4) shows the mathematical expression of I_n for the Anytown problem, which is adapted from

Eq. (4.2) given in Chapter 4 to include tanks as another source of power delivered into a network.

$$I_n = \frac{\sum_{j=1}^{nn} C_j Q_j (H_j - H_j^{\min})}{(\sum_{k=1}^{nr} Q_k H_k + \sum_{i=1}^{npu} \frac{P_i}{\gamma} + \sum_{t=1}^{nt} Q_t H_t) - \sum_{j=1}^{nn} Q_j H_j^{\min}} \tag{6.4}$$

Where, *nt* - number of tanks in the system; Q_t - amount of flow leaving tank *t*; H_t - hydraulic head at tank *t*. The other notations are the same as Eq. (4.2) presented in Chapter 4.

Decision Variables

Figure 6.2 shows the structure of a solution to the Anytown problem, which has a total of 77 decision variables organised in pipe, tank and pump sections. Variables 1 to 35 denote the decisions for existing pipes; including 'do-nothing', cleaning and lining, and duplication with a specific size (see Table 6.2). Variables 36 to 41 denote the decisions for new pipes, including the sizes specified in Table 6.2 and 'do-nothing' which implies that no pipe is laid here. The same decisions are applied to two risers of the new tanks. If a do-nothing option is chosen for a riser, the corresponding new tank will be disconnected from the system, indicating that it is not built. Variables 44 to 53 denote the decisions for two new tanks; of which each five-variable determines the location (i.e., which demand node the tank is linked with), operating levels (minimum normal day level, maximum level and tank bottom elevation) and tank diameter. Here, each tank is assumed to be cylindrical. For tank operating levels, it is worth noting that the randomly generated minimum level may exceed the maximum level. In this situation, these two values will be exchanged before being assigned to the corresponding decision variables. Variables 54 to 77 denote the number of pumps switched on at each time step in a 24-hour simulation horizon.

It is worth noting that this structure is much shorter than the one presented by Farmani et al. (2005b), which contained 112 variables for a solution. The differences (35 variables) between these two structures lie in how the pipe variables are dealt with. In the structure given by Farmani et al. (2005b), the first

35 decision variables were the high-level decisions for the existing pipes, which were to duplicate, to clean and line, or to leave these pipes. In contrast, the one adopted in this chapter merges these high-level decisions into the low-level decisions, which are the specific diameter options for the existing pipes. As a result, the search space size is significantly reduced, which is believed to help decrease the complexity of this problem to some extent.

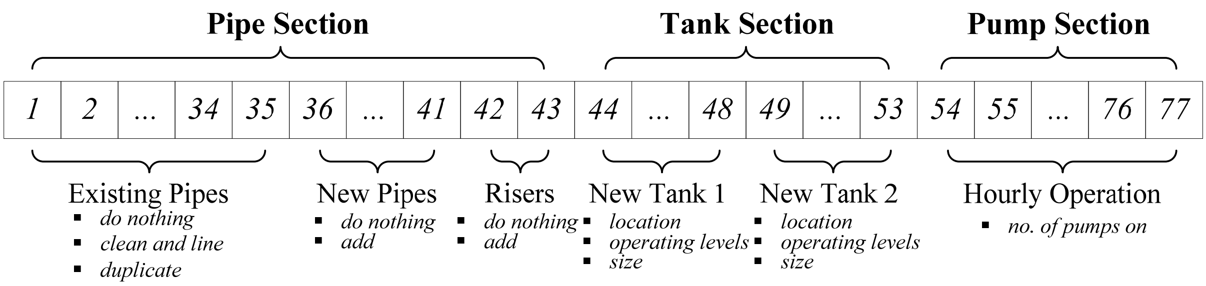


Figure 6.2 Structure of decision variables of a solution to the Anytown problem

Constraints

As introduced in Section 6.3.1, two kinds of explicit constraints are considered for the Anytown problem, that is, the minimum nodal pressure requirements for five loading conditions and the criteria for tank operation under the average daily flow scenario. Pressure deficiencies accumulated during the hydraulic simulations under various loading conditions and tank level differences during the 24-hour simulation are summed up to account for the total constraint violations. Mathematical expressions of pressure deficiencies and tank level differences are given in Eq. (6.5) and Eq. (6.6), respectively. A feasible solution must satisfy all the constraints under all loading conditions. The existence of these constraints is expected to introduce another level of complexity in addition to the vast search space.

$$NPD = \sum_{j=1}^{nl} \sum_{t=1}^{ns} \sum_{i=1}^{nn} \left| H_j^{i,t} - H_j^{\min} \right| \qquad if : H_j^{i,t} < H_j^{\min} \tag{6.5}$$

Where, *NPD* - total nodal pressure deficiencies; *nl* - number of loading conditions; *ns* - number of time steps under loading condition *j*; *nn* - number of demand nodes; $H_j^{i,t}$ - pressure at demand node *i* at time step *t* under loading condition *j*; $H_j^{\min}$ - minimum nodal pressure requirement for loading condition *j*.

$$TLD = \sum_{i=1}^{nt}\left(\left|h_i^{\min} - H_i^{\min}\right| + \left|h_i^{\max} - H_i^{\max}\right| + \left|h_i^{end} - h_i^{start}\right|\right) \quad if: h_i^{end} \neq h_i^{start} \tag{6.6}$$

Where, *nt* - number of tanks in the system; $h_i^{\min}$ and $h_i^{\max}$ - minimum and maximum operation levels of tank *i* during 24 hours, respectively; $H_i^{\min}$ and $H_i^{\max}$ - minimum and maximum boundary levels of tank *i*, respectively; h_i^{end} and h_i^{start} - end and initial levels of tank *i*, respectively.

6.4. Experimental Setup

This formulation is a challenging design problem, and it was very difficult to obtain the reference set of the Pareto-optimal front of the Anytown problem. Unlike other benchmark problems archived in Chapter 4, the Anytown problem introduces various types of decision variables and design criteria, leading to a highly constrained search space with a great number of local optima. In fact, several preliminary runs were carried out using MOEAs with various combinations of population sizes and numbers of function evaluations (NFEs). It is found that the Pareto front obtained was quite sensitive to the quality of the initial population as well as the random seed, which in turn reveals that the Anytown problem is a more complicated design case compared with those presented in Chapter 4.

On the other hand, even with a computational budget of up to 4 million NFEs, the MOEAs under consideration did not show an evident sign of convergence. Quite the opposite was found, in that the Pareto fronts obtained in multiple runs by various methods were dispersed over the objective space. Consequently, it is currently impossible to compare their ultimate or dynamic performances quantitatively, using indicators introduced in Chapter 5 due to the lack of a consistent reference set. Furthermore, it is still unknown what a suitable NFE is as the stopping criterion for the comparison purposes. Therefore, the computational budget is determined by choosing a suitable NFE according to the previous work for the Anytown problem.

6.4.1. Performance Assessment

To provide a quantitative comparison between the Pareto fronts obtained by GALAXY and the other MOEAs, a binary coverage indicator (denoted as I_C) is adopted to assess the relationship between the approximation sets (Zitzler 1999). The I_C is mathematically formulated in Eq. (6.7).

$$I_C(A,B) = \frac{\left|\{b \in B \mid \exists a \in A : a \succ= b\}\right|}{\left|B\right|} \tag{6.7}$$

Where, *A* and *B* - two Pareto approximation sets for comparison purposes; *a* and *b* - any non-dominated solution in set *A* and *B*, respectively; $\left|B\right|$ - the cardinality of set *B*. $a \succ= b$ means that solution *a* weakly dominates solution *b* in terms of objective function values (i.e., *a* is not worse than *b* in all objectives).

Note that Eq. (6.7) maps the ordered pair (*A, B*) to the interval [0, 1]. $I_C(A,B) = 1$ means that all the solutions in set B are weakly dominated by at least one solution in set A. On the contrary, $I_C(A,B) = 0$ suggests the situation when none of solutions in set B are weakly dominated by any solution in set A. It is worth mentioning that $I_C(A,B)$ is not necessarily equal to $1 - I_C(B,A)$. Therefore, the binary I_C can adequately measure the relative quality of two approximation sets identified by GALAXY and its competitors. As in Zitzler et al. (2003), the binary I_C indicator complies with the Pareto-dominance relationship in terms of compatibility and completeness. Such criteria are specified in Table 6.7.

The EAF tool (López-Ibáñez et al. 2010) is not considered for comparing the approximation fronts obtained by GALAXY and the other MOEAs in the objective space. The reason is that the EAF tool is very useful to illustrate where and how much one algorithm is better than the other when both algorithms converge properly for a specific problem, whereas this condition is not satisfied for the Anytown problem. Instead, the Pareto fronts identified by all the MOEAs in a single run are plotted in the objective space to provide a straightforward comparison.

Table 6.7 Interpretation of the results of binary I_C

Results of Binary I_C	Relationship	Interpretation
N/A	$A \succ\succ B$	Every solution *b* in *B* is strictly dominated by at least one solution *a* in *A* (*a* is better than *b* in all objectives).
$I_C(A,B)=1$ $I_C(B,A)=0$	$A \succ B$	Every solution *b* in *B* is dominated by at least one solution *a* in *A* (*a* is better than *b* in at least one objective and is not worse than *b* in the other objectives).
$I_C(A,B)=1$ $I_C(B,A)<1$	$A \rhd B$	Every solution *b* in *B* is weakly dominated by at least one solution *a* in *A* and $A \neq B$.
$I_C(A,B)=1$	$A \succeq B$	Every solution *b* in *B* is weakly dominated by at least one solution *a* in *A* (*a* is not worse than *b* in all objectives).
$I_C(A,B)=1$ $I_C(B,A)=1$	$A = B$	*A* and *B* are identical.
$0<I_C(A,B)<1$ $0<I_C(B,A)<1$	$A \parallel B$	*A* and *B* are incomparable.

6.4.2. Computational Budget

A summary of the literature survey is given in Table 6.8 in order to find out how much effort in terms of NFEs, population size and number of runs was made in solving the Anytown problem under different formulations. It is revealed from this table that most work on the Anytown problem used NFEs between 0.2 million to 1.5 million, except for Vamvakeridou-Lyroudia et al. (2005; 2006) which addressed the problem using a fuzzy reasoning framework. It is worth noting that some papers (Olsson et al. 2009; Fu et al. 2012a) formulated the problem as unconstrained multi-objective optimisations, in which the constraint violations were treated as the second objective. Although it is practical to deal with a real-world design problem under such a formulation, it is unable to challenge an algorithm's ability to handle a difficult problem with massive constraints (like the two-objective constrained formulation given in Section 6.3.2).

If the problem is formulated in a similar way to that described in this chapter, the computational budget is around 0.5 million NFEs and fewer than 5 runs are used, as shown in Table 6.8. Therefore, a uniform budget equal to 1 million NFEs is adopted for each algorithm and 10 independent runs are implemented to alleviate the impact of randomness. Moreover, the initial population for each run is generated in advance and used to initiate each algorithm, which guarantees that they start search from the same location. Hopefully, this can facilitate a better comparison of these algorithms for such a complicated design problems.

Table 6.8 Computational effort spent on the Anytown rehabilitation problem

Papers	Computational Budget		
	NFEs	Population Size	Runs
(Murphy et al. 1994)	N/A	N/A	N/A
(Walters et al. 1999)	50,000	N/A	3
(Farmani et al. 2005b)	500,000	100	N/A
(Vamvakeridou-Lyroudia et al. 2005)	150,000-5,000,000	30-100	N/A
(Farmani et al. 2006)	500,000	100	N/A
(Vamvakeridou-Lyroudia et al. 2006)	2,000,000	50	N/A
(Prasad and Tanyimboh 2008)	200,000-1,500,000	100-300	N/A
(Olsson et al. 2009)	200,000	200-2,000	5
(Fu et al. 2012a)	1,000,000	*adaptive	10
(Wang et al. 2014b)	500,000	100	10

Note: *ε-NSGA-II employed in Fu et al. (2012a) allowed the population size to adapt in search according to the number of best solutions found so far in the archive. N/A indicates that the relevant information cannot be found in the corresponding papers.

Admittedly, the computational budget adopted herein may not be sufficient to guarantee that all the MOEAs considered converge well at the end of search, which in turn may affect the conclusions drawn from the results. However, when dealing with a real-world design problem, it is a desired feature of an algorithm to quickly identify solutions of high quality with a low computational budget. Thus, the comparison of various algorithms under the selected computational budget is still meaningful from a practical perspective.

6.4.3. Algorithmic Setup

The same strategy of specifying the algorithmic parameters as mentioned in Chapter 5 is implemented. In particular, a unified population size equal to 100 is adopted for each MOEA for the Anytown problem. The individual parameters of each MOEA follow the settings presented in Table 5.2 in Chapter 5.

Additionally, for the ε-dominance based MOEAs, it is required to determine the appropriate ε precision for each objective. Therefore, a series of trial runs (10 for each pair of ε setting) were conducted using the Borg algorithm given different combinations of ε precision. More specifically, six different pairs of ε were considered, that is, ($\varepsilon_{\cos t}, \varepsilon_{I_n}$)=(0.1, 0.01), (0.01, 0.001), (0.001, 0.0001), (0.001, 0.001), (0.0001, 0.0001) and (0.00001, 0.00001). It was found that Borg with ($\varepsilon_{\cos t}, \varepsilon_{I_n}$)=(0.001, 0.001) generally outperformed those with the other settings by obtaining a series of relatively consistent Pareto fronts in the objective space. Therefore, this setting was applied to both Borg and ε-MOEA.

It is worth emphasising that it is impossible to estimate the number of solutions found in the archive beforehand. Using a larger ε precision (coarse cells in the objective space) returns fewer solutions, and using a smaller ε precision results in more solutions in the archive. In the trial runs, the Borg always reported solutions fewer than the population capacity even using the smallest ε values (i.e., both equal to 0.00001).

On the other hand, the ε precision cannot only affect the distribution of the approximation set but also the convergence of the approximation set towards the Pareto-optimal front. For example, using smaller ε precision seemed to improve the convergence dramatically for some runs. However, an opposite effect could also be observed, thus leading to inconsistent approximation fronts.

6.5. Results and Discussion

In this section, a comparison between GALAXY and the other MOEAs in terms of ultimate performance is presented. The dynamic performances of these MOEAs are not taken into consideration due to the lack of a reference set of the best-known Pareto front. In addition, the dynamic variations of search operators of hybrid MOEAs are also analysed to reveal their internal performances for the Anytown problem.

Since there is no guarantee that the MOEAs can converge given 1 million NFEs for such a highly constrained design problem, the solutions found by each method are carefully checked. More specifically, only feasible solutions are eligible for comparison purposes, and the dominated solutions which are not in the first front for the Pareto-dominance based MOEAs are removed from the approximation set.

6.5.1. Ultimate Performance

The binary coverage indicator (see Section 6.4.1) is used to provide a quantitative comparison between GALAXY and its competitors. The results of I_C for 10 individual runs are presented in Table 6.9. It is clear that GALAXY convincingly outperformed the ε -dominance based MOEAs by weakly dominating majority of the solutions obtained by those MOEAs. However, there is a tie between GALAXY and the other Pareto-dominance based MOEAs. It scores less I_C compared with AMALGAM and NSGA-II on average, but note that this interpretation can be very sensitive to the shape and cardinality of approximation fronts in the objective space.

Table 6.10 shows the interpretations from the viewpoint of the relationship of approximation fronts (see Table 6.7) for a single run. The fronts obtained by GALAXY dominated those by AMALGAM for 3 runs (run 1, 4, and 10) out of 10, but was dominated by AMALGAM for 2 runs (run 2 and 6). For 3 runs, the fronts reported by both algorithms were incomparable. In addition, they both dominated a subset of solutions by its counterpart once out of 10 runs.

In comparison between GALAXY and NSGA-II, the fronts obtained by the latter dominated those obtained by the former for 4 runs (run 2, 3, 6 and 9), and the former only dominated the front by the latter for 1 run (run 4). For 4 runs, the fronts reported by both algorithms were incomparable. In addition, GALAXY dominated a subset of solutions by the NSGA-II once at run 5.

Table 6.9 Comparison of binary I_C in percentage between GALAXY and the other MOEAs

Algorithm		I_C (%)										
		Run No.										Mean
A	*B*	1	2	3	4	5	6	7	8	9	10	
GALAXY	AMALGAM	100	0	10	100	39	0	1	0	25	100	37.5
		0	100	77	0	0	100	36	85	8	0	40.6
	Borg	100	100	88	100	100	14	100	36	0	100	73.8
		0	0	8	0	0	72	0	85	31	0	19.5
	NSGA-II	60	0	0	100	96	0	40	16	0	44	35.6
		2	100	100	0	0	100	15	80	100	2	49.8
	ε-MOEA	100	100	100	100	100	100	0	100	100	100	90.0
		0	0	0	0	0	0	10	0	0	0	1.0

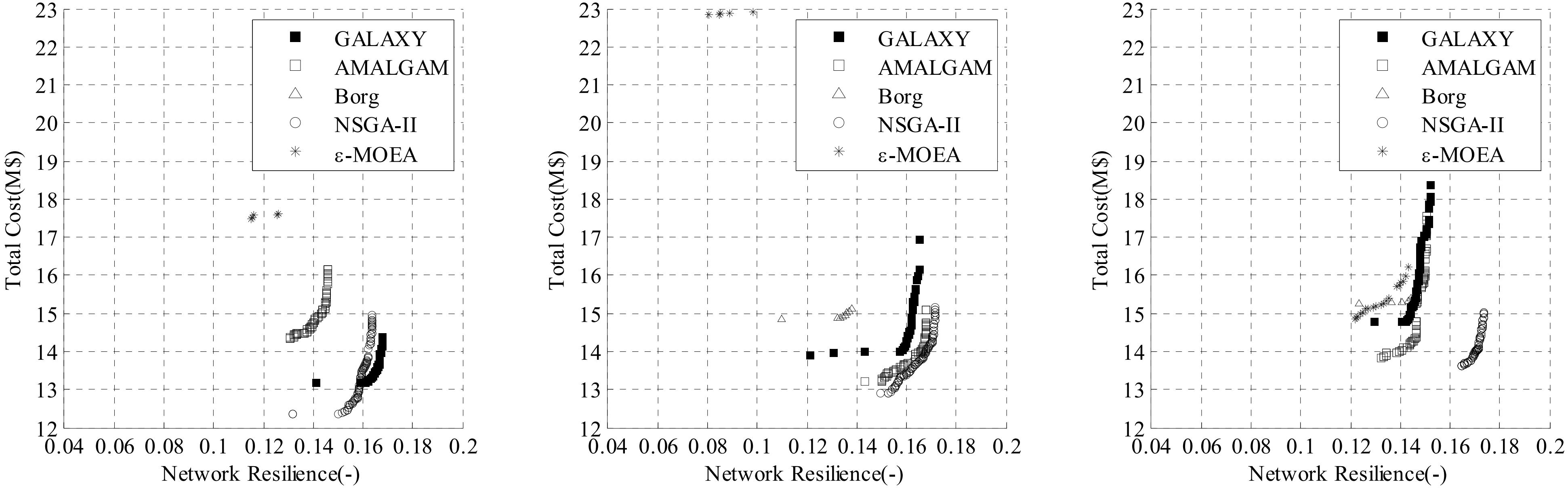
Note: For each comparison between algorithm *A* and *B*, both $I_C(A,B)$ and $I_C(B,A)$ were calculated and shown in the top and bottom cells of each run, respectively.

Table 6.10 Comparison of approximation sets in terms of dominance relationship

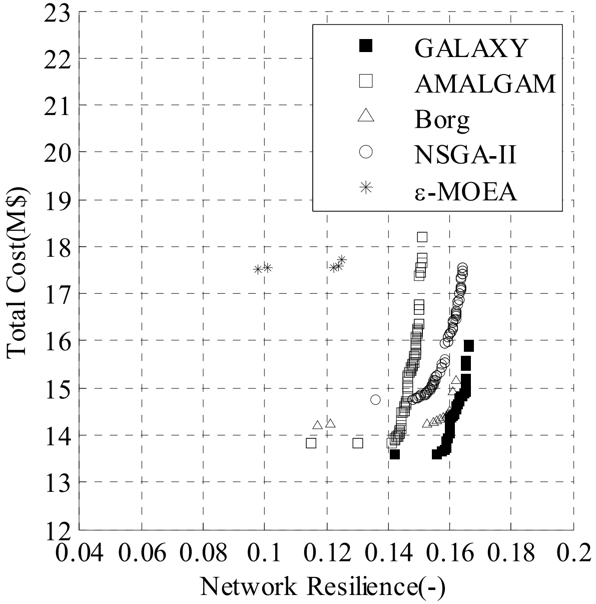
Algorithm		No. of Occurrences in 10 runs				
A	*B*	$A \succ B$	$B \succ A$	$A \parallel B$	$A \succ subset(B)$	$B \succ subset(A)$
GALAXY	AMALGAM	3	2	3	1	1
	Borg	6	0	3	0	1
	NSGA-II	1	4	4	1	0
	ε-MOEA	9	0	0	0	1

Note: $A \succ subset(B)$ means that every solution *b* in a subset of *B* is weakly dominated by at least one solution *a* in *A*, and every solution *a* in *A* is not dominated by any solution *b* in *B*.

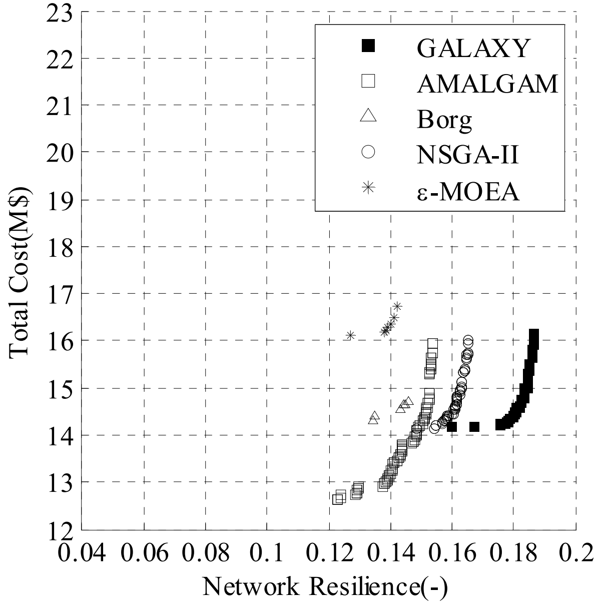
Figure 6.3 illustrates the approximation fronts obtained by all the MOEAs for each run. Note that all the sub-figures were adjusted to reside in the same range of the objective space. It can be seen from these sub-figures that the performance of each MOEA was not as consistent as it was for solving the benchmark problems presented in Chapter 5. This again implies that the Anytown problem still represents a very challenging multi-objective design problem in the community. In most runs, the approximation fronts obtained by all the MOEAs resided in the range of the total cost (in M$) between (12, 18) and the network resilience between (0.12, 0.18). Surprisingly, GALAXY successfully reported a number of solutions with the network resilience greater than 0.18 at run 5, in which it convincingly dominated the fronts (or a subset of the front) obtained by the other MOEAs. In addition, the fronts obtained by GALAXY generally demonstrated a well-distributed feature compared with those by its competitors.



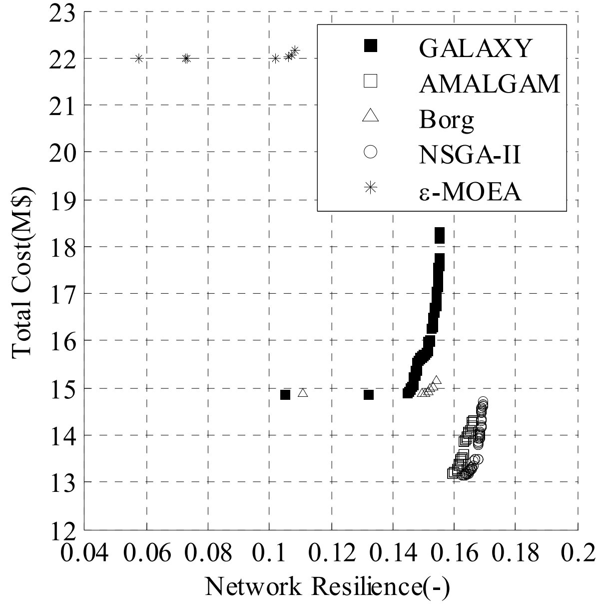
(a) Run 01 (b) Run 02 (c) Run 03



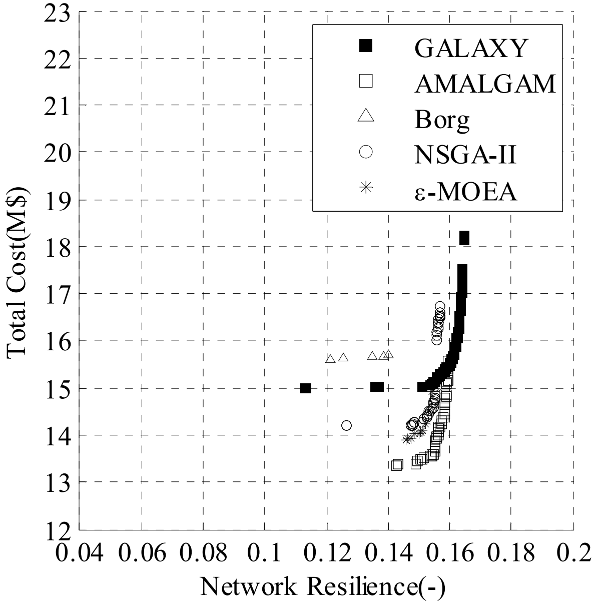
(d) Run 04



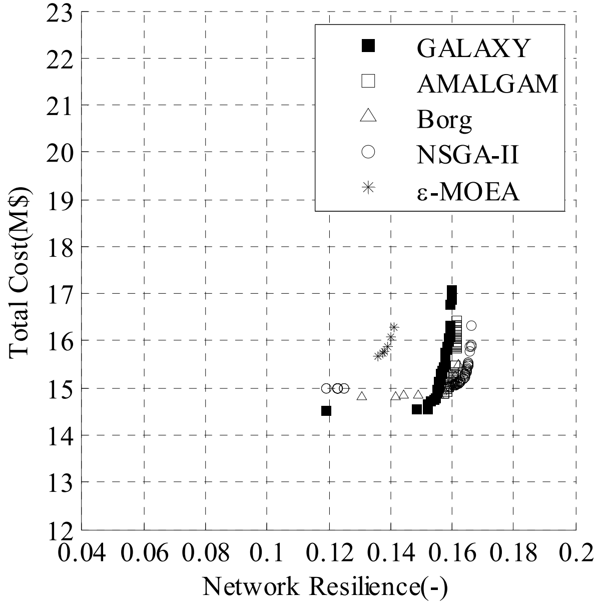
(e) Run 05



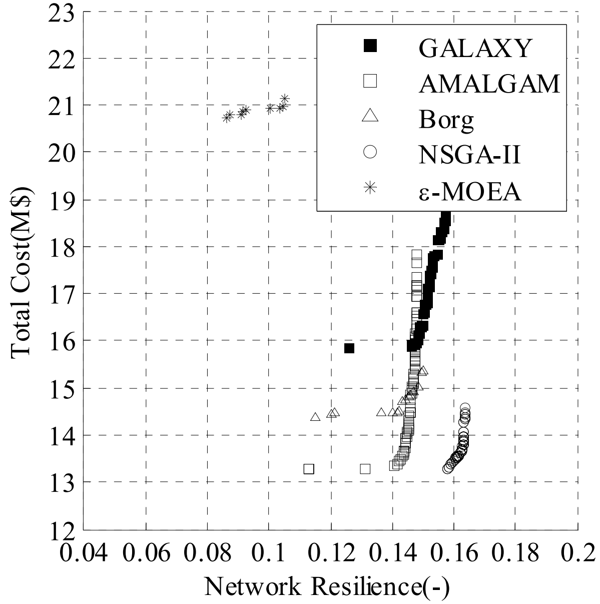
(f) Run 06



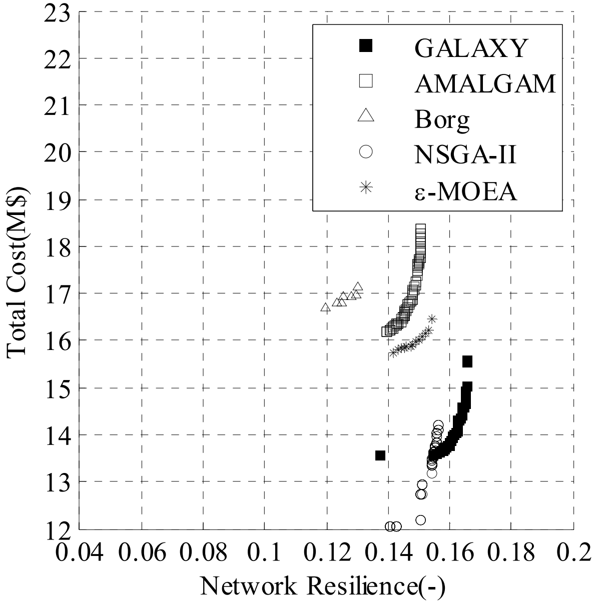
(g) Run 07



(h) Run 08



(i) Run 09



(j) Run 10

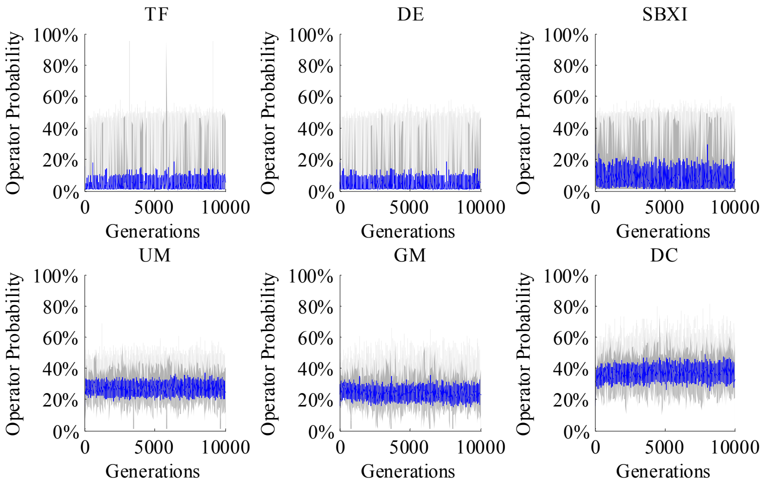
Figure 6.3 Comparison of the approximation front obtained by each MOEA for each run

6.5.2. Dynamic Variations of Search Operators

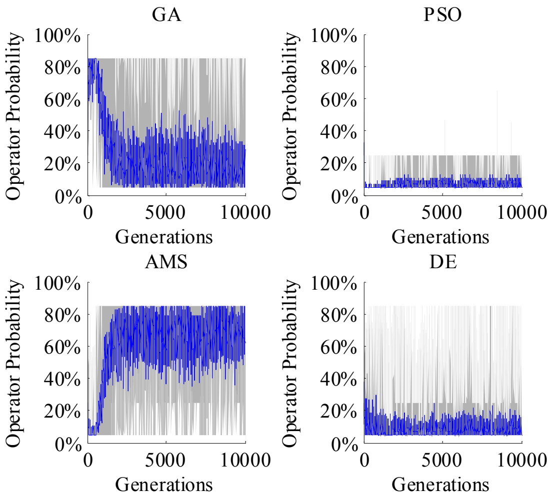
Figure 6.4 provides a comparison of the dynamic variation of search operators of hybrid algorithms. For GALAXY, each operator was utilised consistently throughout the search. However, this variation did not show similar patterns as seen for the benchmark design problems presented in Chapter 5. In other words, the operators with better leaping ability in the global sense (e.g., TF, DE) did not lead the population effectively at the initial stage. On the contrary, the operators with better leaping ability in the local sense (e.g., UM, GM and DC) consistently outperformed the others throughout the search. This implies that for such a highly constrained problem, local search was more likely to steer the population towards the feasible region and eventually result in a better approximation front.

For AMALGAM, AMS turned out to be the most suitable operator compared with the other ones. GA was the dominant operator at the beginning of search; however its contribution to the population was gradually overtaken by AMS after about 2,000 generations. DE and PSO did not perform very well compared with GA and AMS. This variation was very similar to the patterns when AMALGAM was applied to solve the benchmark problems with the number of pipes between 30 and 58. One probable reason is that AMS was particularly useful for solving the small to intermediate sized problems (normally less than 100 decision variables).

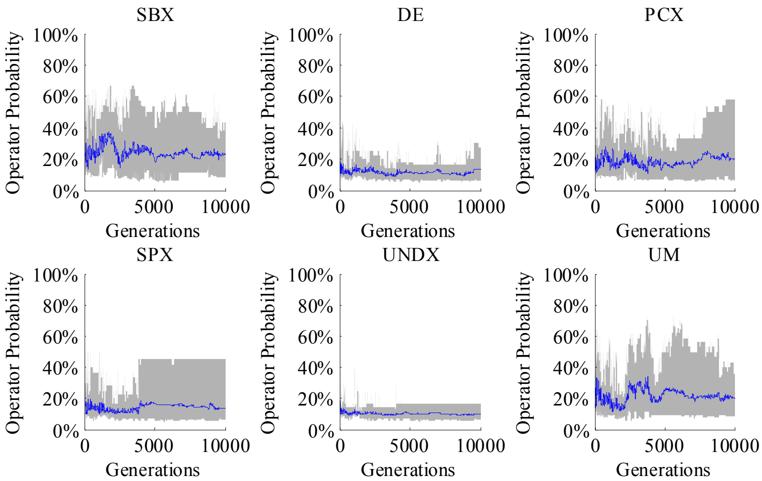
For Borg, it is relatively hard to summarise a clear pattern of six operators throughout the search. Generally speaking, SBX, PSX, SPX and UM were more frequently employed compared to DE and UNDX. This may be attributed to the fact that their performances were relatively sensitive to the quality of initial population, which differed from the operators within GALAXY and AMALGAM.



(a) GALAXY



(b) AMALGAM



(c) Borg

Figure 6.4 Dynamic variations of search operators of hybrid algorithms for the ATN problem

6.5.3. Impact of the Quality of the Initial Population

It was observed that for the ε-dominance based MOEAs, in most of computational times, they suffered from the failure to find feasible solutions. For example, Borg did not report any feasible solution at the end of run 1. In contrast, the Pareto-dominance based MOEAs were able to jump out of the infeasible region effectively at the initial stage of search, which in turn ensured that better solutions could be found in the subsequent search. The poor performance of the ε-dominance based MOEAs for the Anytown problem may be attributed to the steady-state algorithmic structure which restricted the population to efficiently get rid of infeasible solutions.

On the other hand, since the initial population was unified for each MOEA in the experiment, it is interesting to investigate the impact of the quality of the initial population on the performance of these MOEAs. To this end, the initial populations were analysed in terms of constraint violations. Because the majority of solutions in the initial population had constraint violations on the order of magnitude of 10^8 to 10^9, the quality of the top five individuals as well as all the individuals on average are shown in Table 6.11. It can be seen from the table below that run 5 and run 8 had better quality initial populations compared to the other runs, and run 4 had the worst quality across 10 runs. Therefore, MOEAs were expected to obtain better results in these two runs. This anticipation may be true if comparing the approximation fronts by various MOEAs in run 4 and run 5. Results obtained in run 5 were indeed better than those obtained in run 4 except for the Borg algorithm. However, it is not always the case if ten runs are considered together. This fact suggests that the quality of initial population did not have a substantial impact on the final achievement of MOEAs for the Anytown problems.

Table 6.11 Quality of initial population in terms of constraint violations

Runs	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Mean Value	
						Top Five	All
Run01	-4.9E+03	-8.9E+03	-6.9E+08	-1.1E+09	-1.1E+09	-5.8E+08	-5.0E+09
Run02	-2.8E+04	-9.9E+08	-1.1E+09	-1.1E+09	-1.1E+09	-8.5E+08	-5.1E+09
Run03	-3.5E+03	-7.8E+03	-3.2E+04	-8.9E+08	-9.9E+08	-3.8E+08	-4.9E+09
Run04	-1.1E+09	-1.1E+09	-1.2E+09	-1.2E+09	-1.3E+09	-1.2E+09	-5.2E+09
Run05	-2.2E+03	-2.7E+03	**-3.8E+03**	**-1.7E+04**	-9.9E+08	-2.0E+08	**-4.5E+09**
Run06	-4.1E+03	-6.9E+08	-9.9E+08	-9.9E+08	-1.2E+09	-7.7E+08	-4.8E+09
Run07	**-5.8E+02**	-6.9E+08	-9.9E+08	-1.1E+09	-1.2E+09	-7.9E+08	-4.7E+09
Run08	-6.1E+02	**-7.7E+02**	-5.1E+03	-2.0E+04	**-8.9E+08**	**-1.8E+08**	-4.8E+09
Run09	-3.8E+03	-6.9E+08	-1.1E+09	-1.1E+09	-1.1E+09	-7.9E+08	-5.0E+09
Run10	-1.3E+03	-8.9E+08	-1.1E+09	-1.2E+09	-1.2E+09	-8.7E+08	-4.9E+09

Note: The minimum constraint violations in each column is shown in bold.

6.6. Summary and Conclusions

This chapter applied the GALAXY method and its competitors to solve the Anytown rehabilitation problem, which represents one of challenging design cases in the water community for taking the operation of pumps and tanks into consideration. Firstly, the Anytown network was described from a historical viewpoint followed by a mathematical optimisation problem formulation. Then, due to the lack of a reference set for this design problem, the binary coverage indicator was introduced to provide a quantitative way for comparing the approximation fronts obtained by different algorithms. In addition, the computational budget and algorithmic setup were justified. Finally, GALAXY was compared with the other MOEAs considered from the viewpoint of ultimate performance using the binary coverage indicator as well as in the objective space. For hybrid algorithms, the dynamic variations of search operators were also taken into account and discussed. Moreover, the impact of the quality of initial population on the performances of various MOEAs for such a highly constrained design problem was also investigated.

The main conclusions drawn from this chapter are as follows:

- Generally speaking, the Pareto-dominance based MOEAs (i.e., AMALGAM and NSGA-II) and GALAXY outperformed the ε-dominance based approaches for dealing with the Anytown rehabilitation problem. There was a tie between GALAXY and the Pareto-dominance based MOEAs. In other words, no one method was consistently better than the others and their performances were sensitive to the quality of initial population and random seed.

- From the viewpoint of dynamic variations of search operators, GALAXY demonstrated a different pattern from those for solving the benchmark design problems as shown in Chapter 5. In particular, the operators with better leaping ability in the global sense did not lead the search at the initial stage. On the contrary, the operators which are good at leaping in the local sense consistently made major contributions to the population, which may account for the fact that it was not superior to the other MOEAs. It also implies that the search space of the Anytown problem was probably far more complex than those of benchmark design problems.
- The failure of Borg and ε-MOEA can be attributed to the use of the ε-dominance concept, which was not practical in solving the Anytown rehabilitation problem. It might cause ‘genetic drift’ during search, which resulted in solutions that were concentrated in a small area of search space (i.e., poor performance in terms of diversity and spread). On the other hand, one advantage of using the ε-dominance concept is to provide an effective way to control the precision of results. For the total cost objective, it may be true to specify such an ε precision according to the preference of decision makers as the range of this objective is relatively easy to estimate. However, for the network resilience, it is impossible to know the rough range a priori. In this situation, an arbitrary setting of ε precision may lead to an unsatisfactory Pareto front or even failure. Furthermore, the time spent on selecting the best ε precision for a more complex design problem is probably unacceptable in practice. In contrast, GALAXY and the traditional Pareto-dominance based MOEAs do not require the specification of ε precision and are more suitable for solving the discrete design problems.

- Finally, the quality of the initial population can greatly affect the performances of various MOEAs for the Anytown problem. However, this point does not mean that better initial populations will necessarily lead to a better approximation front at the end of the search.

7. SUMMARY, CONCLUSIONS AND FUTURE WORK RECOMMENDATIONS

7.1. Summary

7.1.1. Summary of the Thesis

The main objective of this thesis is to develop the theoretical basis and implement an enhanced hybrid framework for the two-objective design or extended design of WDSs. Its performance is verified and compared with state-of-the-art MOEAs for a wide range of benchmark problems in terms of ultimate and dynamic performance.

A novel hybrid algorithm, termed GALAXY, has been developed and presented in this thesis. Its main features are summarised as follows:

1. It is based on the generational MOEA algorithmic framework, in which the population gets updated after the offspring are created.
2. It combines the traditional Pareto-dominance concept and the ε-dominance concept for population replacement.
3. It adopts the integer coding scheme, which supports a straightforward representation of decision variables for discrete design problems.
4. It employs six search operators organised in a high-level teamwork hybrid structure for generating the offspring, including turbulence factor, differential evolution, simulated binary crossover for integers, uniform mutation, Gaussian mutation and dither creeping.
5. Multiple search operators are applied simultaneously and the number of individuals that each operator is allowed to produce is determined according to its contribution to the current population.
6. Duplicate solutions are checked and removed from the population after the offspring and current population are merged.
7. It eliminates the need for fine-tuning of individual parameters, which is time-consuming and often results in suboptimal performance.

An archive of benchmark design or extended design problems of WDSs is established, termed WDSBA. Optimal design is formulated as a constrained two-objective optimisation problem, minimising the total capital cost and maximising the network resilience, which is a compact and surrogate measure of network reliability. WDSBA encompasses twelve benchmark problems collected from the literature that are categorised into four groups, i.e., small, medium, intermediate and large, according to the search space sizes.

The proposed GALAXY method is verified and compared to four state-of-the-art MOEAs, including two hybrid algorithms (i.e., Borg and AMALGAM) and two baseline algorithms (i.e., NSGA-II and ε-MOEA), for solving the benchmark problems in the WDSBA. To obtain the best-known Pareto front of each benchmark problem as a reference set, these MOEAs were implemented given an extensive computational budget. Both ultimate and dynamic performances of these algorithms are considered by means of four numerical indicators and the EAF based graphical tool. For hybrid algorithms, the dynamic variations of search operators are also illustrated and compared.

In addition, GALAXY and other MOEAs are applied to solve the challenging Anytown rehabilitation problem. The rehabilitation problem is formulated in the similar way as the benchmark problems in the WDSBA, but considering design and operations of the system simultaneously. The performances of these MOEAs are analysed and compared by plotting the approximation fronts in the objective space and by using the binary coverage indicator.

7.1.2. Summary of the Contributions

The main contributions of this thesis regarding the hybrid optimisation framework are summarised as follows:

1. A novel hybrid framework for solving the discrete two-objective design or extended design of WDSs.
2. A novel perspective of selecting search operators for the hybrid framework according to their leaping abilities in the decision variable space.

3. Six modified search operators dedicated for the discrete design problems of WDSs that eliminate the need for fine-tuning of majority of individual parameters.
4. A new, specific way of handling duplicate solutions that significantly improves the performance of MOEAs in terms of ε-performance.
5. A new, hybrid usage of the traditional Pareto-dominance and the ε-dominance concepts for achieving a better balance between convergence and diversity.

The main contributions of this thesis regarding the benchmark problems are summarised as follows:

1. Creation of an archive of benchmark problems related to the design or extended design of WDSs, including twelve cases ranging from small to large sized optimisation problems.
2. The best-known Pareto front for each benchmark problem, which can influence the future work of others in water research community.

The main contributions of this thesis regarding the performance assessment are summarised as follows:

1. A comprehensive use of a number of numerical indicators for evaluating particular aspects of multi-objective optimisation (i.e., convergence, diversity and consistency) when a reference set is available for a specific benchmark problem.
2. Consideration of both ultimate and dynamic performances of MOEAs for the benchmark problems which enables an in-depth analysis and avoids bias effectively.
3. A novel use of the EAF graphical tool to facilitate the direct comparison of the approximation fronts obtained by GALAXY and other MOEAs.
4. A novel combined use of the binary coverage indicator and approximation front plots (in the objective space) to compare MOEAs quantitatively and intuitively for the Anytown rehabilitation problem, for which the reference set is difficult to obtain beforehand.

7.2. Main Conclusions

The main conclusions regarding the hybrid algorithm for solving the two-objective design of WDSs are summarised as follows:

1. A novel hybrid algorithm, termed GALAXY, has been developed in this thesis, featuring a generational framework, a hybrid usage of Pareto-dominance and ε-dominance concepts, an integer coding scheme, six search operators deployed based on their leaping abilities in the search space and utilised in a high-level teamwork hybrid manner. GALAXY is actually a hybrid framework, rather than an algorithm, which can be further extended (or improved) by incorporating new search operators and strategies. This algorithm was shown to be superior to the state-of-the-art MOEAs for the majority of benchmark problems, according to the numerical indicators and the EAF tool.
2. GALAXY is a dedicated optimiser for solving the two-objective design of WDSs, and only involves two general parameters which are common to the majority of MOEAs, that is the population size and the number of function evaluations. In other words, there is no individual parameter required for fine-tuning before its application. GALAXY was demonstrated to be a very effective and efficient tool for dealing with real-world cases. This characteristic is achieved by deploying search operators which are tailored for discrete design problems.
3. The essential criterion of selecting multiple search operators according to their leaping ability in the decision variable space for hybridisation is important to GALAXY, because it effectively leads to a near-optimal search process in terms of exploration and exploitation. In particular, the operators which are good at leaping in the global sense are mainly employed at the initial stage of search, driving the population towards the near-optimal region; whereas the operators which are good at leaping in the local sense are mainly used in the subsequent generations, refining the current solutions to approach the Pareto-optimal front.
4. Several strategies inside the GALAXY method ensure its performance in terms of convergence, diversity and consistency on a wide range of

benchmark problems. In particular, the strategy that avoids duplicates and the new hybrid replacement procedure effectively accommodate and maintain current best solutions and increase the chance for exploring potential solutions in search space. They are of great importance for solving discrete design problems and resulted in a much better achievement of GALAXY in terms of ε-performance. Note that these strategies are not limited to GALAXY and can be implemented in other MOEAs to further enhance their performances.

The main conclusions regarding the comparison of GALAXY with other MOEAs for solving the benchmark problems in WDSBA are summarised as follows:

1. Generally speaking, GALAXY was superior to other state-of-the-art MOEAs considered in this thesis in terms of both ultimate and dynamic performances for majority of benchmark problems. This was demonstrated via several numerical indicators as well as the EAF graphical tool. It effectively captured boundary solutions (in both low and high cost regions) and consistently maintained a wide spread of solutions in the objective space. In addition, the behaviour of search operators within GALAXY were as expected, that is, the TF, DE, and SBXI contributed more individuals during the exploration phase, whereas the UM, GM and DC were the dominant operators during the exploitation phase.
2. GALAXY was convincingly outperformed by the Borg and AMALGAM for the EXN problem which was probably attributed to its weakness of dealing with infeasible solutions. However, note that none of the algorithms fully converged for the comparison, which implies that GALAXY still has a chance to perform better than these MOEAs provided that a greater computational budget is allocated.
3. The combination of many numerical indicators and the EAF graphical tool provided a better way to evaluate the performances of MOEAs from different perspectives comprehensively.

The main conclusions regarding the comparison of GALAXY with other MOEAS solving the Anytown rehabilitation problem are summarised as follows:

1. GALAXY and the Pareto-dominance based MOEAs (i.e., AMALGAM and NSGA-II) generally outperformed the ε-dominance based approaches for the Anytown rehabilitation problem. However, there was a tie between GALAXY and the Pareto-dominance based MOEAs and their performances were sensitive to the quality of the initial population and the random seeds used.
2. GALAXY demonstrated a different pattern of search operator dynamics compared to those shown for the benchmark problems. More specifically, TF, DE and SBXI did not lead the search at the initial stage. Rather, UM, GM and DC, which are good at leaping in the local sense, consistently dominated the other operators by contributing majority of offspring. This implied that the operators with good capability of leaping in the global sense were not suitable for such a highly-constrained problem, and it also partially explained why GALAXY cannot exceed the other MOEAs on the Anytown problem.
3. The failure of Borg and ε-MOEA on this problem can be partially attributed to the usage of the ε-dominance concept, which may cause 'genetic drift' during search, leading to the loss of boundary solutions. However, a proper setting of ε precision for certain objective (e.g., the network resilience considered in this thesis) is probably unrealistic, which in turn may result in an unsatisfactory Pareto front or even a failure (i.e., no feasible solution found at the end of search). Furthermore, the time spent on determining the suitable ε setting is probably unacceptable for large and/or complex cases (e.g., the Anytown problem); in contrast, GALAXY, which does not rely on the ε precision, is more practical (i.e., saving time and maintaining the diversity of solutions) in such a situation.
4. The quality of the initial population in terms of constraint violation can greatly affect the performances of various MOEAs; however, this does not mean that better initial population can necessarily lead to better approximation front at the end of search.

7.3. Recommendations for Future Research

Regarding hybrid frameworks for solving the multi-objective design of WDSs, recommendations for future research work are summarised as follows:

1. To establish a quantitative analysis framework which can evaluate the capability of a given search operator in terms of exploration and exploitation.
2. To incorporate more efficient search operators, for instance the multi-parent operators (PCX, SPX and UNDX) as used in Borg, into the framework of GALAXY.
3. To extend the current GALAXY method to deal with many-objective design of WDSs (Fu et al. 2012a) by involving more advanced dominance concept, like grid-dominance concept (Yang et al. 2013), for *Selection* and *Replacement*.
4. To develop a novel constraint handling strategy that can improve the performance of GALAXY in the region of low cost, where the Pareto-optimal solutions usually lie on the boundary between the feasible and infeasible regions.

Regarding the benchmark problems which can be included in the WDSBA, recommendations for future research work are summarised as follows:

1. To generate and archive more benchmark networks, especially those derived from real-world WDSs, such as the D-Town network used in the Battle of the Water Networks II (Marchi et al. 2013), into the WDSBA to benefit the comparative studies in the future.
2. To optimise for objective functions other than the ones used in the thesis leading to more realistic engineering problems being solved.
3. To increase the dimensionality of benchmark problems in terms of number of objectives, i.e., to move from the multi-objective (two to three) to many-objective (more than three) benchmark optimisation problems.

Regarding solving more complex design problems which are highly constrained, recommendations for future research work are summarised as follows:

1. To develop an effective approach which can generate a high quality initial population in terms of constraint violations for complex design problems.
2. To reduce the size of search space for those problems which have a large number of decision variables via special techniques, such as the global sensitivity analysis (Fu et al. 2012b) or graph theory (Zheng et al. 2013a).

APPENDIX A MAXIMUM PRESSURE HEAD REQUIREMENT AT EACH NODE OF MOD

Table A.1 Maximum pressure head requirement at each node of MOD

N_l	H_{max} (m)	N_l	H_{max} (m)	N_l	H_{max} (m)	N_l	H_{max} (m)	N_l	H_{max} (m)	N_l	H_{max} (m)
1	35.007	47	40.820	93	38.571	139	40.375	185	42.155	231	38.816
2	34.875	48	40.794	94	36.861	140	35.150	186	41.674	232	39.110
3	35.797	49	42.823	95	37.332	141	35.396	187	40.806	233	39.612
4	37.254	50	41.155	96	37.395	142	34.659	188	41.325	234	39.642
5	38.235	51	41.668	97	37.529	143	34.659	189	41.271	235	39.505
6	38.545	52	41.722	98	37.503	144	35.051	190	41.157	236	41.959
7	38.545	53	35.224	99	37.761	145	34.795	191	40.728	237	40.087
8	38.413	54	37.377	100	39.724	146	36.549	192	40.732	238	38.343
9	36.321	55	38.016	101	40.243	147	36.890	193	42.296	239	39.195
10	37.497	56	38.084	102	40.840	148	36.549	194	40.095	240	39.329
11	38.000	57	38.365	103	40.716	149	38.814	195	41.111	241	41.582
12	37.112	58	38.451	104	40.754	150	39.183	196	40.155	242	41.434
13	36.426	59	37.735	105	41.123	151	38.690	197	39.473	243	42.590
14	37.481	60	39.016	106	39.650	152	38.688	198	40.061	244	42.498
15	33.243	61	39.451	107	40.227	153	38.481	199	39.966	245	42.452
16	35.150	62	39.395	108	40.203	154	36.246	200	39.565	246	42.446
17	34.971	63	36.549	109	40.546	155	36.996	201	39.796	247	43.795
18	37.906	64	36.058	110	40.580	156	36.964	202	37.800	248	43.168
19	37.739	65	36.693	111	42.183	157	37.421	203	38.297	249	38.204
20	36.785	66	36.282	112	39.742	158	37.745	204	39.469	250	38.669
21	37.188	67	35.773	113	40.287	159	38.615	205	37.735	251	37.555
22	36.877	68	35.547	114	39.576	160	38.732	206	38.303	252	36.487
23	37.513	69	34.799	115	38.544	161	39.796	207	36.621	253	37.850
24	39.295	70	33.911	116	43.811	162	39.131	208	36.465	254	37.595
25	39.387	71	33.688	117	43.905	163	39.507	209	37.637	255	37.727
26	39.846	72	33.436	118	43.769	164	38.573	210	37.262	256	43.003
27	40.175	73	33.047	119	43.797	165	38.235	211	37.842	257	35.849
28	38.355	74	32.670	120	43.480	166	41.833	212	38.010	258	34.957
29	38.204	75	33.065	121	43.468	167	41.746	213	37.200	259	34.919
30	38.403	76	33.408	122	42.755	168	41.616	214	34.201	260	34.919
31	38.361	77	33.757	123	42.500	169	40.415	215	34.651	261	33.949
32	38.700	78	35.895	124	42.452	170	38.407	216	33.502	262	33.714
33	41.239	79	37.585	125	42.402	171	38.451	217	33.340	263	33.546
34	41.163	80	37.751	126	40.740	172	38.459	218	39.451	264	36.745
35	40.987	81	37.687	127	42.229	173	38.483	219	40.580	265	38.537
36	41.800	82	37.455	128	42.640	174	42.743	220	42.356	266	37.691

37	41.853	83	38.617	129	42.083	175	42.590	221	40.333	267	38.289
38	41.935	84	38.046	130	41.498	176	42.701	222	39.403	268	38.888
39	40.935	85	38.339	131	40.874	177	43.017	223	42.951	-	-
40	42.905	86	39.509	132	38.134	178	43.384	224	42.755	-	-
41	43.119	87	38.888	133	38.806	179	43.404	225	42.434	-	-
42	41.833	88	39.608	134	38.976	180	43.306	226	42.556	-	-
43	41.001	89	38.914	135	38.940	181	44.108	227	42.843	-	-
44	40.929	90	38.800	136	38.583	182	43.953	228	43.460	-	-
45	40.726	91	39.305	137	39.133	183	43.366	229	43.450	-	-
46	40.363	92	38.860	138	39.443	184	42.690	230	36.008	-	-

Note: N_I - node index which is a consecutive number starting from one to the total number of nodes in the network; H_{max} - maximum pressure head requirement.

APPENDIX B MATLAB EXECUTABLE FUNCTION (MEX-FUNCTION)

Matlab EXecutable Functions enable the invocation of subroutines created in *C*, *C++*, or *Fortran* from the MATLAB command line as built-in functions. These programs, known as binary MEX-files, are dynamically linked with the MATLAB interpreter. Using the MEX function can take advantages of both languages. In particular, the performance-critical routines which are time-consuming are implemented via the MEX interface, and the other routines are efficiently constructed within the MATLAB environment, which is highly productive by eliminating the need for most low-level programming.

By analysing the time spent on solving a typical multi-objective optimal design problem of Water Distribution Systems, it is found that the hydraulic simulation accounts for the majority of the time elapsed. Therefore, the computational overhead when dealing with such kind of problems especially for large problems can be effectively reduced by calling the MEX functions, which encapsulate objective functions involving hydraulic simulations written in *C* language. Consequently, the computational costs in terms of running time using the GALAXY and AMALGAM methods, which are implemented in MATLAB, are comparable to those required by other *C* language based algorithms. Figure B.1 illustrates the general framework of using the MEX function for the multi-objective design of Water Distribution Systems.

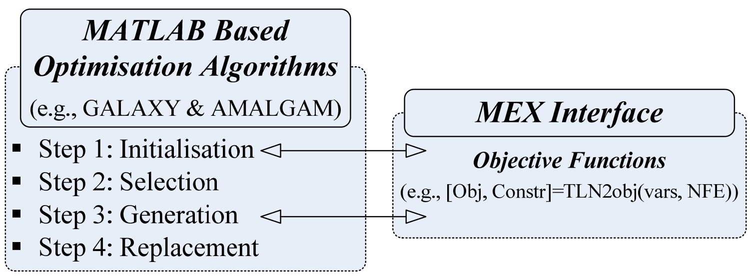


Figure B.1 Framework of solving multi-objective design of WDSs via MEX function

An example of such MEX function for solving the TLN problem is presented in Figure B.2, which includes the MEX function interface as well as part of the problem definition.

```c
#include <math.h>
#include "mex.h"
#include "epanet2mex.h"

void TLN2obj(double *vars, double *objs, double *constrs, int NFE) {
    ****************************************************************
    * HERE IS THE CODE FOR COMPUTING OBJECTIVE VALUES.             *
    * HYDRAULIC SIMULATION IS ALSO IMPLEMENTED IN THIS FUNCTION.   *
    ****************************************************************
    objs[0] = totalCost/1000000.0; // $ => $ MM
    objs[1] = -networkResilience;
    constrs[0] = -(pressureViolation+errorCode);
    return;
}

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
{
    double *vars,*objs,*constrs;
    int NFE;
    /* Check for proper number of arguments. */
    if(nrhs!=2)
    {
        mexErrMsgIdAndTxt( "MATLAB:TLN2obj:invalidNumInputs", "Two inputs required..."
        );
    }
    else if(nlhs>2)
    {
        mexErrMsgIdAndTxt( "MATLAB:TLN2obj:maxlhs", "Too many output arguments...");
    }
    /* Create matrix for return arguments. */
    plhs[0] = mxCreateDoubleMatrix((mwSize)2, (mwSize)1, mxREAL);
    plhs[1] = mxCreateDoubleMatrix((mwSize)1, (mwSize)1, mxREAL);
    /* Assign a pointer to each input and output. */
      vars = mxGetPr(prhs[0]);
      NFE = mxGetScalar(prhs[1]);
      objs = mxGetPr(plhs[0]);
      constrs = mxGetPr(plhs[1]);
    /* Call the TLN2obj subroutine. */
    TLN2obj(vars, objs, constrs, NFE);
}
```

Figure B.2 Sample code of MEX function used for this thesis

APPENDIX C VERIFICATION OF COMPUTATIONAL BUDGET USING THE NSGA-II FOR BENCHMARK PROBLEMS

C.1 Sensitivity analysis on the indices of SBX and PM

Table C.1 Sensitivity analysis for the BAK problem

Pop=100; *Gen*=2000; P_{SBX}=0.9; P_{PM}=1/*ND*				
Indices=1	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9999**	0.9991	0.9982	**0.5871**
Max:	1.0000	0.9998	0.9988	0.6259
Min:	0.9998	0.9929	0.9973	0.5324
Std:	0.0000	0.0020	0.0004	0.0239
Indices=5	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9999**	0.9993	**0.9983**	0.5811
Max:	1.0000	0.9998	0.9988	0.6187
Min:	0.9999	0.9930	0.9973	0.5324
Std:	0.0000	0.0017	0.0004	0.0208
Indices=10	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9999**	0.9990	0.9982	0.5736
Max:	1.0000	0.9998	0.9989	0.6115
Min:	0.9999	0.9929	0.9973	0.5324
Std:	0.0000	0.0020	0.0004	0.0215
Indices=15	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9999**	**0.9995**	**0.9983**	0.5664
Max:	1.0000	0.9998	0.9988	0.6331
Min:	0.9999	0.9930	0.9973	0.5252
Std:	0.0000	0.0012	0.0003	0.0281
Indices=20	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9999**	0.9988	0.9981	0.5640
Max:	1.0000	0.9998	0.9988	0.6187
Min:	0.9999	0.9929	0.9970	0.5036
Std:	0.0000	0.0023	0.0005	0.0330

Note: The best value on average according to each performance indicator is shown in bold. *Pop* denotes population size. *Gen* denotes number of generations. P_{SBX} denotes probability of Simulated Binary Crossover. P_{PM} denotes probability of Polynomial Mutation. *ND* denotes number of decision variables.

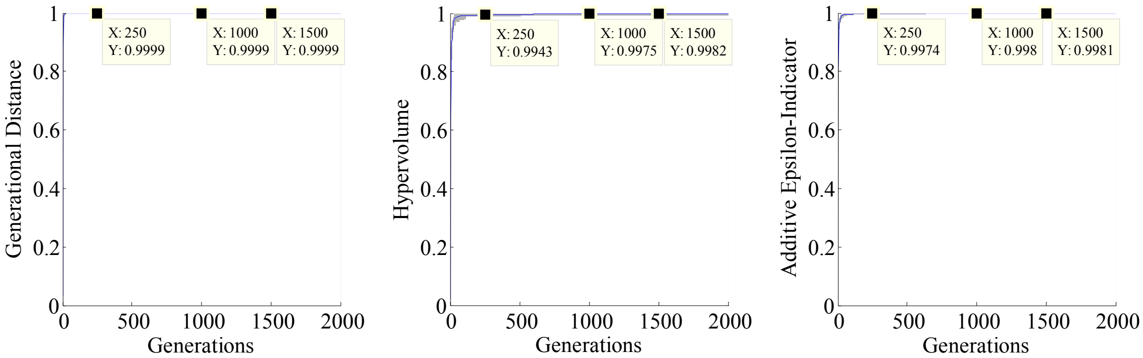
Table C.2 Sensitivity analysis for the GOY problem

Pop=100; *Gen*=2000; P_{SBX}=0.9; P_{PM}=1/*ND*				
Indices=1	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9974**	**0.9659**	**0.9612**	0.1614
Max:	0.9992	0.9845	0.9802	0.2658
Min:	0.9940	0.9468	0.9443	0.0900
Std:	0.0011	0.0098	0.0096	0.0449
Indices=5	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	0.9971	0.9630	0.9587	0.1530
Max:	0.9988	0.9843	0.9803	0.2434
Min:	0.9927	0.9391	0.9379	0.0879
Std:	0.0015	0.0102	0.0097	0.0401
Indices=10	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	0.9973	0.9582	0.9543	0.1570
Max:	0.9990	0.9804	0.9760	0.2311
Min:	0.9951	0.9393	0.9382	0.0798
Std:	0.0010	0.0108	0.0097	0.0435
Indices=15	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	0.9969	0.9605	0.9566	0.1630
Max:	0.9993	0.9830	0.9779	0.2352
Min:	0.9915	0.9419	0.9402	0.0838
Std:	0.0016	0.0105	0.0100	0.0377
Indices=20	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9974**	0.9590	0.9548	**0.1673**
Max:	0.9991	0.9811	0.9755	0.2556
Min:	0.9930	0.9429	0.9410	0.0920
Std:	0.0015	0.0088	0.0080	0.0435

Table C.3 Sensitivity analysis for the PES problem

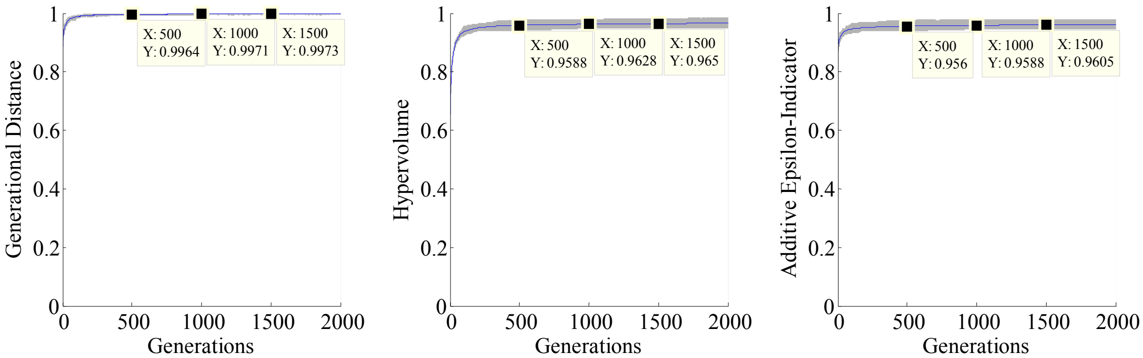
Pop=100; *Gen*=2000; P_{SBX}=0.9; P_{PM}=1/*ND*				
Indices=1	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	0.9900	**0.9412**	**0.9253**	0.0000
Max:	0.9939	0.9647	0.9524	0.0000
Min:	0.9841	0.9240	0.9054	0.0000
Std:	0.0023	0.0095	0.0113	0.0000
Indices=5	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	0.9933	0.9391	0.9206	0.0000
Max:	0.9960	0.9583	0.9436	0.0000
Min:	0.9897	0.9181	0.8978	0.0000
Std:	0.0017	0.0102	0.0116	0.0000
Indices=10	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	**0.9942**	0.9367	0.9177	**0.0001**
Max:	0.9971	0.9550	0.9386	0.0026
Min:	0.9914	0.9283	0.9092	0.0000
Std:	0.0017	0.0063	0.0071	0.0005
Indices=15	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	0.9935	0.9320	0.9134	0.0000
Max:	0.9962	0.9477	0.9326	0.0000
Min:	0.9894	0.9128	0.8924	0.0000
Std:	0.0019	0.0087	0.0099	0.0000
Indices=20	I_{GD}	I_{HV}	$I_{\varepsilon+}$	I_{EP}
Avg:	0.9927	0.9296	0.9126	0.0000
Max:	0.9963	0.9441	0.9283	0.0000
Min:	0.9884	0.9145	0.8946	0.0000
Std:	0.0021	0.0082	0.0095	0.0000

C.2 Dynamic performance of the NSGA-II for the BAK and GOY problems



(a) I_{GD} Indicator (b) I_{HV} Indicator (c) $I_{\varepsilon+}$ Indicator

Figure C.1 Dynamic performance of the NSGA-II for the BAK problem



(a) I_{GD} Indicator (b) I_{HV} Indicator (c) $I_{\varepsilon+}$ Indicator

Figure C.2 Dynamic performance of the NSGA-II for the GOY problem

APPENDIX D COMPARISON BETWEEN GALAXY AND THE OTHER MOEAS FOR THE OTHER BENCHMARK PROBLEMS

D.1 Ultimate Performance

Table D.1 Ultimate performance of various MOEAs for the TRN problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	**1.0000**	0.9998	0.9999	0.9998	0.9999
	Max.	**1.0000**	0.9999	**1.0000**	0.9999	**1.0000**
	Min.	**0.9999**	0.9996	0.9996	0.9996	0.9995
	Std.	**0.0000**	0.0001	0.0001	0.0001	0.0001
I_{HV}	Avg.	**0.9996**	0.9991	0.9987	0.9989	0.9984
	Max.	0.9997	0.9993	**0.9999**	0.9993	**0.9999**
	Min.	**0.9995**	0.9987	0.9958	0.9969	0.9943
	Std.	**0.0001**	0.0002	0.0010	0.0005	0.0016
$I_{\varepsilon+}$	Avg.	**0.9963**	0.9946	0.9962	0.9943	0.9958
	Max.	0.9967	0.9963	0.9982	0.9956	**0.9988**
	Min.	**0.9962**	0.9908	0.9908	0.9925	0.9889
	Std.	**0.0002**	0.0011	0.0016	0.0009	0.0024
I_{EP}	Avg.	**0.7828**	0.6391	0.7276	0.5721	0.7211
	Max.	**0.7969**	0.6797	0.7578	0.6328	0.7578
	Min.	**0.7656**	0.5703	0.6953	0.5234	0.6797
	Std.	**0.0090**	0.0235	0.0180	0.0232	0.0227

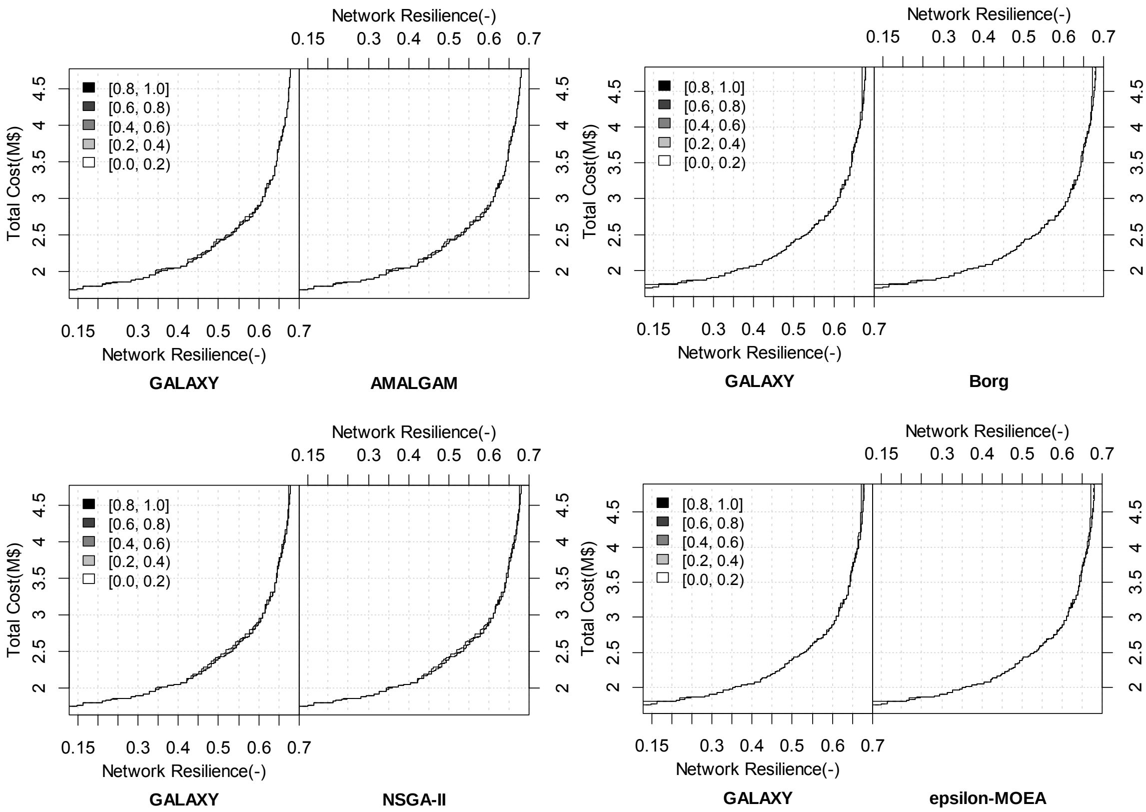


Figure D.1 Comparison of GALAXY with the other MOEAs for the TRN problem using the EAF tool

Table D.2 Ultimate performance of various MOEAs for the TLN problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	0.9997	**0.9998**	**0.9998**	0.9989	0.9977
	Max.	**1.0000**	0.9999	**1.0000**	0.9998	0.9991
	Min.	0.9984	**0.9994**	0.9990	0.9961	0.9882
	Std.	0.0004	**0.0001**	0.0002	0.0009	0.0020
I_{HV}	Avg.	**0.9987**	0.9974	0.9971	0.9921	0.9791
	Max.	**0.9999**	0.9979	0.9981	0.9987	0.9898
	Min.	**0.9980**	0.9960	0.9944	0.9773	0.9477
	Std.	0.0007	**0.0006**	0.0008	0.0052	0.0091
$I_{\varepsilon+}$	Avg.	**0.9953**	0.9939	0.9930	0.9784	0.9613
	Max.	**0.9977**	0.9948	0.9948	0.9925	0.9834
	Min.	0.9901	**0.9907**	0.9891	0.9573	0.9314
	Std.	0.0016	**0.0013**	0.0019	0.0094	0.0118
I_{EP}	Avg.	**0.8254**	0.6196	0.5848	0.5289	0.4137
	Max.	**0.8684**	0.6491	0.6140	0.6228	0.4825
	Min.	**0.7719**	0.5526	0.5351	0.4386	0.3070
	Std.	0.0251	0.0248	**0.0180**	0.0455	0.0411

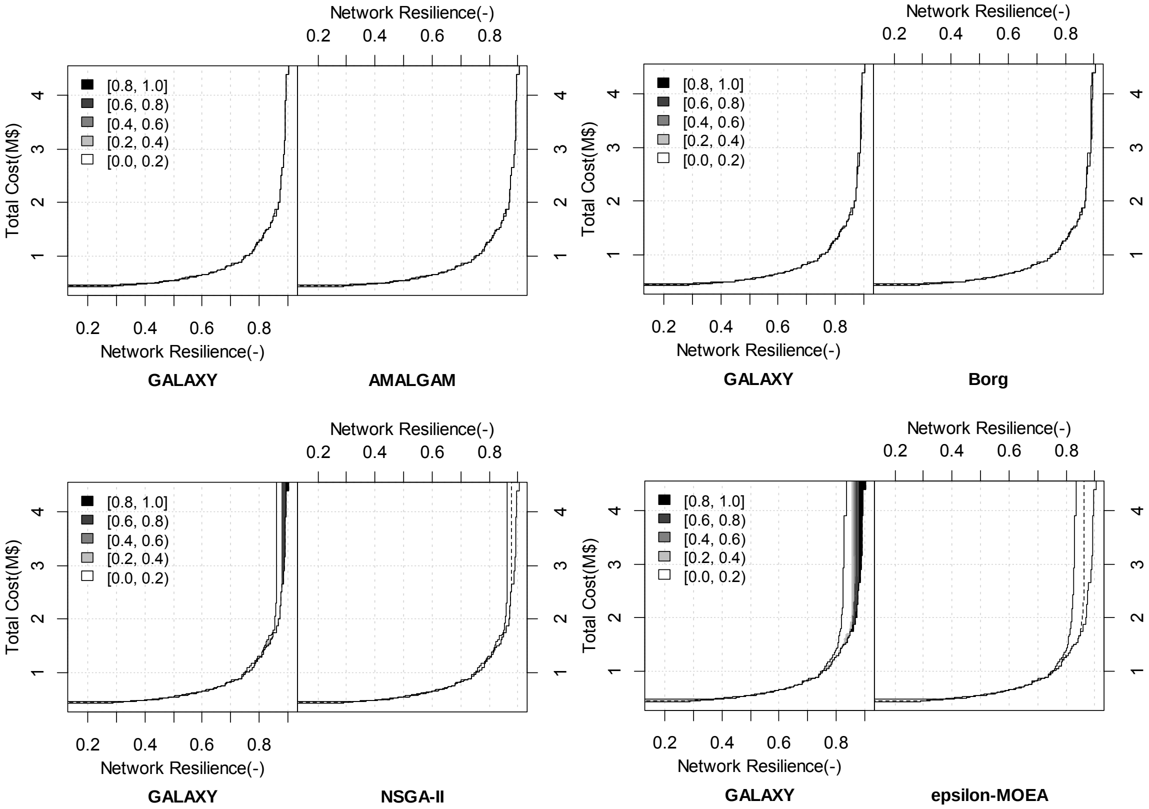


Figure D.2 Comparison of GALAXY with the other MOEAs for the TLN problem using the EAF tool

Table D.3 Ultimate performance of various MOEAs for the NYT problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	0.9931	0.9970	0.9905	0.9983	**0.9995**
	Max.	0.9986	0.9991	0.9919	0.9993	**0.9999**
	Min.	0.9808	0.9871	0.9891	0.9929	**0.9983**
	Std.	0.0044	0.0023	0.0008	0.0012	**0.0004**
I_{HV}	Avg.	0.9740	0.9744	0.9371	**0.9766**	0.9492
	Max.	0.9889	0.9907	0.9676	**0.9914**	0.9699
	Min.	0.9423	0.9341	0.9190	**0.9528**	0.9283
	Std.	0.0114	0.0137	0.0125	0.0114	**0.0111**
$I_{\varepsilon+}$	Avg.	0.9745	0.9726	0.9528	**0.9772**	0.9571
	Max.	0.9902	0.9901	0.9726	**0.9922**	0.9708
	Min.	0.9476	0.9269	0.9421	**0.9560**	0.9365
	Std.	0.0126	0.0167	0.0081	0.0118	**0.0069**
I_{EP}	Avg.	0.0655	0.0855	0.0190	0.1308	**0.2375**
	Max.	0.1356	0.1691	0.0415	0.1802	**0.2695**
	Min.	0.0000	0.0239	0.0064	0.0303	**0.1738**
	Std.	0.0430	0.0367	**0.0084**	0.0312	0.0231

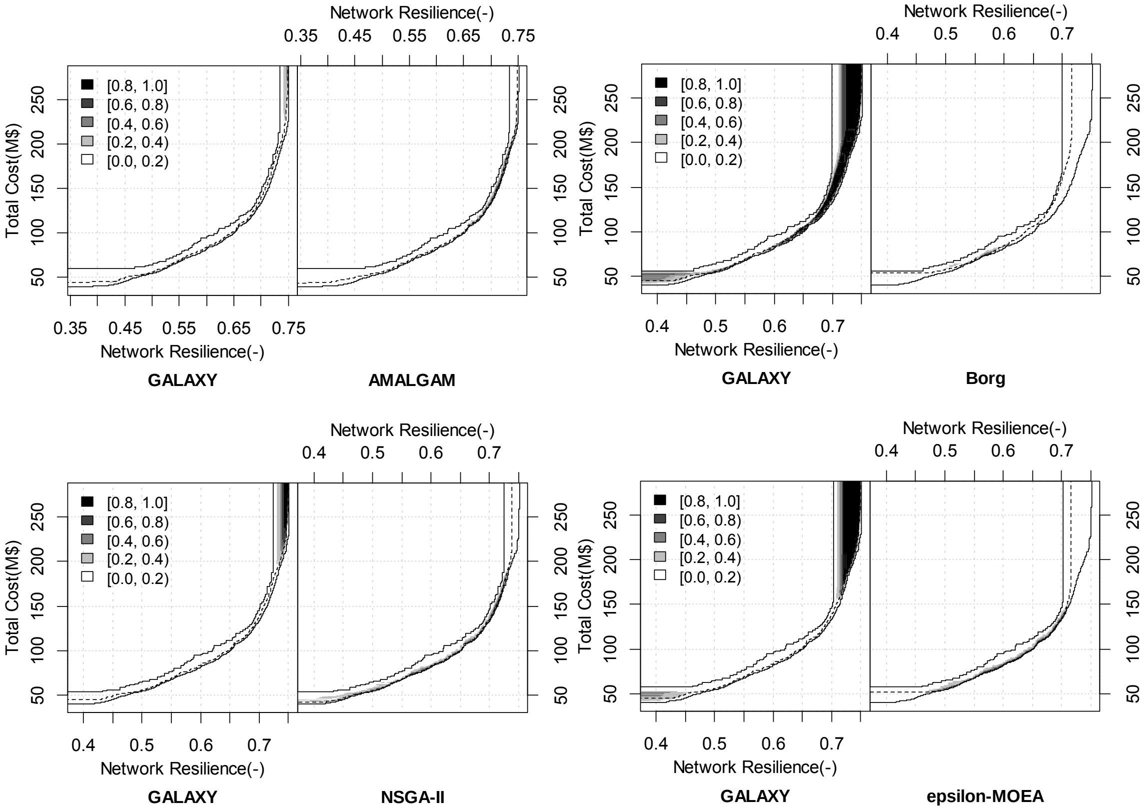


Figure D.3 Comparison of GALAXY with the other MOEAs for the NYT problem using the EAF tool

Table D.4 Ultimate performance of various MOEAs for the BLA problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	**0.9982**	0.9962	0.9941	0.9964	0.9969
	Max.	**0.9993**	0.9988	0.9982	0.9979	0.9986
	Min.	**0.9964**	0.9894	0.9883	0.9939	0.9938
	Std.	**0.0007**	0.0018	0.0025	0.0011	0.0012
I_{HV}	Avg.	**0.9921**	0.9828	0.9731	0.9869	0.9744
	Max.	**0.9958**	0.9924	0.9810	0.9912	0.9831
	Min.	**0.9879**	0.9718	0.9633	0.9821	0.9577
	Std.	**0.0019**	0.0050	0.0048	0.0023	0.0063
$I_{\varepsilon+}$	Avg.	**0.9927**	0.9846	0.9787	0.9883	0.9770
	Max.	**0.9956**	0.9927	0.9880	0.9927	0.9858
	Min.	**0.9882**	0.9727	0.9681	0.9828	0.9575
	Std.	**0.0019**	0.0046	0.0055	0.0026	0.0068
I_{EP}	Avg.	**0.0396**	0.0144	0.0049	0.0036	0.0152
	Max.	0.0788	0.0721	0.0522	0.0466	**0.1110**
	Min.	0.0000	0.0000	0.0000	0.0000	0.0000
	Std.	0.0158	0.0196	0.0117	**0.0088**	0.0237

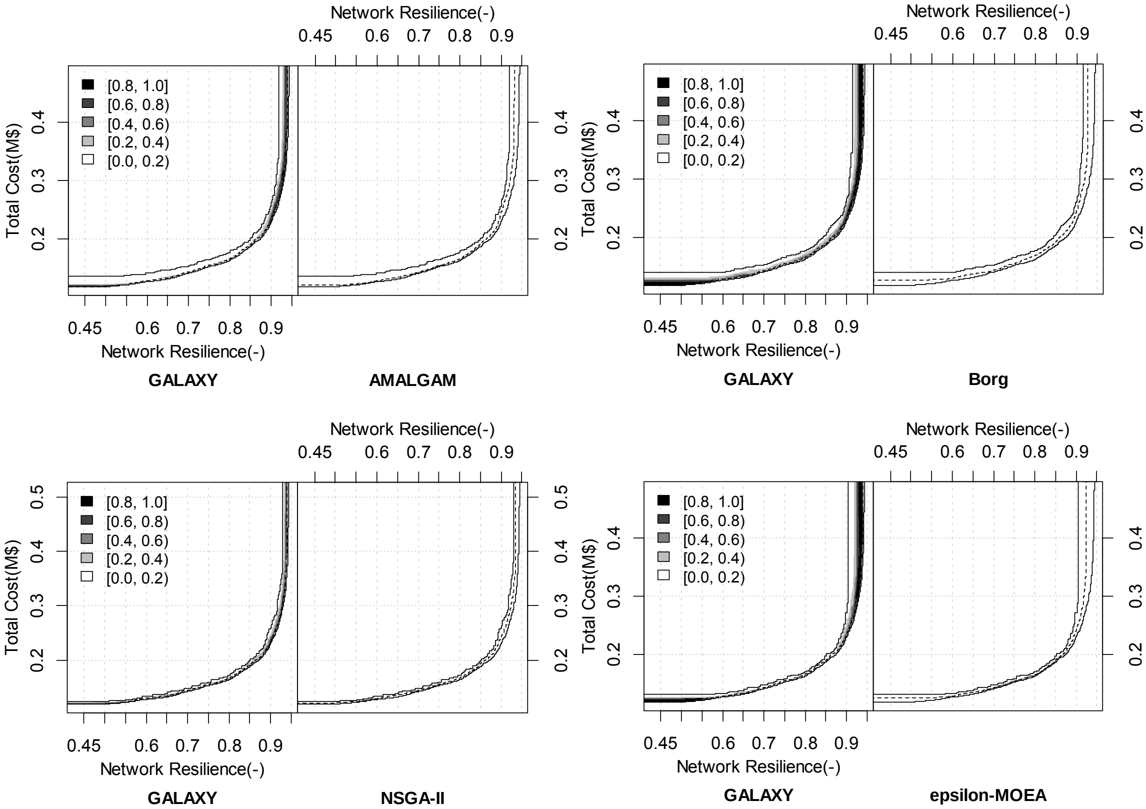


Figure D.4 Comparison of GALAXY with the other MOEAs for the BLA problem using the EAF tool

Table D.5 Ultimate performance of various MOEAs for the GOY problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	**0.9983**	**0.9983**	0.9969	0.9964	0.9955
	Max.	0.9989	0.9988	0.9987	**0.9990**	0.9988
	Min.	0.9974	**0.9977**	0.9937	0.9926	0.9905
	Std.	0.0004	**0.0002**	0.0011	0.0018	0.0019
I_{HV}	Avg.	0.9932	**0.9941**	0.9615	0.9588	0.9366
	Max.	0.9951	**0.9955**	0.9910	0.9784	0.9581
	Min.	0.9898	**0.9924**	0.9430	0.9397	0.9031
	Std.	0.0013	**0.0007**	0.0135	0.0092	0.0115
$I_{\varepsilon+}$	Avg.	0.9904	**0.9916**	0.9623	0.9560	0.9437
	Max.	0.9933	0.9941	**0.9944**	0.9770	0.9657
	Min.	0.9856	**0.9899**	0.9449	0.9385	0.9158
	Std.	0.0022	**0.0010**	0.0140	0.0094	0.0101
I_{EP}	Avg.	0.1243	0.1123	0.0255	**0.1352**	0.0342
	Max.	0.1697	0.1431	0.0654	**0.2127**	0.0859
	Min.	**0.1002**	0.0736	0.0000	0.0818	0.0061
	Std.	**0.0167**	0.0176	0.0175	0.0296	0.0197

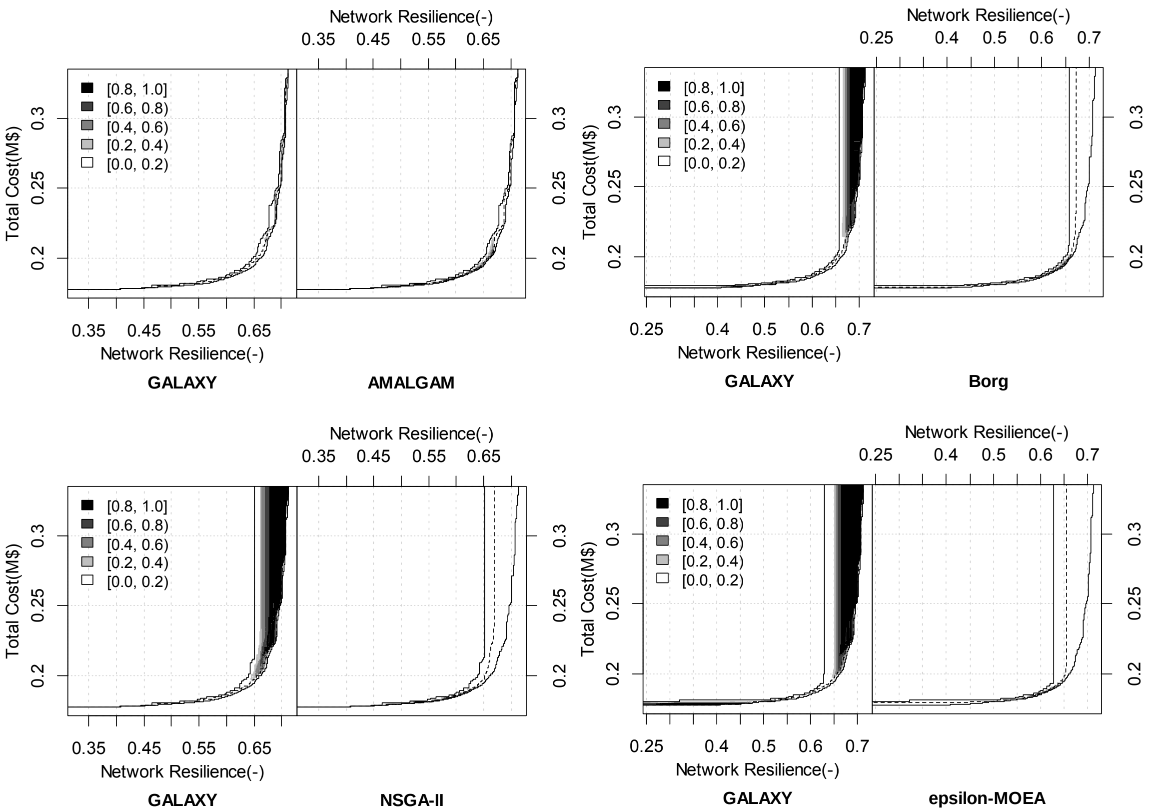


Figure D.5 Comparison of GALAXY with the other MOEAs for the GOY problem using the EAF tool

Table D.6 Ultimate performance of various MOEAs for the FOS problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	0.9945	0.9619	0.9908	0.9923	**0.9955**
	Max.	0.9958	0.9754	0.9962	0.9961	**0.9983**
	Min.	**0.9900**	0.9500	0.9866	0.9823	0.9870
	Std.	**0.0013**	0.0062	0.0021	0.0029	0.0024
I_{HV}	Avg.	**0.9849**	0.9552	0.8883	0.9100	0.8767
	Max.	**0.9870**	0.9699	0.9128	0.9337	0.9150
	Min.	**0.9813**	0.9409	0.8615	0.8806	0.8528
	Std.	**0.0014**	0.0074	0.0128	0.0110	0.0132
$I_{\varepsilon+}$	Avg.	**0.9526**	0.9499	0.8871	0.9053	0.8724
	Max.	**0.9694**	0.9651	0.9124	0.9317	0.9143
	Min.	**0.9407**	0.9336	0.8582	0.8762	0.8466
	Std.	**0.0065**	0.0081	0.0140	0.0117	0.0141
I_{EP}	Avg.	**0.1592**	0.0390	0.0000	0.0000	0.0000
	Max.	**0.1747**	0.0390	0.0000	0.0000	0.0000
	Min.	**0.1171**	0.0390	0.0000	0.0000	0.0000
	Std.	0.0116	**0.0000**	0.0000	0.0000	0.0000

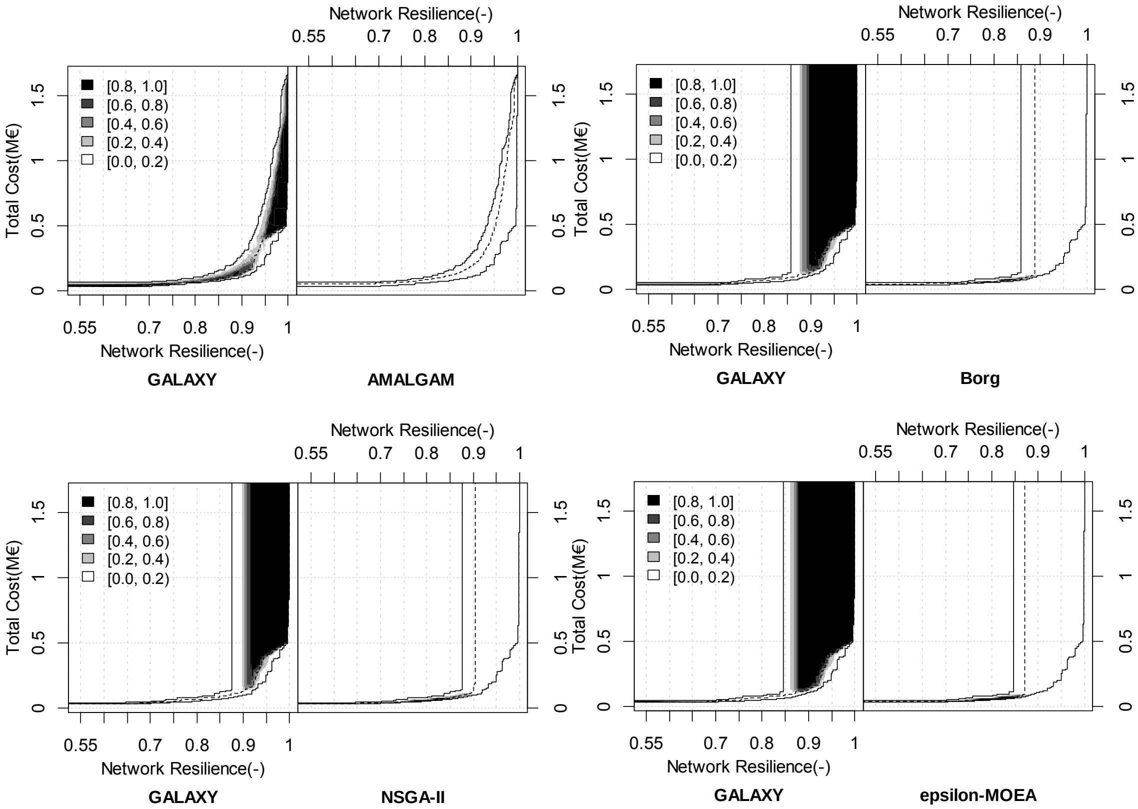


Figure D.6 Comparison of GALAXY with the other MOEAs for the FOS problem using the EAF tool

Table D.7 Ultimate performance of various MOEAs for the MOD problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	**0.9883**	0.9813	0.9693	0.9857	0.9878
	Max.	0.9902	0.9859	0.9739	0.9894	**0.9914**
	Min.	**0.9863**	0.9605	0.9635	0.9809	0.9841
	Std.	**0.0011**	0.0046	0.0028	0.0024	0.0019
I_{HV}	Avg.	**0.9498**	0.9220	0.8904	0.9192	0.9076
	Max.	**0.9648**	0.9334	0.9073	0.9338	0.9171
	Min.	**0.9405**	0.9108	0.8756	0.9013	0.9001
	Std.	0.0054	0.0059	0.0075	0.0073	**0.0052**
$I_{\varepsilon+}$	Avg.	**0.9392**	0.9142	0.8957	0.9083	0.9002
	Max.	**0.9562**	0.9474	0.9138	0.9254	0.9108
	Min.	**0.9280**	0.9006	0.8808	0.8897	0.8929
	Std.	0.0062	0.0086	0.0076	0.0081	**0.0055**
I_{EP}	Avg.	0.0000	0.0000	0.0000	0.0000	0.0000
	Max.	0.0000	0.0000	0.0000	0.0000	0.0000
	Min.	0.0000	0.0000	0.0000	0.0000	0.0000
	Std.	0.0000	0.0000	0.0000	0.0000	0.0000

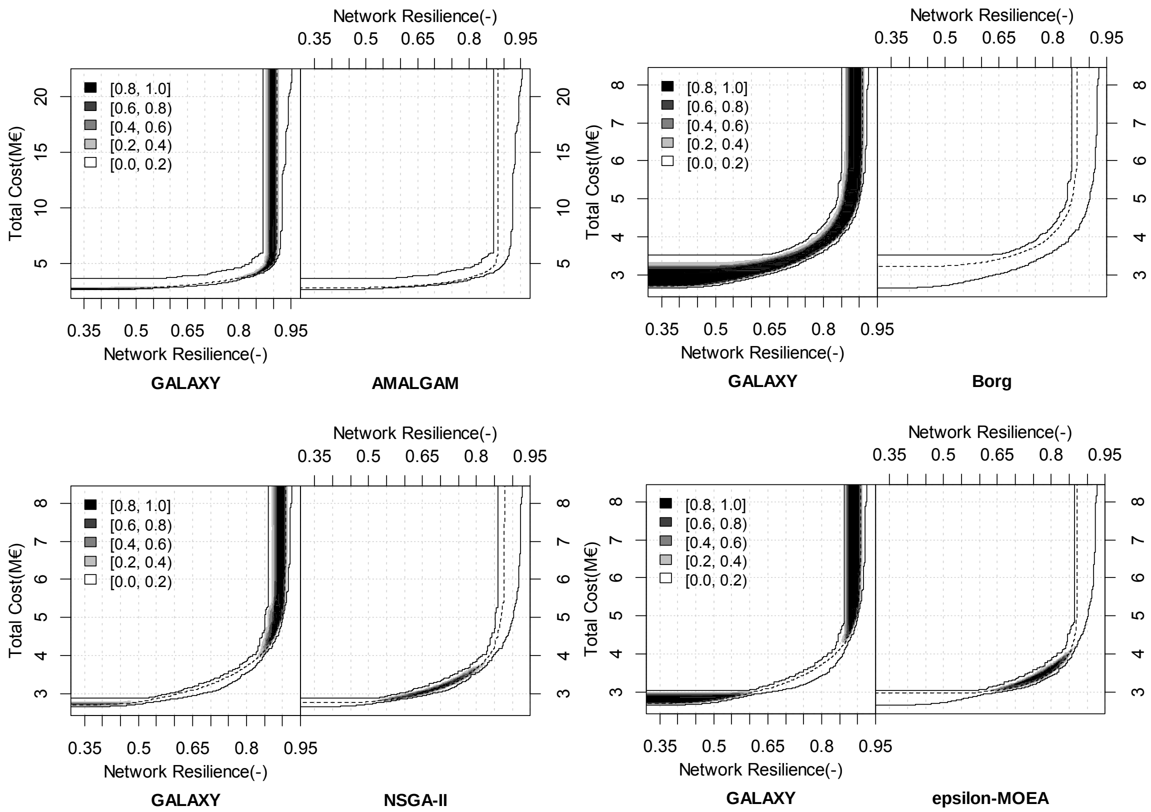


Figure D.7 Comparison of GALAXY with the other MOEAs for the MOD problem using the EAF tool

Table D.8 Ultimate performance of various MOEAs for the BIN problem

Indicators		GALAXY	AMALGAM	Borg	NSGA-II	ε-MOEA
I_{GD}	Avg.	0.9775	0.9651	0.9833	0.9821	**0.9860**
	Max.	0.9805	0.9711	0.9890	0.9853	**0.9905**
	Min.	0.9752	0.9581	0.9793	0.9790	**0.9795**
	Std.	**0.0012**	0.0036	0.0022	0.0018	0.0029
I_{HV}	Avg.	**0.9470**	0.9117	0.8518	0.9058	0.8797
	Max.	**0.9530**	0.9262	0.8676	0.9159	0.8934
	Min.	**0.9409**	0.8916	0.8246	0.8925	0.8665
	Std.	**0.0026**	0.0080	0.0090	0.0063	0.0077
$I_{\varepsilon+}$	Avg.	**0.9572**	0.9374	0.8517	0.8986	0.8746
	Max.	**0.9623**	0.9501	0.8707	0.9103	0.8889
	Min.	**0.9526**	0.9206	0.8240	0.8871	0.8607
	Std.	**0.0024**	0.0078	0.0089	0.0065	0.0078
I_{EP}	Avg.	**0.0001**	0.0000	0.0000	0.0000	0.0000
	Max.	**0.0016**	0.0000	0.0000	0.0000	0.0000
	Min.	0.0000	0.0000	0.0000	0.0000	0.0000
	Std.	**0.0003**	0.0000	0.0000	0.0000	0.0000

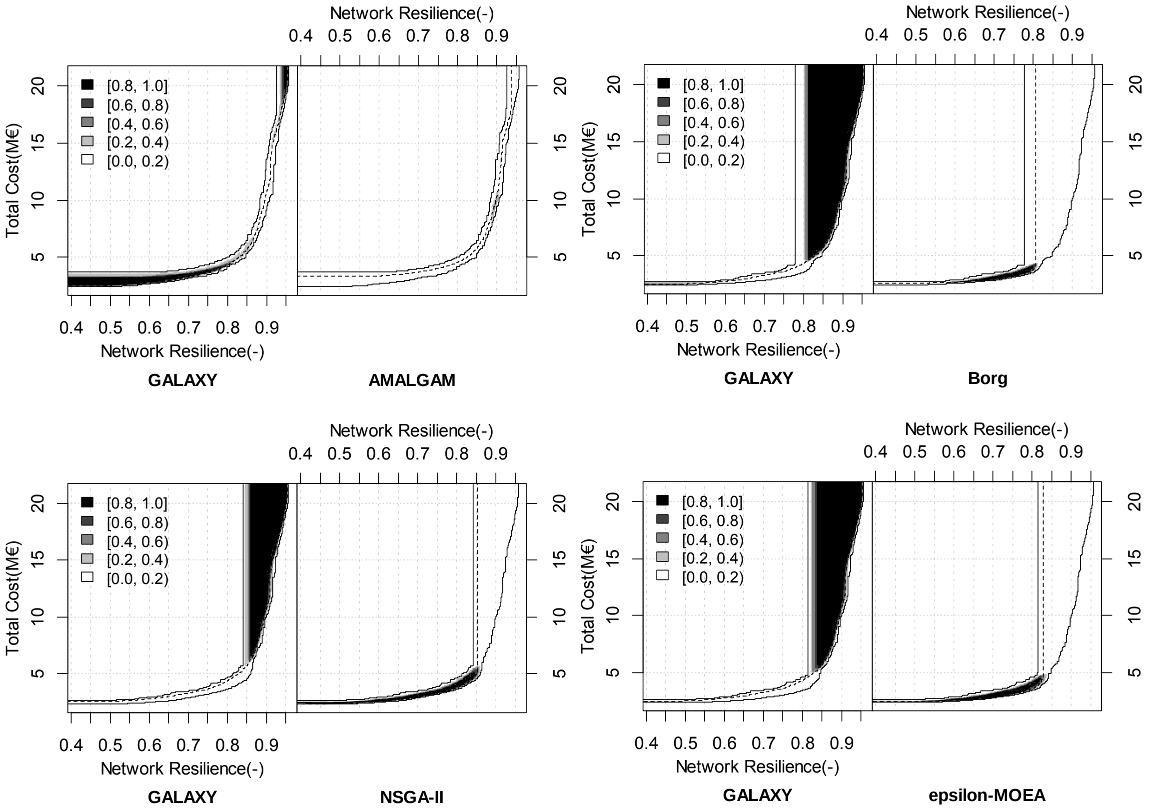
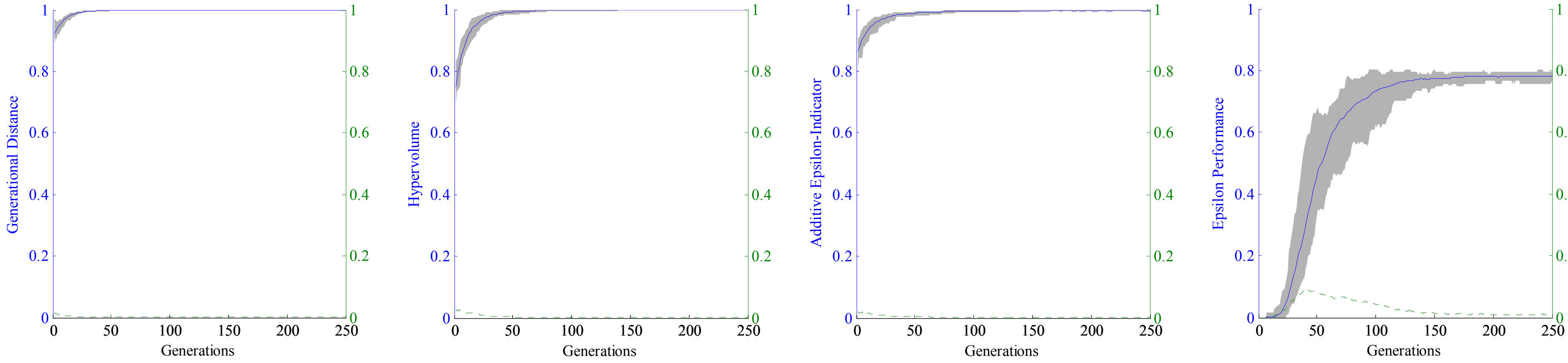


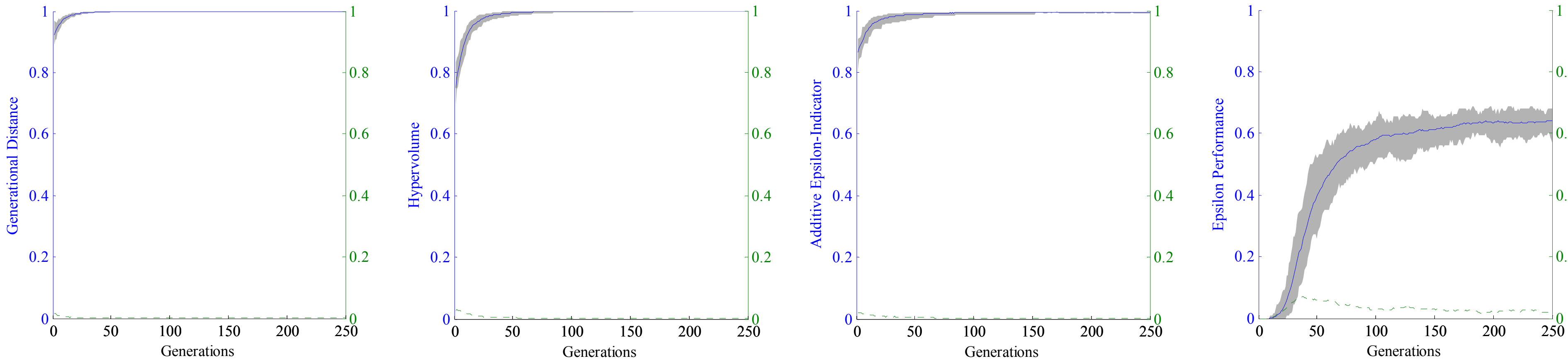
Figure D.8 Comparison of GALAXY with the other MOEAs for the BIN problem using the EAF tool

D.2 Dynamic Performance

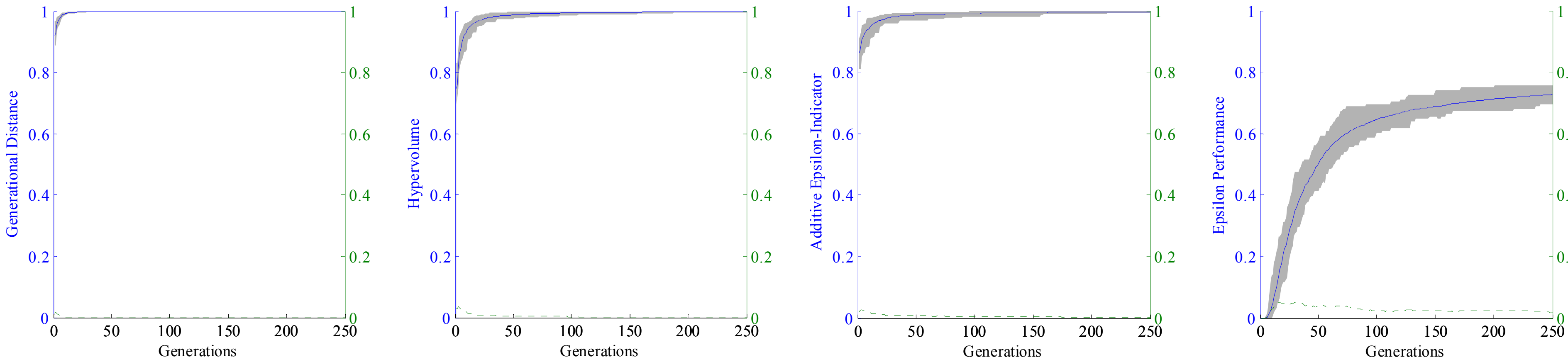
GALAXY



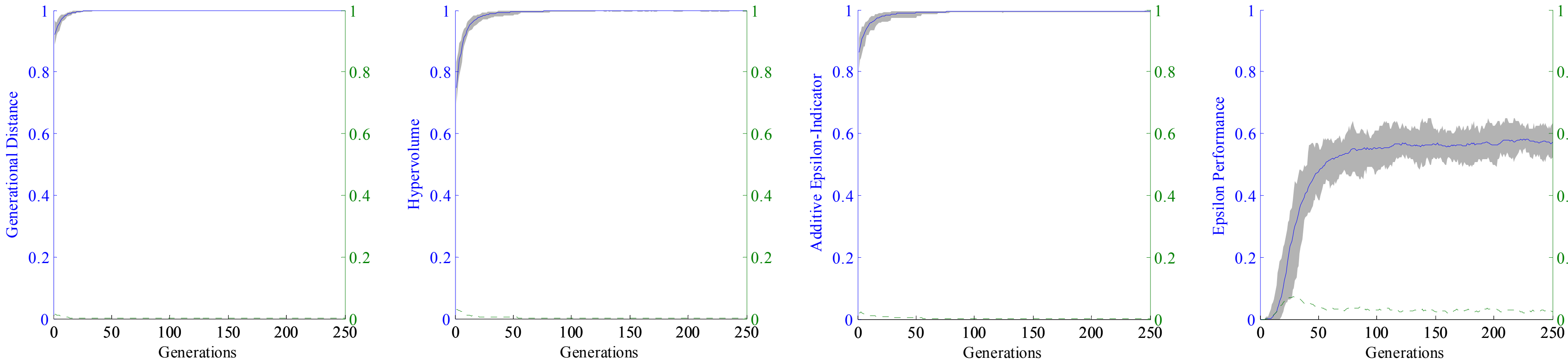
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0
50
100
150
200
250
0
0.2
0.4
0.6
0.8
1

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0
50
100
150
200
250
0
0.2
0.4
0.6
0.8
1

ε-MOEA

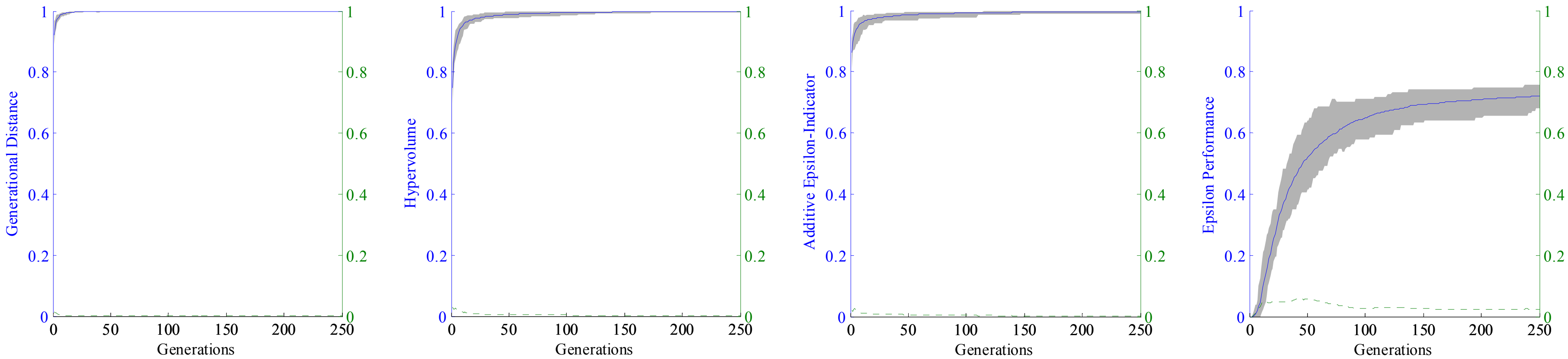
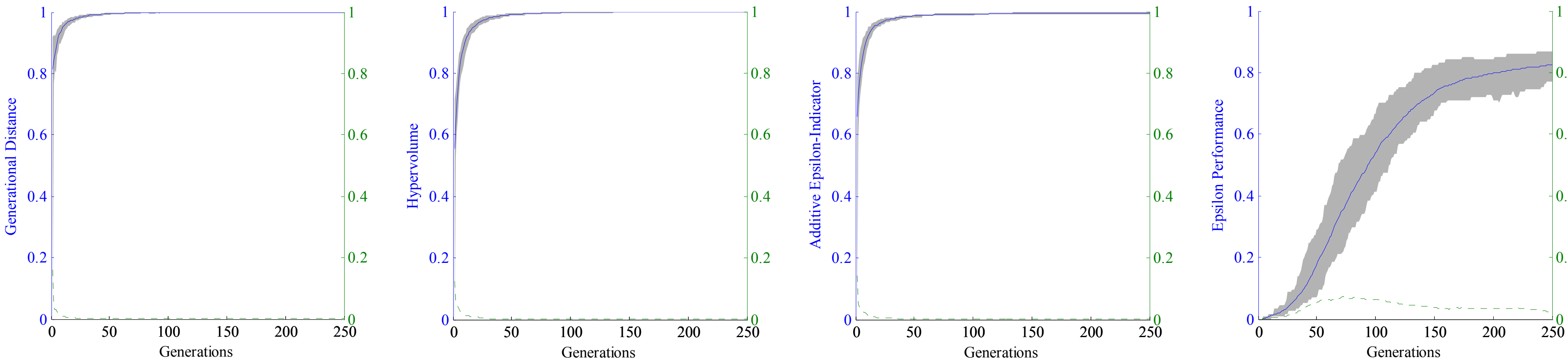
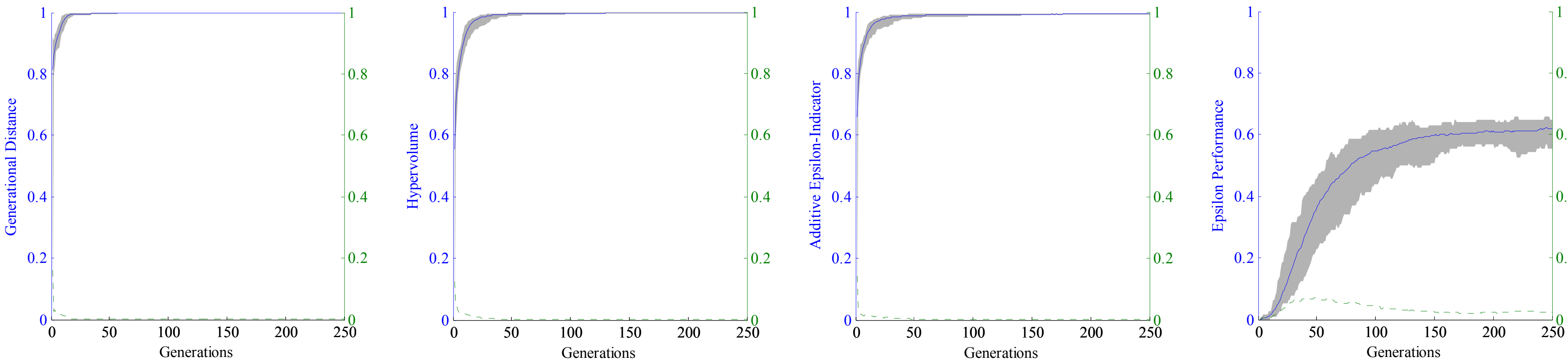


Figure D.9 Dynamic performances of various MOEAs for the TRN problem

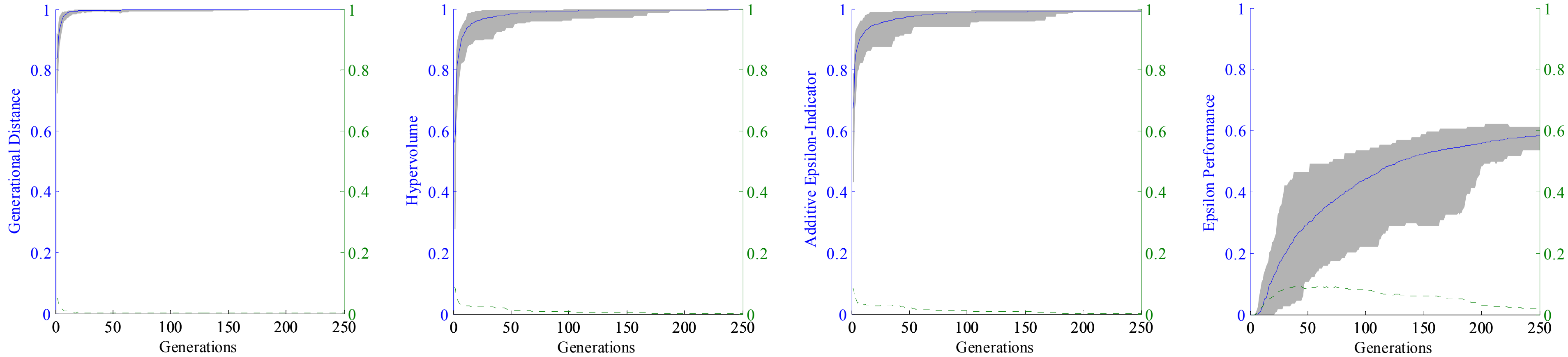
GALAXY



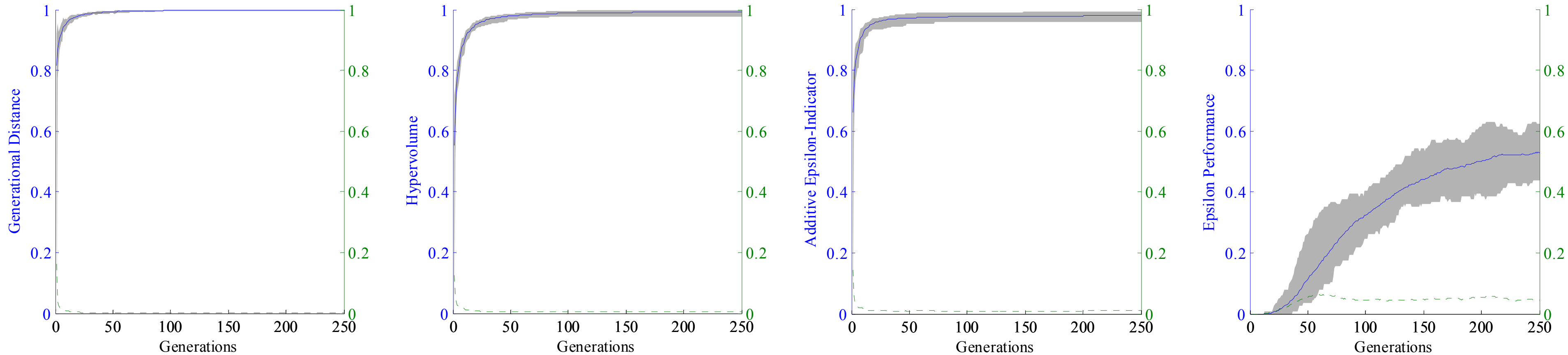
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 50 100 150 200 250
0 0.2 0.4 0.6 0.8 1

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 50 100 150 200 250
0 0.2 0.4 0.6 0.8 1

ε-MOEA

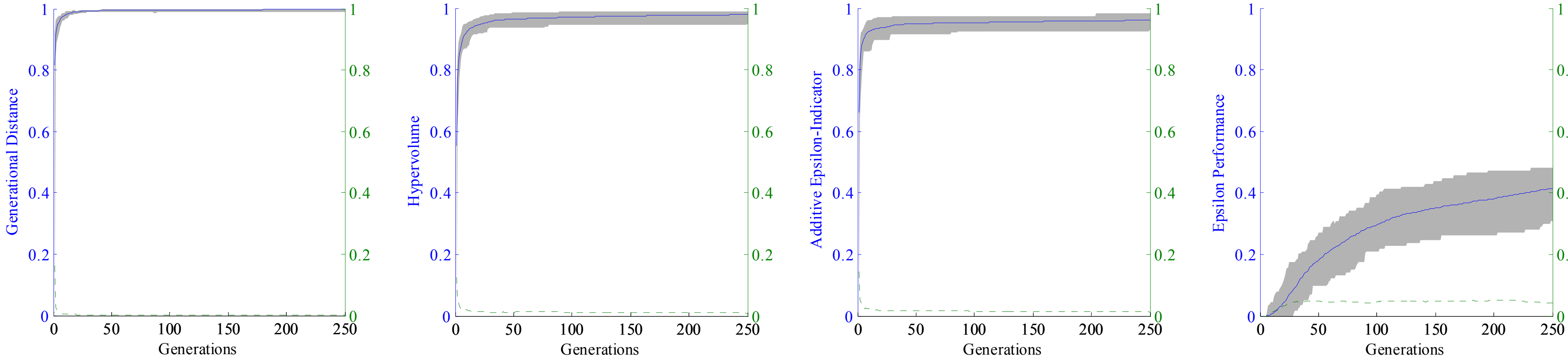
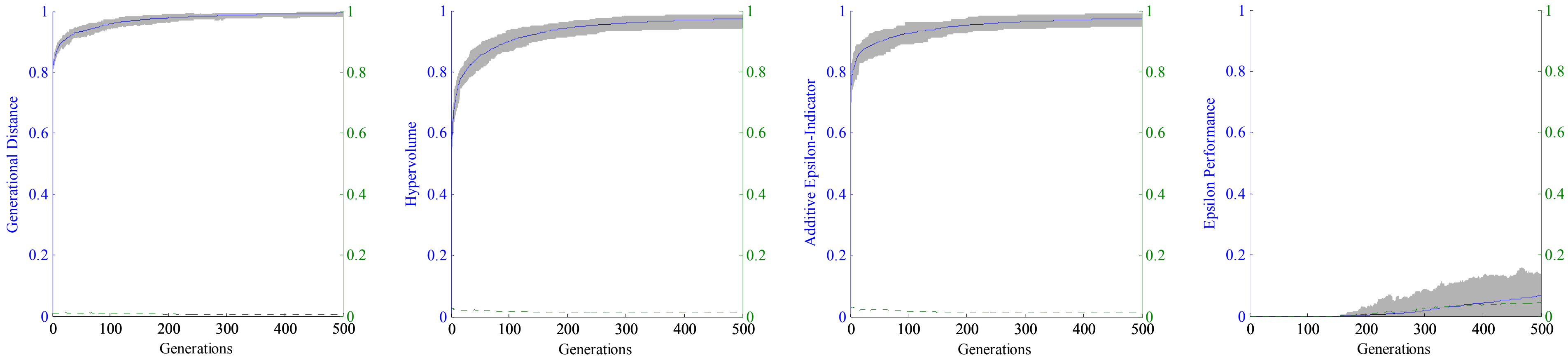
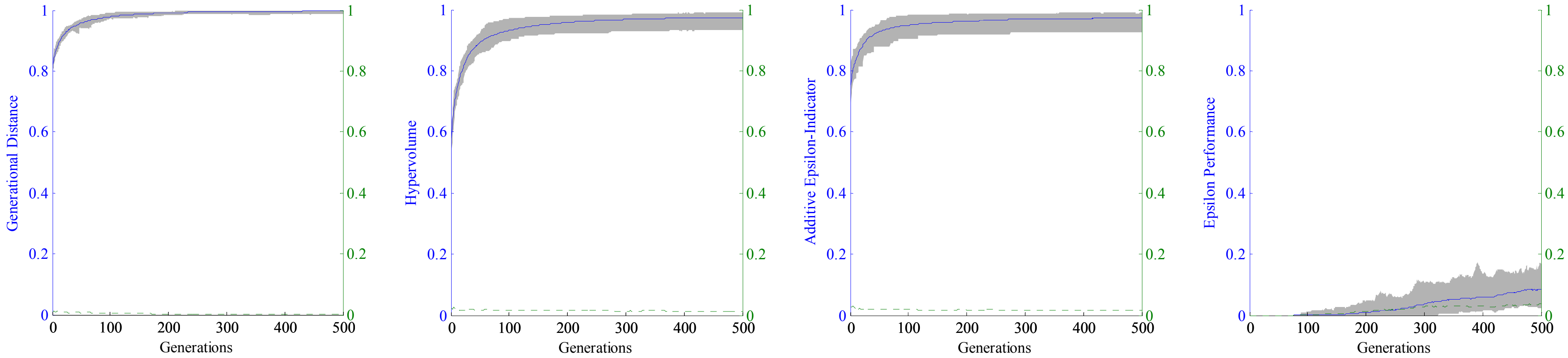


Figure D.10 Dynamic performances of various MOEAs for the TLN problem

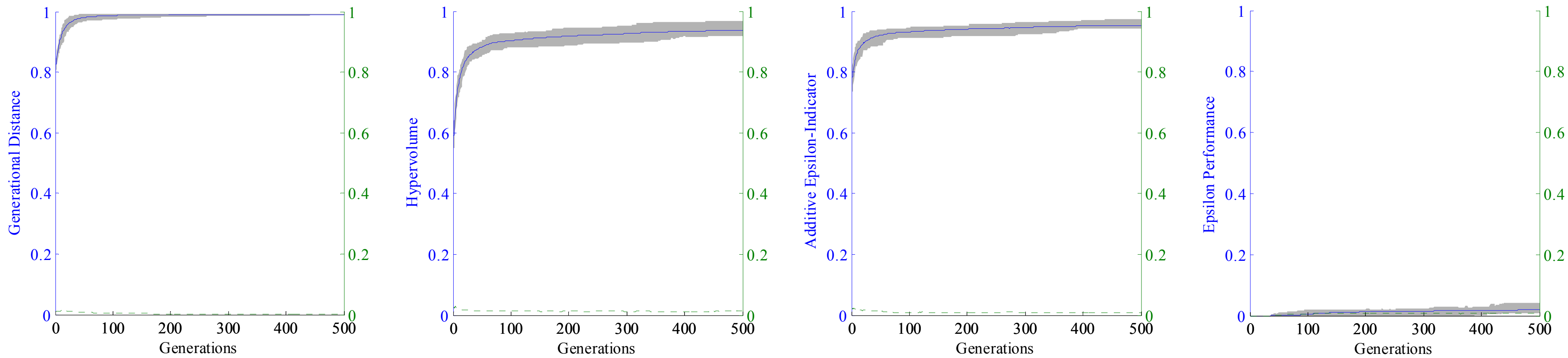
GALAXY



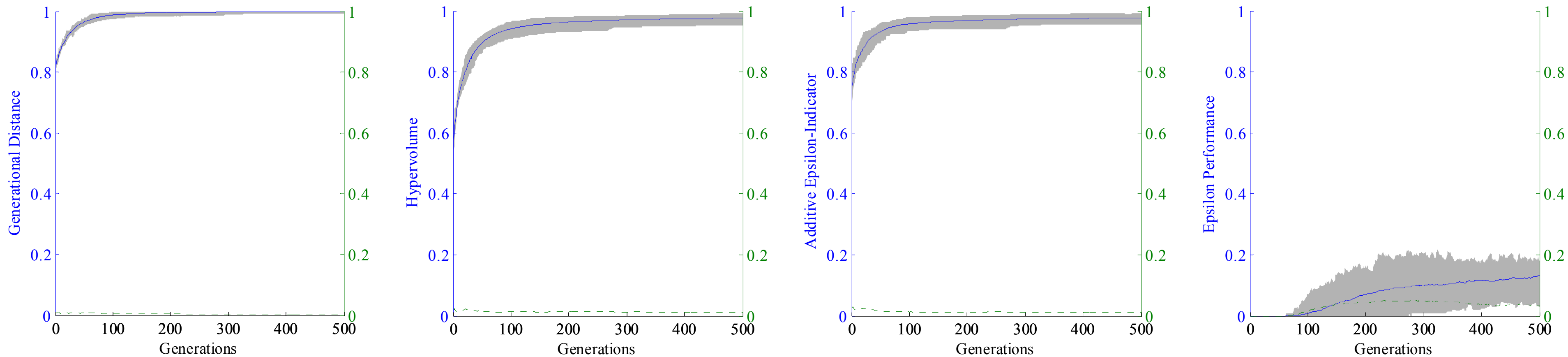
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 100 200 300 400 500
0 0.2 0.4 0.6 0.8 1

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 100 200 300 400 500
0 0.2 0.4 0.6 0.8 1

ε-MOEA

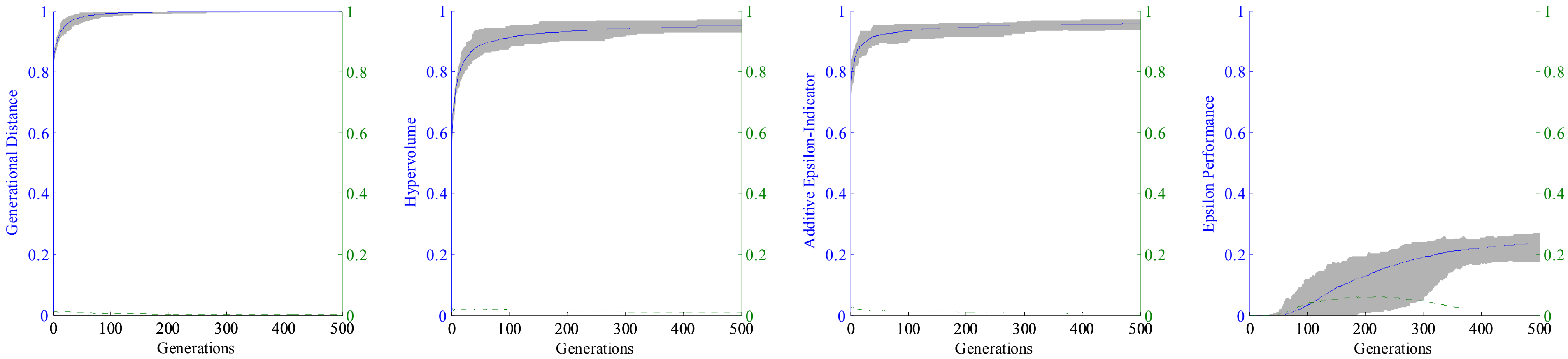
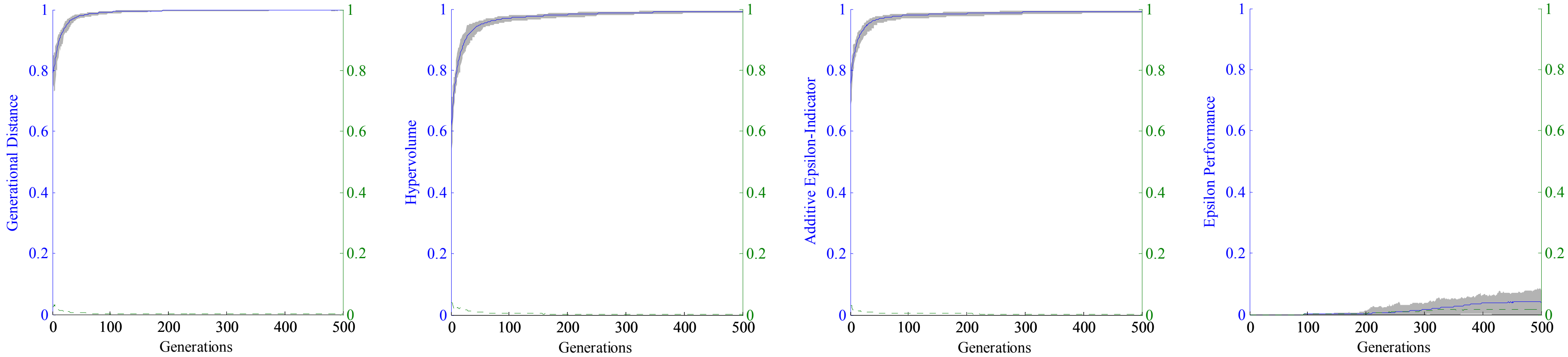
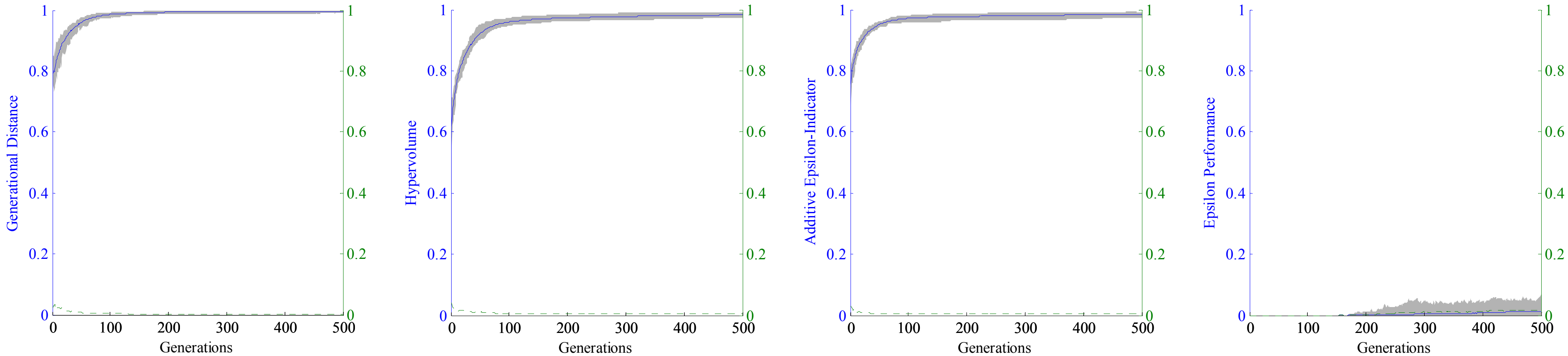


Figure D.11 Dynamic performances of various MOEAs for the NYT problem

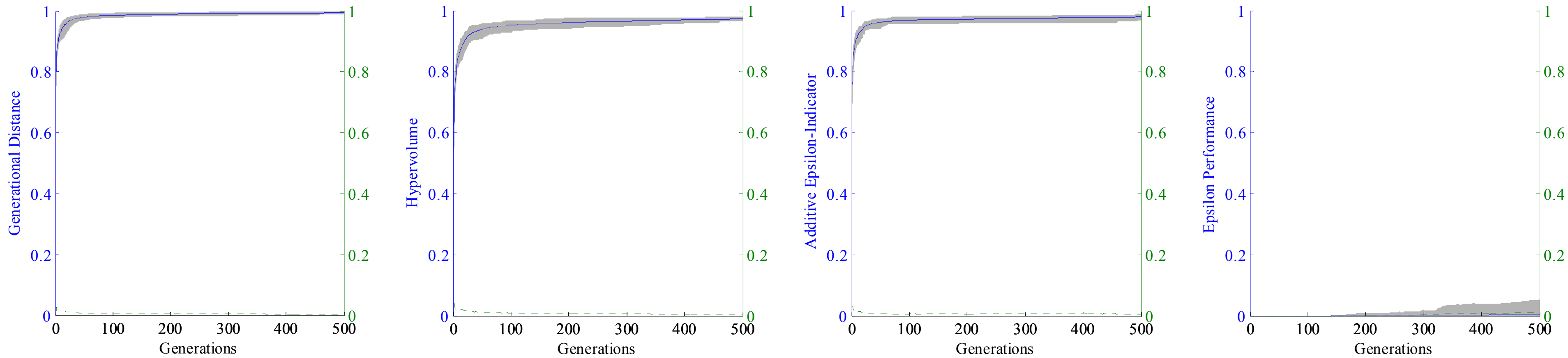
GALAXY



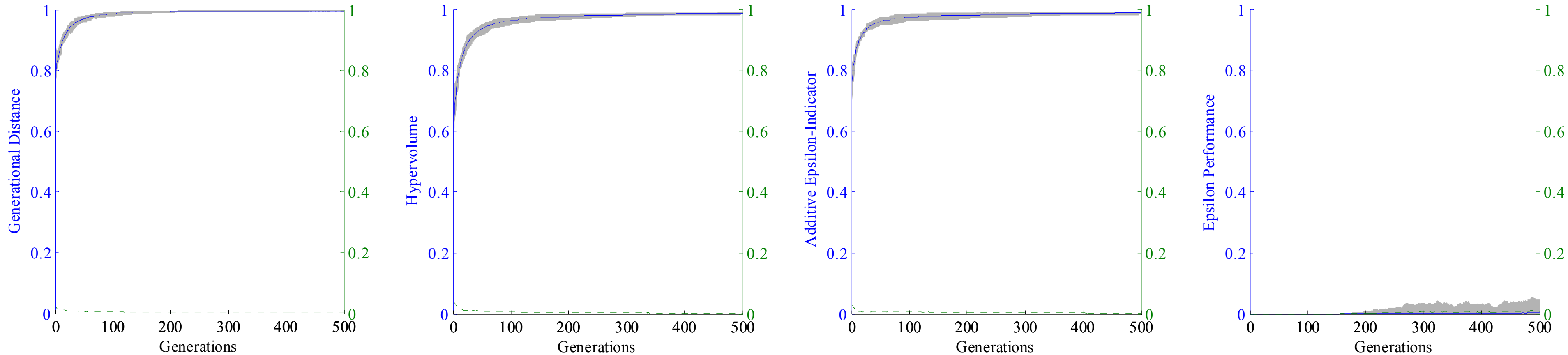
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
0 0.2 0.4 0.6 0.8 1
0 100 200 300 400 500
Generations
Generations
Generations
Generations

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
0 0.2 0.4 0.6 0.8 1
0 100 200 300 400 500
Generations
Generations
Generations
Generations

ε-MOEA

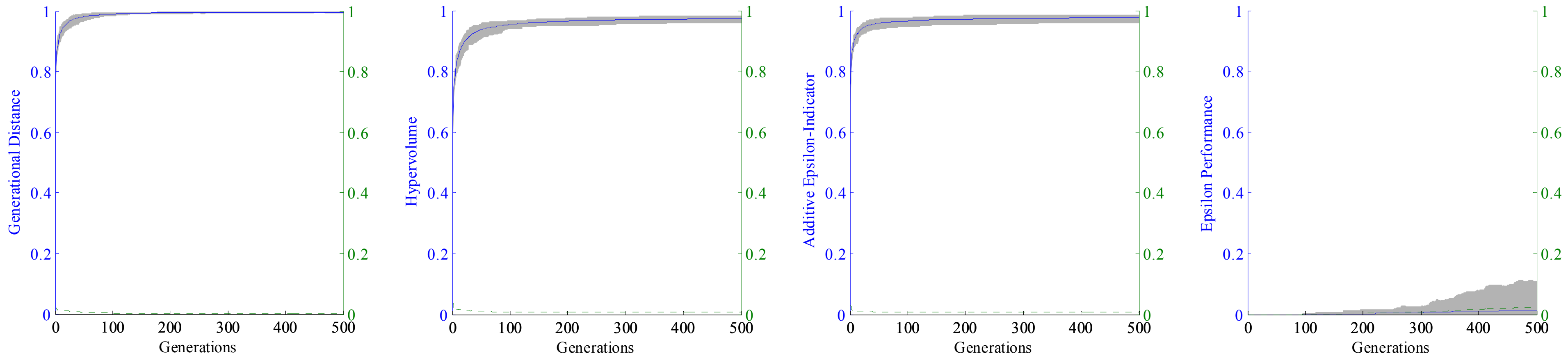
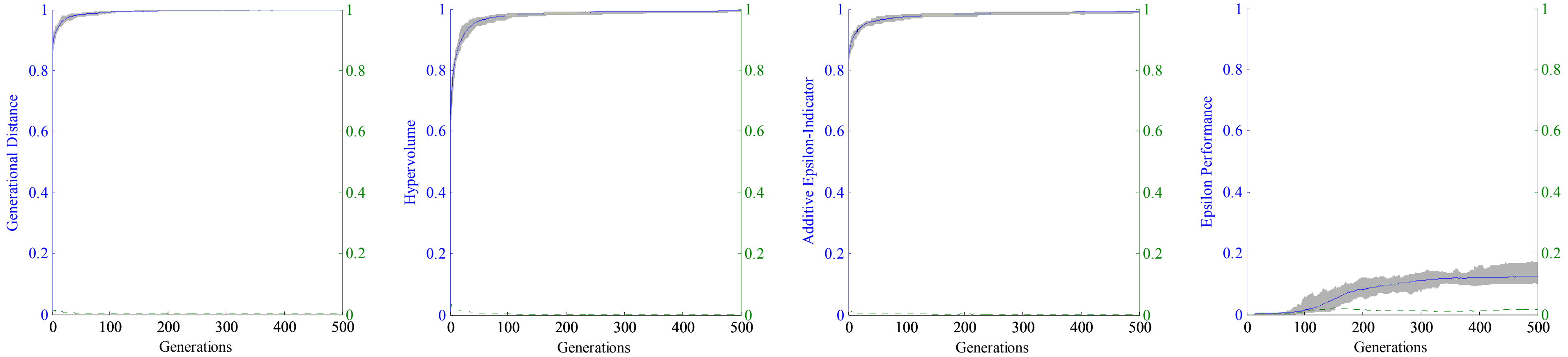
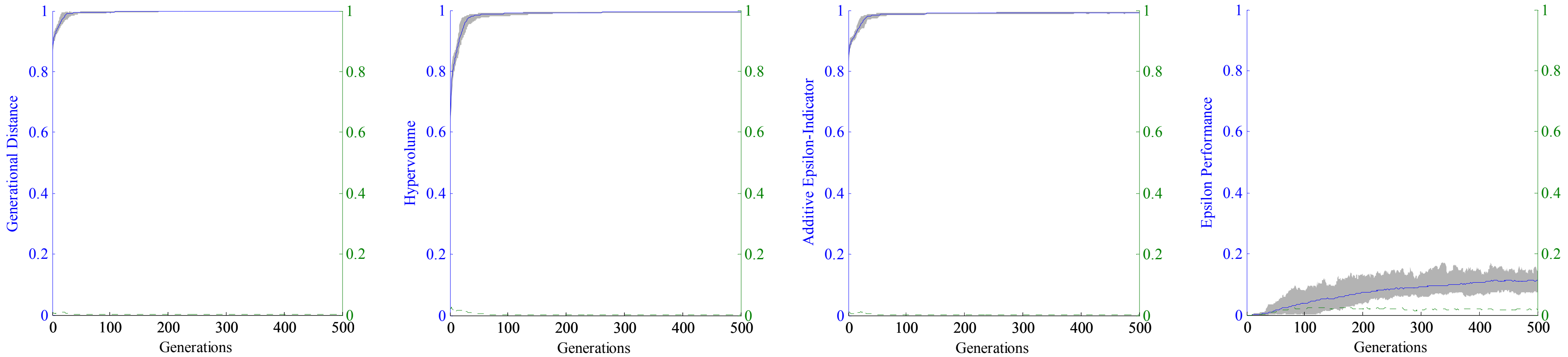


Figure D.12 Dynamic performances of various MOEAs for the BLA problem

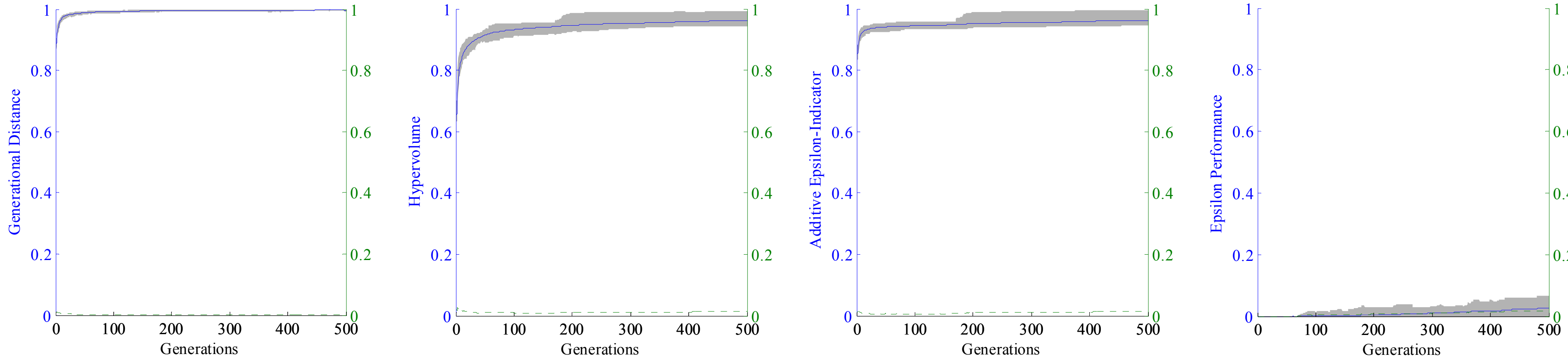
GALAXY



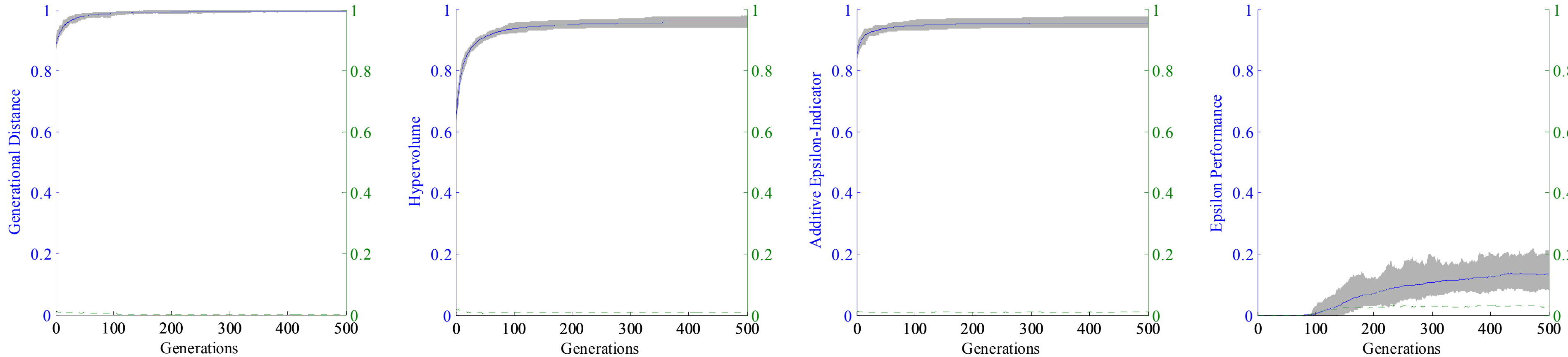
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0
100
200
300
400
500
0
0.2
0.4
0.6
0.8
1

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0
100
200
300
400
500
0
0.2
0.4
0.6
0.8
1

ε-MOEA

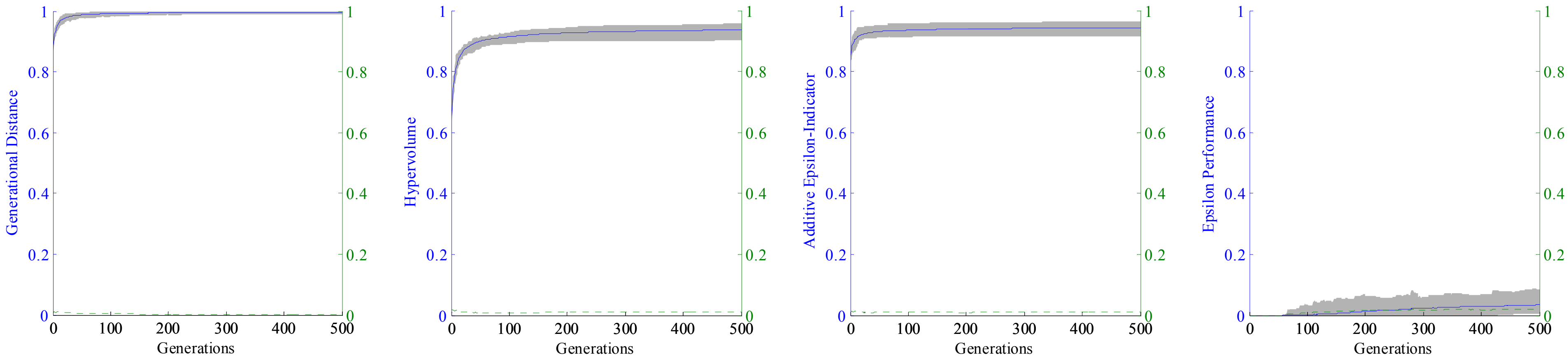
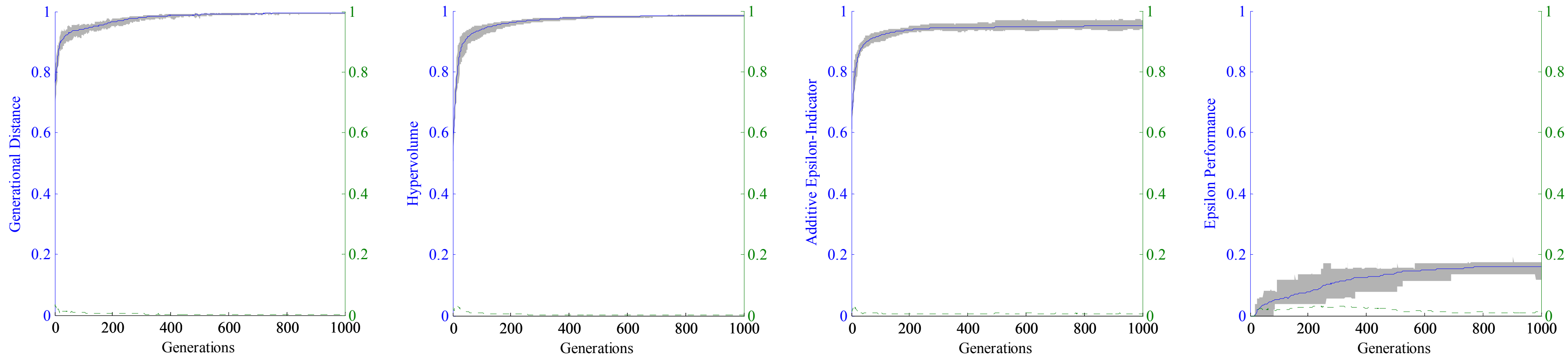
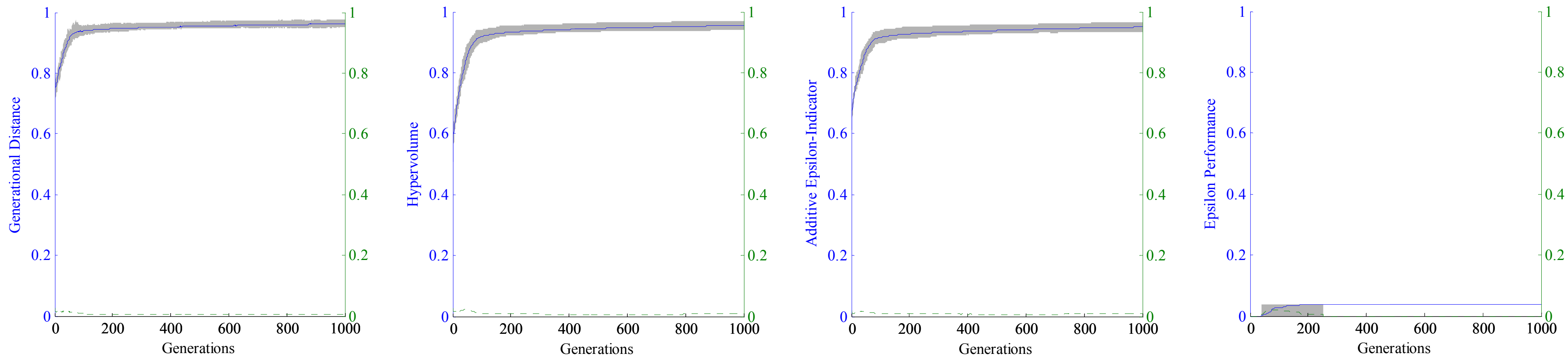


Figure D.13 Dynamic performances of various MOEAs for the GOY problem

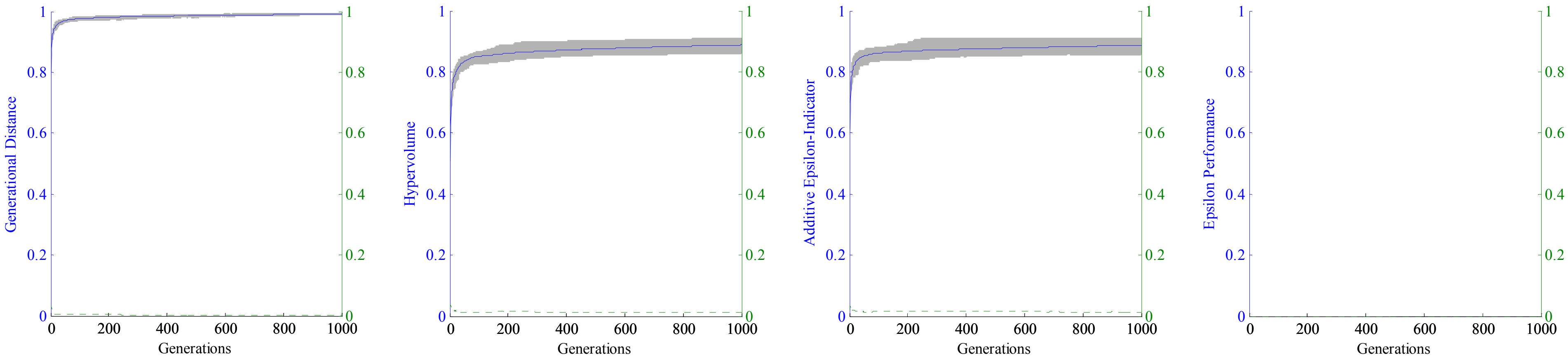
GALAXY



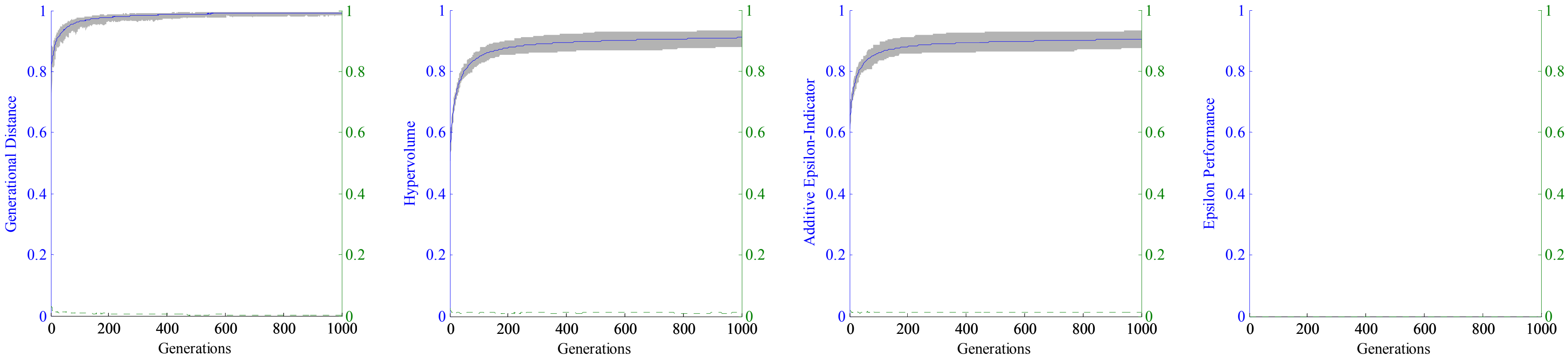
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 200 400 600 800 1000
0 0.2 0.4 0.6 0.8 1

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 200 400 600 800 1000
0 0.2 0.4 0.6 0.8 1

ε-MOEA

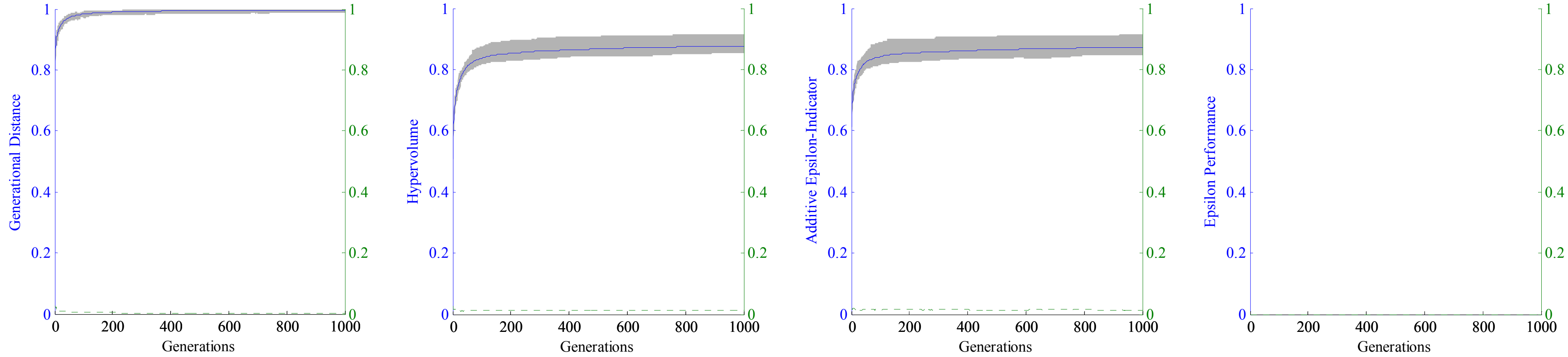
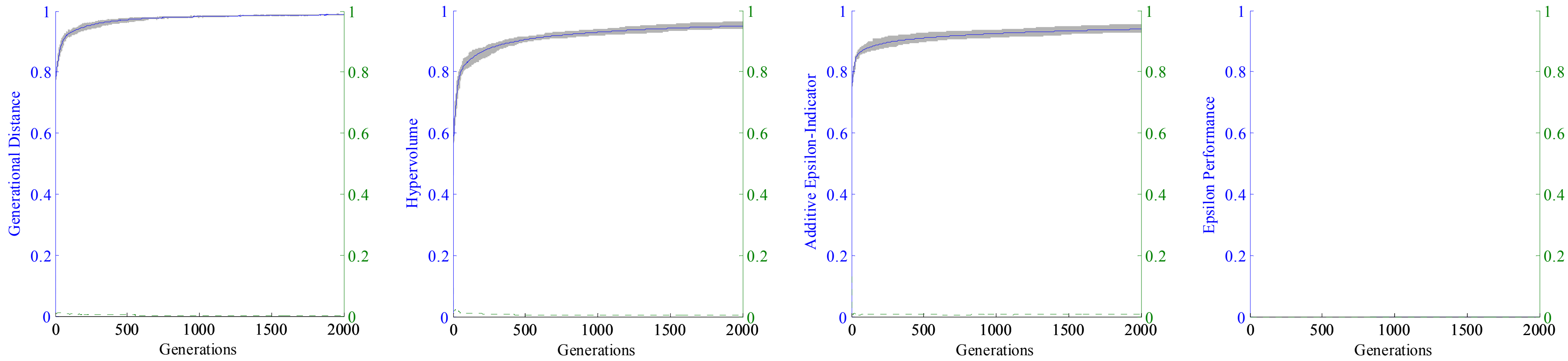
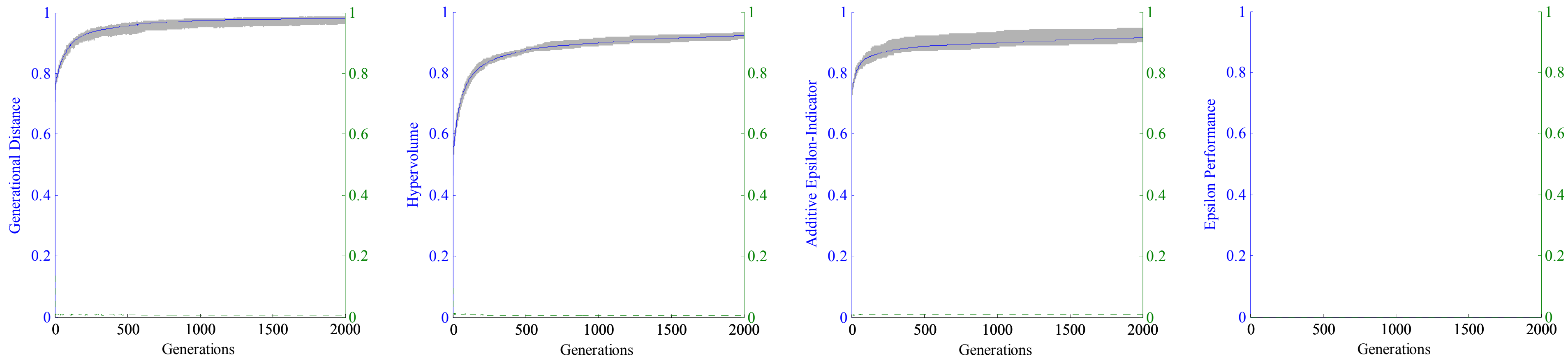


Figure D.14 Dynamic performances of various MOEAs for the FOS problem

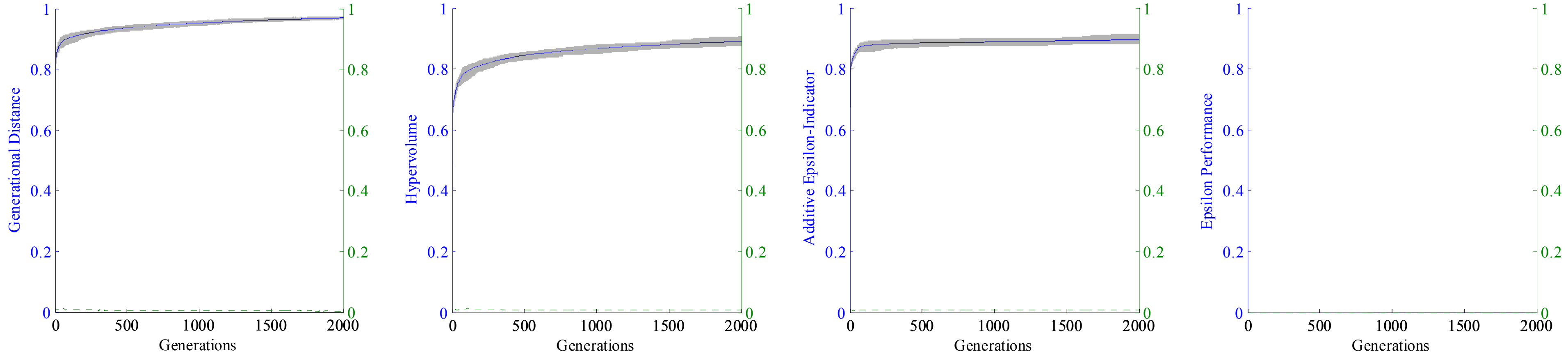
GALAXY



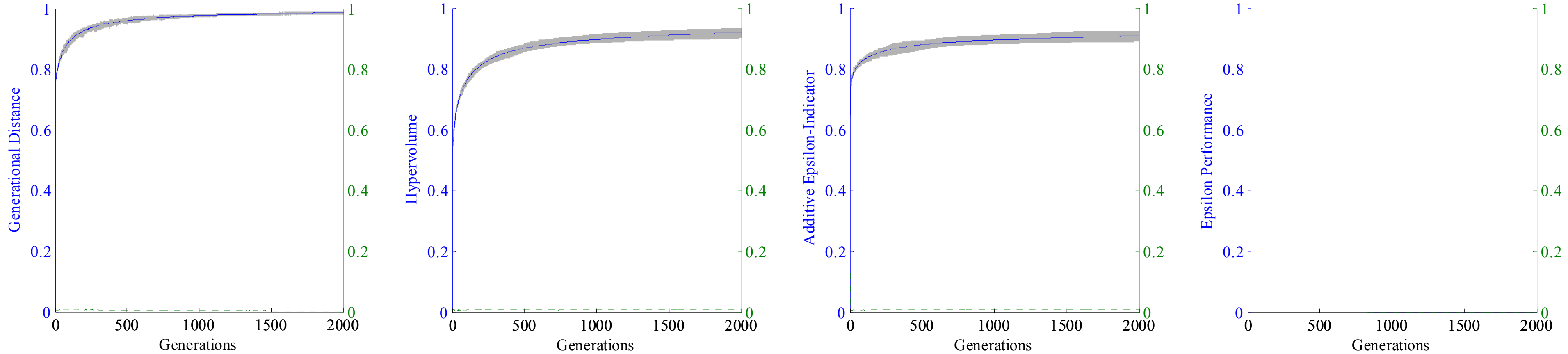
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
0 0.2 0.4 0.6 0.8 1
0 500 1000 1500 2000
Generations

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
0 0.2 0.4 0.6 0.8 1
0 500 1000 1500 2000
Generations

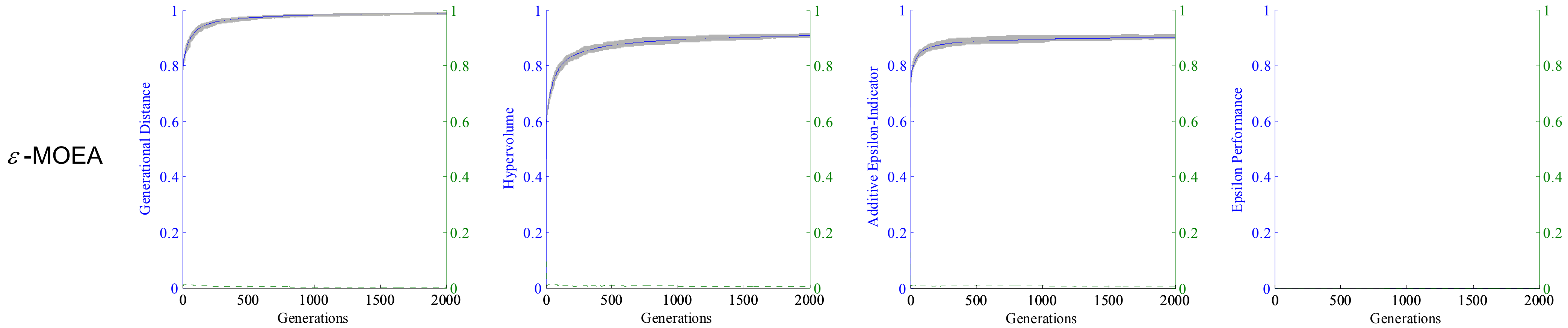
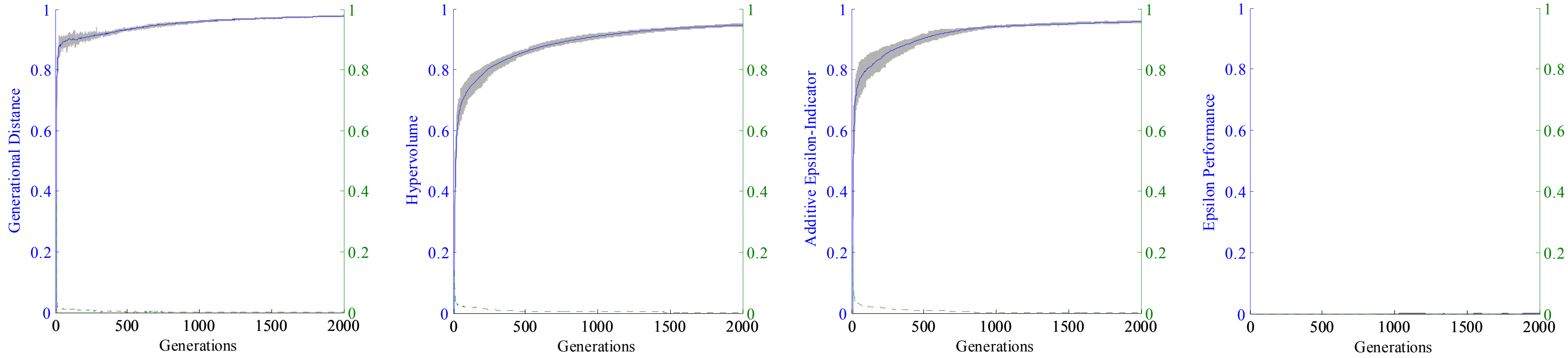
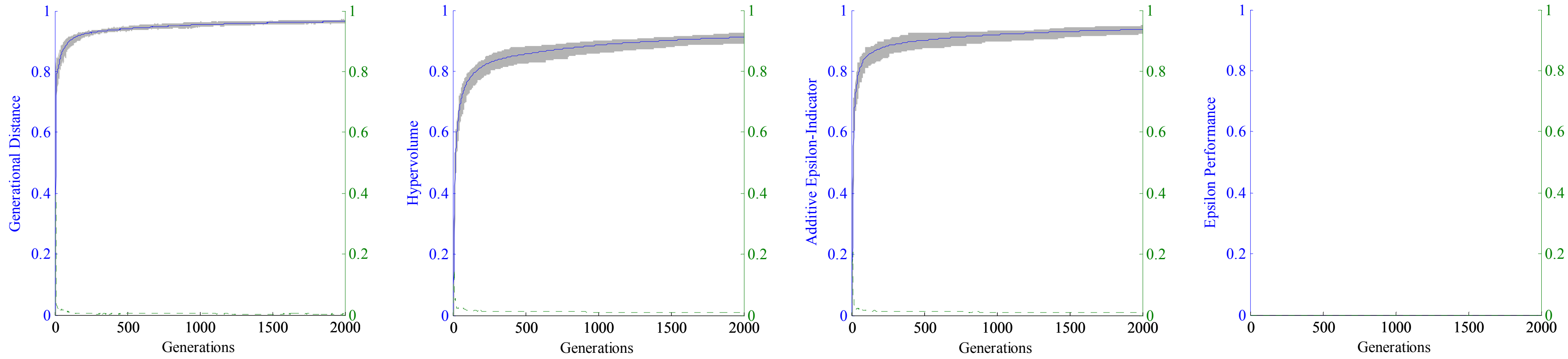


Figure D.15 Dynamic performances of various MOEAs for the MOD problem

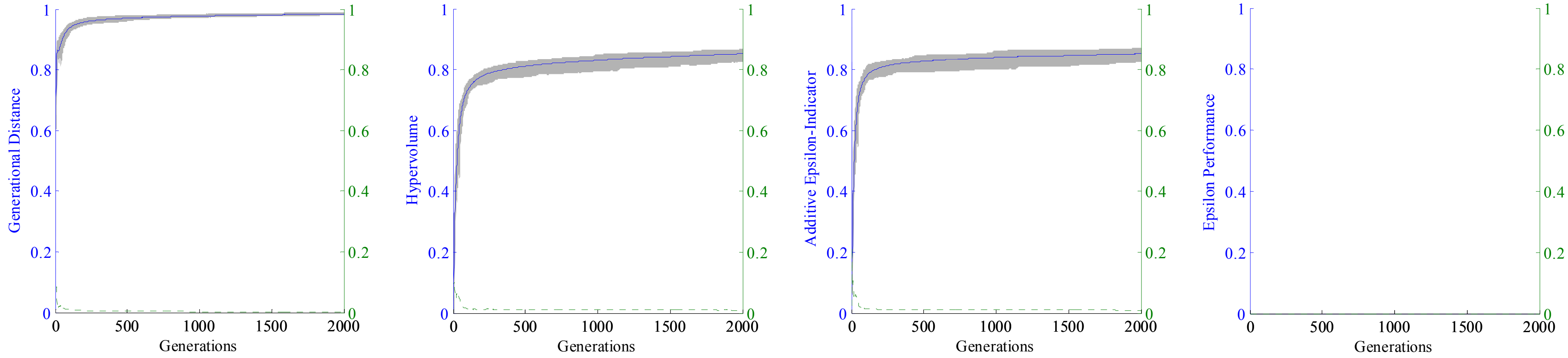
GALAXY



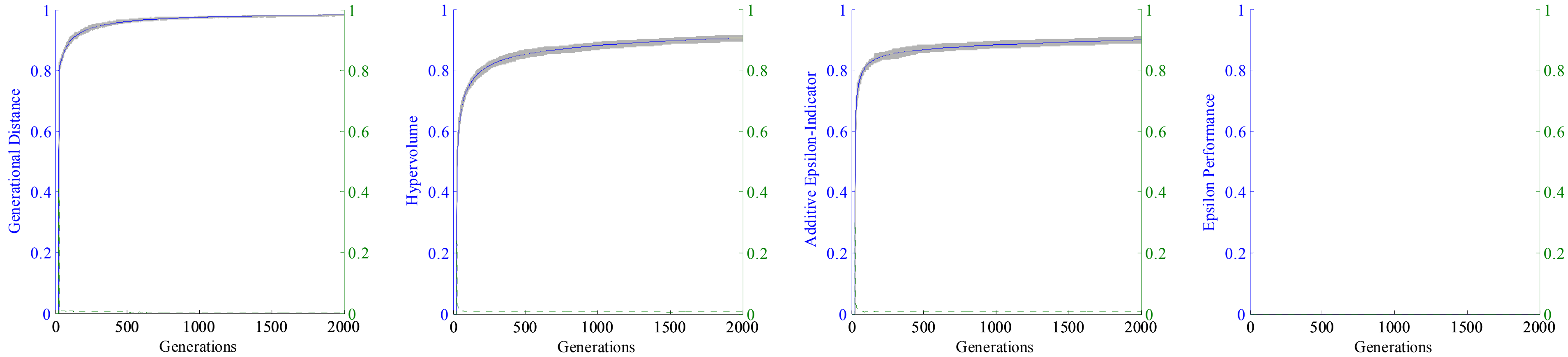
AMALGAM



Borg


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 500 1000 1500 2000
0 0.2 0.4 0.6 0.8 1

NSGA-II


Generational Distance
Hypervolume
Additive Epsilon-Indicator
Epsilon Performance
Generations
0 500 1000 1500 2000
0 0.2 0.4 0.6 0.8 1

ε-MOEA

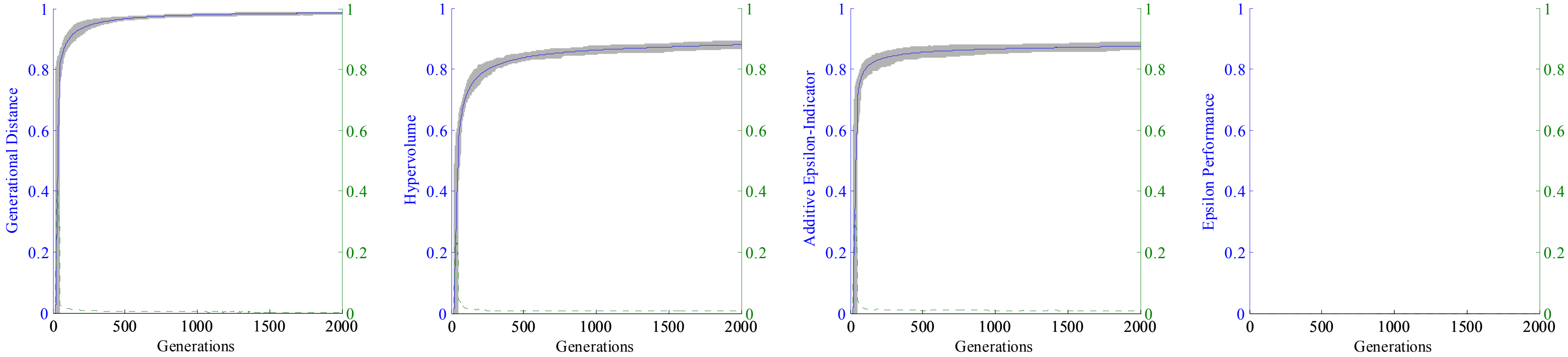
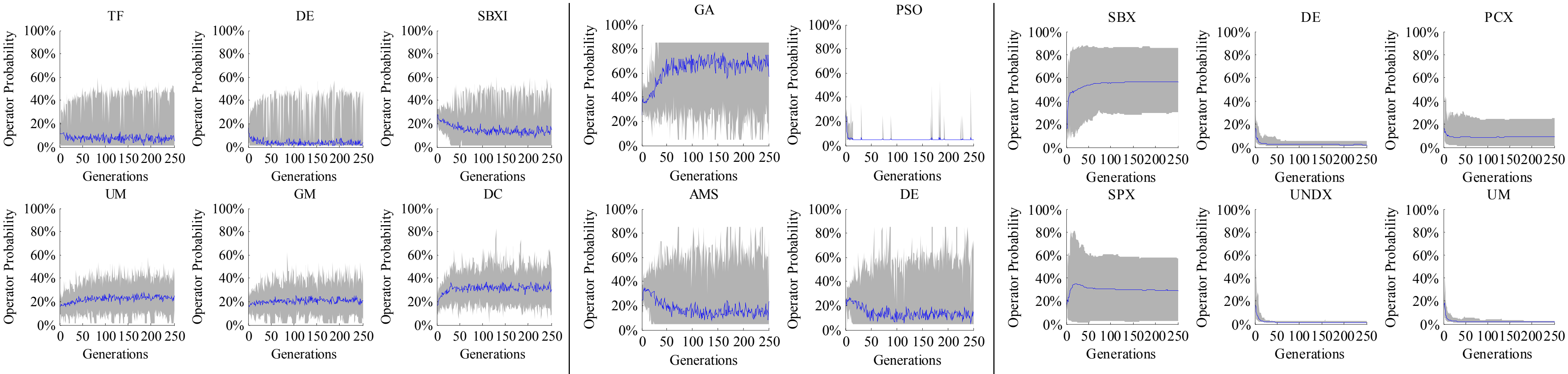


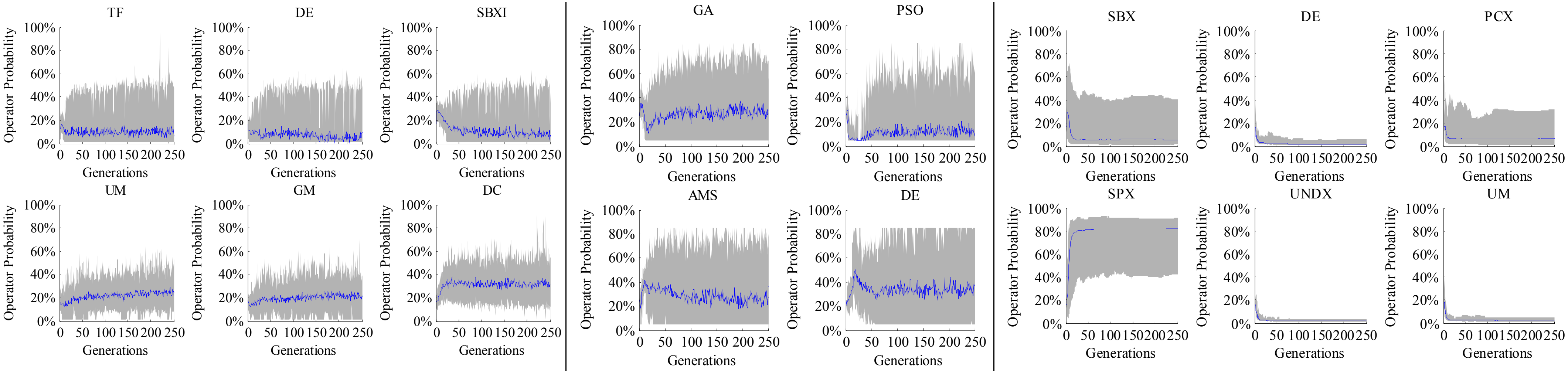
Figure D.16 Dynamic performances of various MOEAs for the BIN problem

D.3 Dynamic Variations of Search Operators



(a) GALAXY (b) AMALGAM (c) Borg

Figure D.17 Dynamic performances of search operators of hybrid algorithms for the TRN problem



(a) GALAXY (b) AMALGAM (c) Borg

Figure D.18 Dynamic performances of search operators of hybrid algorithms for the TLN problem

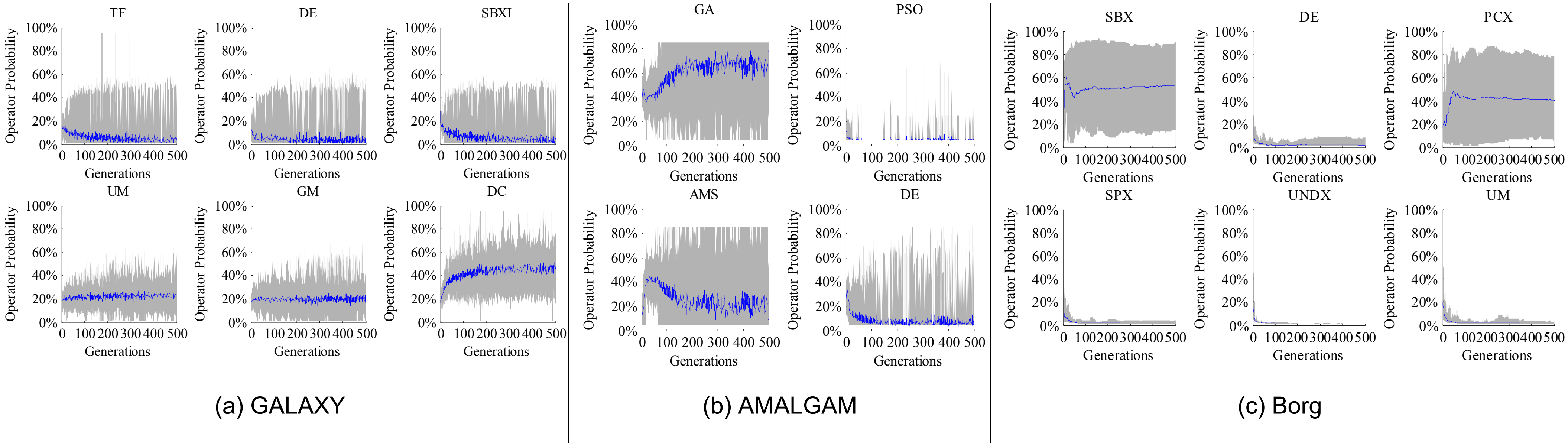


Figure D.19 Dynamic performances of search operators of hybrid algorithms for the NYT problem

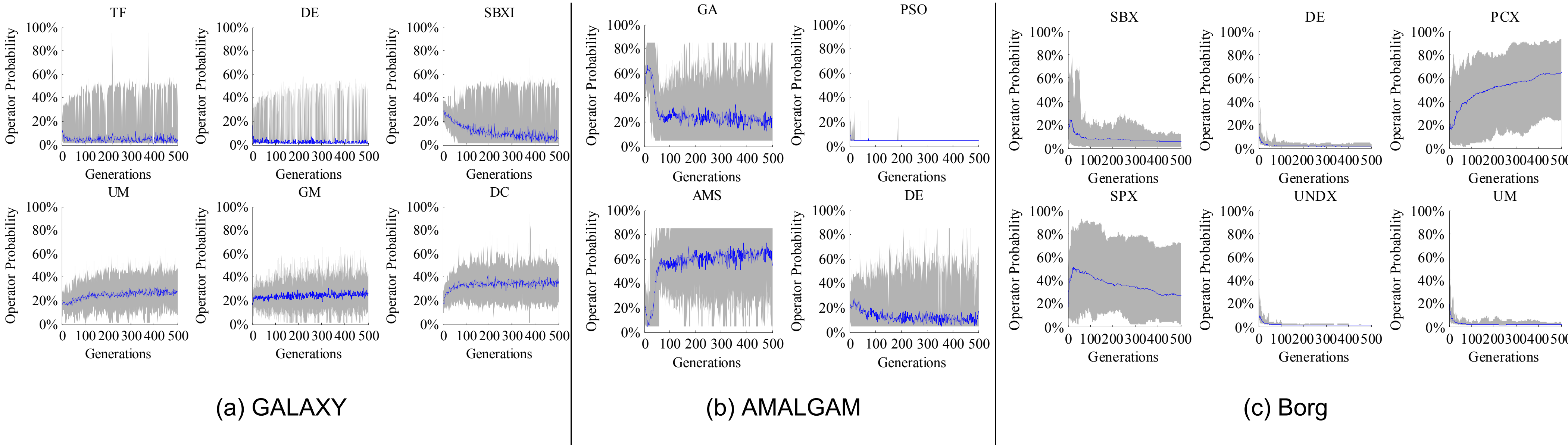


Figure D.20 Dynamic performances of search operators of hybrid algorithms for the BLA problem

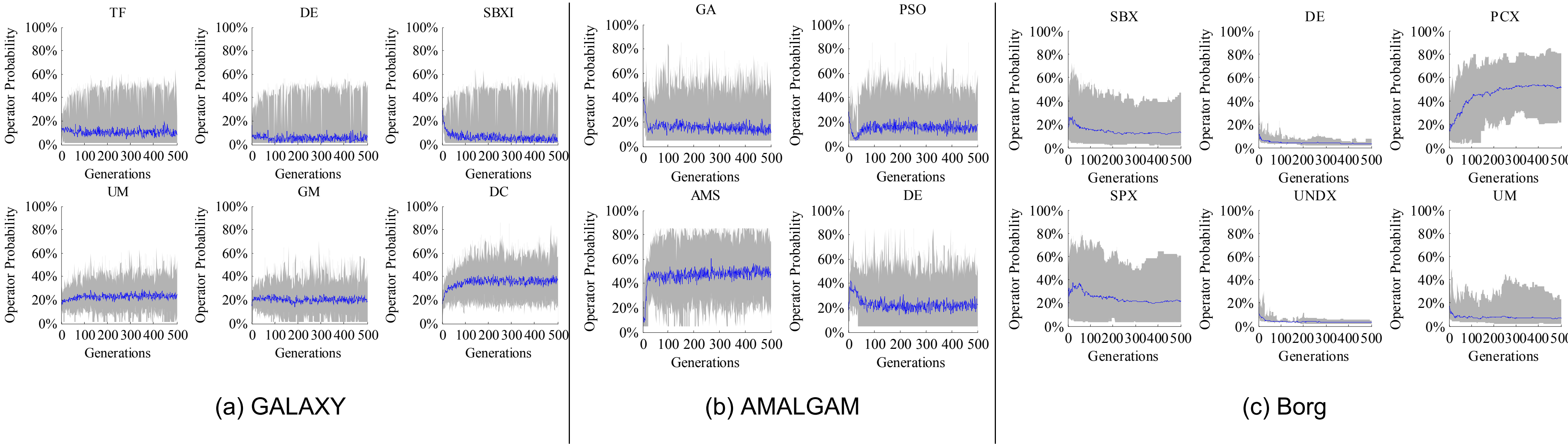
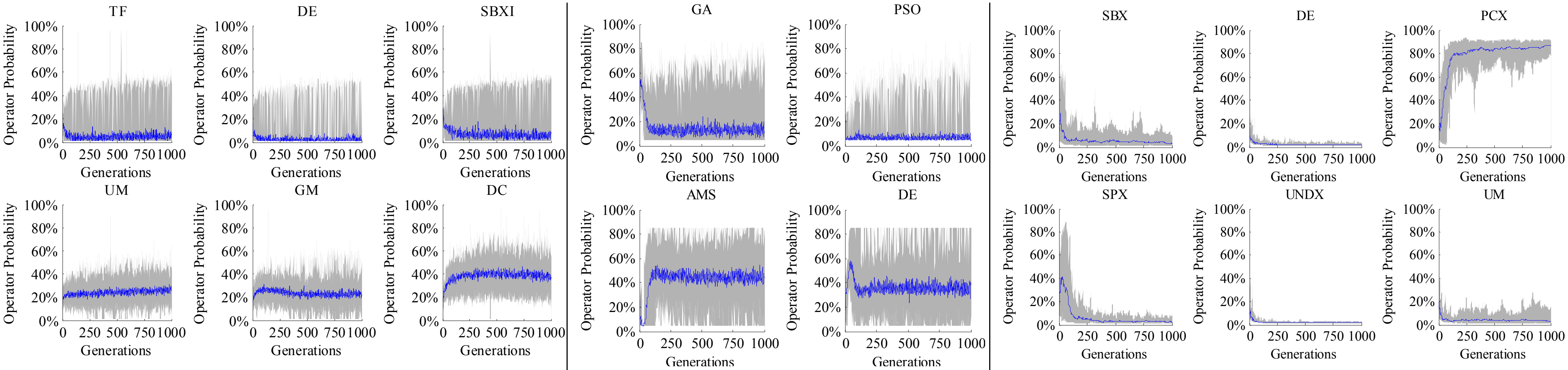


Figure D.21 Dynamic performances of search operators of hybrid algorithms for the GOY problem



(a) GALAXY (b) AMALGAM (c) Borg

Figure D.22 Dynamic performances of search operators of hybrid algorithms for the FOS problem

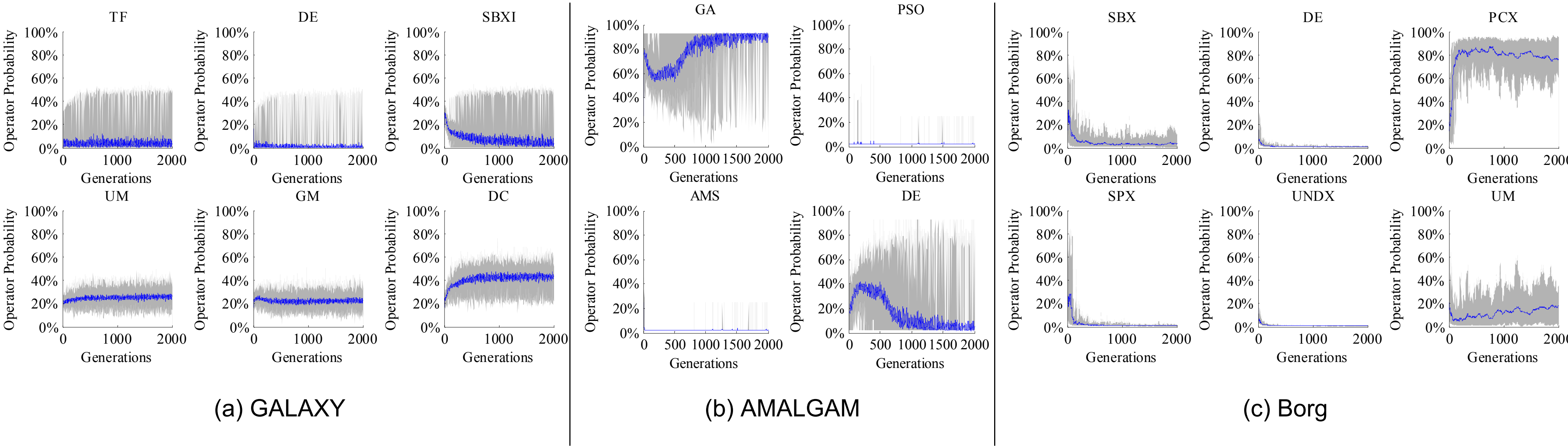


Figure D.23 Dynamic performances of search operators of hybrid algorithms for the MOD problem

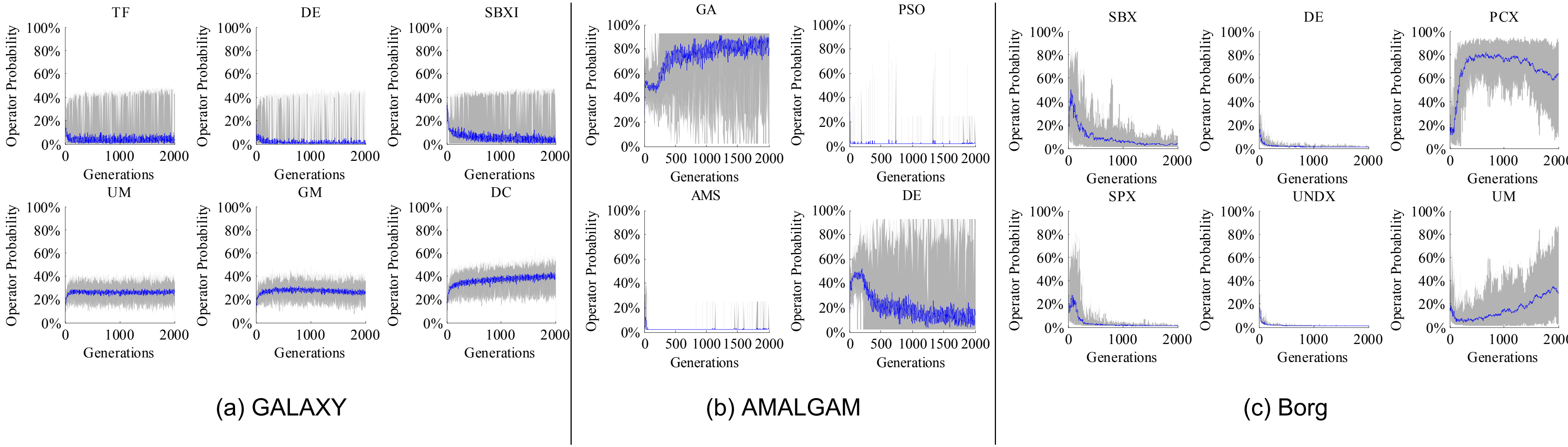


Figure D.24 Dynamic performances of search operators of hybrid algorithms for the BIN problem

BIBLIOGRAPHY

Papers Presented by the Candidate

Wang, Q., Creaco, E., Franchini, M., Savić, D. A., and Kapelan, Z. (2014). “Comparing Low and High-Level Hybrid Algorithms on the Two-Objective Optimal Design of Water Distribution Systems.” *Water Resources Management*, (submitted for publication).

Wang, Q., Guidolin, M., Savić, D. A., and Kapelan, Z. (2014). “Two-Objective Design of Benchmark Problems of Water Distribution System via MOEAs: Towards the Best-Known Approximation to the True Pareto Front.” *Journal of Water Resources Planning and Management*, (accepted).

Wang, Q., and Liu, S. (2014). “Comparison of initialization methods for multi-objective design of Water Distribution Systems.” *Journal of Computing in Civil Engineering*, (submitted for publication).

Wang, Q., Savić, D. A., and Kapelan, Z. (2014). “Hybrid metaheuristics for multi-objective design of water distribution systems.” *Journal of Hydroinformatics*, 16(1), 165-177.

Marchi, A., Salomons, E., Ostfeld, A., Kapelan, Z., Simpson, A., Zecchin, A., Maier, H., Wu, Z., Elsayed, S., Song, Y., Walski, T., Stokes, C., Wu, W., Dandy, G., Alvisi, S., Creaco, E., Franchini, M., Saldarriaga, J., Páez, D., Hernández, D., Bohórquez, J., Bent, R., Coffrin, C., Judi, D., McPherson, T., van Hentenryck, P., Matos, J., Monteiro, A., Matias, N., Yoo, D., Lee, H., Kim, J., Iglesias-Rey, P., Martínez-Solano, F., Mora-Meliá, D., Ribelles-Aguilar, J., Guidolin, M., Fu, G., Reed, P., Wang, Q., Liu, H., McClymont, K., Johns, M., Keedwell, E., Kandiah, V., Jasper, M., Drake, K., Shafiee, E., Barandouzi, M., Berglund, A., Brill, D., Mahinthakumar, G., Ranjithan, R., Zechman, E., Morley, M., Tricarico, C., de Marinis, G., Tolson, B., Khedr, A., and Asadzadeh, M. (2013). "Battle of the Water Networks II." *Journal of Water Resources Planning and Management*, (Just Released).

Wang Q., Liu, S., Liu, W., Kapelan, Z., and Savić, D. A. (2013). “Decision Support System for emergency scheduling of raw water supply systems with multiple sources.” *Frontiers of Environmental Science and Engineering*, 7(5), 777-786.

Wang, Q., Liu, S., Wang, H., and Savić, D. A. (2012). “Multi-objective cuckoo search for the optimal design of water distribution systems.” *Proceedings of 2nd International Conference on Civil Engineering and Urban Planning*, Yantai, China.

Wang, Q., Liu, H., McClymont, K., Johns, M., and Keedwell, E. (2012). “A hybrid of multi-phase optimisation and iterated manual intervention for BWN-II.” *Proceedings of 14th Water Distribution Systems Analysis Conference*, Adelaide, Australia.

Wang, Q., Savić, D. A., and Kapelan, Z. (2012). “Hybrid optimisation algorithms for multi-objective design of water distribution systems.” *Proceedings of 10th International Conference on Hydroinformatics*, Hamburg, Germany.

List of References

Abbass, H. A. "The self-adaptive pareto differential evolution algorithm." *CEC'02. Proceedings of the 2002 Congress on Evolutionary Computation*, 831-836.

Alperovits, E., and Shamir, U. (1977). "Design of optimal water distribution systems." *Water Resources Research*, 13(6), 885-900.

Ang, W. H., and Jowitt, P. W. (2006). "Solution for water distribution systems under pressure-deficient conditions." *Journal of Water Resources Planning and Management* 132(3), 175-182.

Awad, A., and Poser, I. V. (2005). "Genetic Algorithm Optimization of Water Supply Networks." *Water Intelligence Online*, 4(8), 1-25.

Babayan, A., Kapelan, Z., Savic, D., and Walters, G. (2005). "Least-Cost Design of Water Distribution Networks under Demand Uncertainty." *Journal of Water Resources Planning and Management*, 131(5), 375-382.

Beume, N., and Rudolph, G. (2006). "Faster S-metric calculation by considering dominated hypervolume as Klee's measure problem." Second IASTED Conference on Computational Intelligence, 231-236.

Bhave, P. R. (1991). *Analysis of flow in water distribution networks*, Technomic Publishing, Lancaster.

Bradstreet, L., Barone, L., While, L., Huband, S., and Hingston, P. "Use of the WFG Toolkit and PISA for Comparison of MOEAs." *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, 382-389.

Bragalli, C., D'Ambrosio, C., Lee, J., Lodi, A., and Toth, P. (2008). "Water Network Design by MINLP." Rep. No. RC24495, IBM Research, Yorktown Heights, NY.

Chaiyaratana, N., Piroonratana, T., and Sangkawelert, N. (2007). "Effects of diversity control in single-objective and multi-objective genetic algorithms." *Journal of Heuristics* 13(1), 1-34.

Cheung, P. B., Reis, L. F. R., Formiga, K. T. M., Chaudhry, F. H., and Ticona, W. G. C. (2003). "Multiobjective evolutionary algorithms applied to the rehabilitation of a water distribution system: a comparative study." The 2nd International Conference on Evolutionary Multi-criterion Optimization, Springer-Verlag, Faro, Portugal, 662-676.

Cheung, P. B., Van Zyl, J. E., and Reis, L. F. R. (2005). "Extension of epanet for pressure driven demand modeling in water distribution system." *Proc., Computer and Control in Water Industry, Water Management for the 21st Century*, 1(2).

Coello, C. A. C. (2006). "Evolutionary multi-objective optimization: a historical view of the field." *IEEE Computational Intelligence Magazine*, 1(1), 28-36.

Coello, C. A. C. (2006). "Evolutionary multi-objective optimization: a historical view of the field." *IEEE Computational Intelligence Magazine*, 1(1), 28-36.

Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J. (2001). "PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization."

Proceedings of the Genetic and Evolutionary Computation Conference In: GECCO, vol. 1, Morgan Kaufmann Publishers, 283-290.

Creaco, E., and Franchini, M. (2013). "Low level hybrid procedure for the multi-objective design of water distribution networks." 12th International Conference on Computing and Control for the Water Industry, CCWI2013, Elsevier, Italy.

Črepinšek, M., Liu, S.-H., and Mernik, M. (2013). "Exploration and exploitation in evolutionary algorithms: a survey." *ACM Computing Surveys (CSUR)*, 45(3), 35.

Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). "An improved genetic algorithm for pipe network optimization." *Water Resources Research*, 32(2), 449-458.

Das, S., Konar, A., and Chakraborty, U. K. "Two improved differential evolution schemes for faster global search." *Proceedings of the 2005 conference on Genetic and evolutionary computation*, 991-998.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK.

Deb, K., and Agrawal, R. B. (1995). "Simulated Binary Crossover for Continuous Search Space." *Complex Systems*, 9, 115-148.

Deb, K., and Jain, S. (2002). "Running performance metrics for evolutionary multi-objective optimization." *2002004*, KanGAL Indian Institute of Technology, Kanpur.

Deb, K., Mohan, M., and Mishra, S. (2003). "Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions." Evolutionary Multi-Criterion Optimization, C. M. Fonseca, P. J. Fleming, E. Zitzler, and L. Thiele, eds., Springer Berlin Heidelberg, 222-236.

Deb, K., Mohan, M., and Mishra, S. (2005). "Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions." *Evolutionary Computation*, 13(4), 501-525.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

Di Pierro, F., Khu, S.-T., Savic, D., and Berardi, L. (2006). "Metaheuristic Algorithms For Multi-objective Optimal Design Of Water Distribution Networks." 7th International Conference on Hydroinformatics, Nice, France.

Di Pierro, F., Khu, S.-T., Savic, D., and Berardi, L. (2009). "Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms." *Environmental Modelling & Software*, 24(2), 202-213.

Dong, X., Liu, S., Tao, T., Li, S., and XIN, K. (2012). "A comparative study of differential evolution and genetic algorithms for optimizing the design of water distribution systems." *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, 13(9), 674-686.

Eberhardt, R., Shi, Y., and Kennedy, J. (2001). *Swarm Intelligence*, Morgan Kaufmann, San Mateo, CA.

Engelhardt, M. O., Skipworth, P. J., Savic, D. A., Saul, A. J., and Walters, G. A. (2000). "Rehabilitation strategies for water distribution networks: a literature review with a UK perspective." *Urban Water*, 2(2), 153-170.

Farmani, R., Savic, D. A., and Walters, G. A. (2003a). "Benchmark problems for design and optimisation of water distribution systems." ADVANCES IN WATER SUPPLY MANAGEMENT, Maksimovic, Butler, and Memon, eds., 249-256.

Farmani, R., Savic, D. A., and Walters, G. A. (2003b). "Multi-objective optimization of water system: A comparative study." Pumps, Electromechanical Devices and Systems Applied to Urban Water Management, C. a. C. Jr., ed., pp. 247-256.

Farmani, R., Savic, D. A., and Walters, G. A. (2004). ""EXNET" Benchmark Problem for Multi-Objective Optimization of Large Water Systems." Modelling and Control for Participatory Planning and Managing Water Systems, IFAC workshop, Venice, Italy.

Farmani, R., Savic, D. A., and Walters, G. A. (2005a). "Evolutionary multi-objective optimization in water distribution network design." *Engineering Optimization*, 37(2), 167-183.

Farmani, R., Walters, G., and Savic, D. (2006). "Evolutionary multi-objective optimization of the design and operation of water distribution network : total cost vs. reliability vs. water quality." *Journal of hydroinformatics*, 8(3), 165-179.

Farmani, R., Walters, G. A., and Savic, D. A. (2005b). "Trade-Off between Total Cost and Reliability for Anytown Water Distribution Network." *Journal of Water Resources Planning and Management*, 131(3), 161-171.

Fogel, D. B. (2006). *Evolutionary computation: toward a new philosophy of machine intelligence*, John Wiley & Sons.

Fonseca, C. M., and Fleming, P. J. "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization." *Genetic Algorithms: Proceedings of the Fifth International Conference*, 416-423.

Fonseca, C. M., Knowles, J. D., Thiele, L., and Zitzler, E. (2006). "A tutorial on the performance assessment of stochastic multiobjective optimizers, (Unpublished) " *TIK-Report No. 214*, Computer Engineering and Network Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

Fonseca, V. G. d., Fonseca, C. M., and Hall., A. (2001). "Inferential performance assessment of stochastic optimizers and the attainment function." Evolutionary Multi-criterion Optimization (EMO 2001), Lecture Notes in Computer Science, E. Zitzler, K. Deb, L. Thiele, C. C. Coello, and D. Corne, eds., Springer Verlag, Berlin, Germany.

Fu, G., Kapelan, Z., Kasprzyk, J., and Reed, P. (2012a). "Optimal Design of Water Distribution Systems Using Many-Objective Visual Analytics." *Journal of Water Resources Planning and Management*, 139(6), 624-633.

Fu, G., Kapelan, Z., and Reed, P. (2012b). "Reducing the Complexity of Multiobjective Water Distribution System Optimization through Global Sensitivity Analysis." *Journal of Water Resources Planning and Management*, 138(3), 196-207.

Fujiwara, O., and Khang, D. (1990). "A two-phase decomposition method for optimal design of looped water distribution networks." *Water Resources Research*, 26(4), 539-549.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian data analysis*, Chapmann and Hall/CRC, New York.

Gessler, J. (1985). "Pipe network optimization by enumeration." In Proc., Computer Applications for Water Resources, ASCE, New York, N.Y., 572-581.

Giustolisi, O., Laucelli, D., and Colombo, A. (2009). "Deterministic versus Stochastic Design of Water Distribution Networks." *Journal of Water Resources Planning and Management*, 135(2), 117-127.

Giustolisi, O., Savic, D., and Kapelan, Z. (2008). "Pressure-Driven Demand and Leakage Simulation for Water Distribution Networks." *Journal of Hydraulic Engineering*, 134(5), 626-635.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Goldberg, D. E. (2002). *The design of innovation: Lessons from and for competent genetic algorithms*, Kluwer Academic Publishers, Norwell, MA.

Gupta, R., and Bhave, P. (1996). "Comparison of methods for predicting deficient network performance." *Journal of Water Resources Planning and Management*, 122(3), 214-217.

Haario, H., Saksman, E., and Tamminen, J. (2001). "An adaptive Metropolis algorithm." *Bernoulli*, 7(2), 223-242.

Hadka, D., and Reed, P. (2012). "Diagnostic Assessment of Search Controls and Failure Modes in Many-Objective Evolutionary Optimization." *Evolutionary Computation*, 20(3), 423-452.

Hadka, D., and Reed, P. (2013). "Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework." *Evolutionary Computation*, 21(2), 231-259.

Halhal, D., Walters, G., Ouazar, D., and Savic, D. (1997). "Water Network Rehabilitation with Structured Messy Genetic Algorithm." *Journal of Water Resources Planning and Management*, 123(3), 137-146.

Hansen, M. P., and Jaszkiewicz, A. (1998). "Evaluating the quality of approximations to the non-dominated set." *IMM Technical Report IMM-REP-1998-7*, Technical University of Denmark.

Hassan, R., Cohanim, B., De Weck, O., and Venter, G. "A comparison of particle swarm optimization and the genetic algorithm." *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, U Michigan Press.

Hu, X., Eberhart, R. C., and Shi, Y. "Swarm intelligence for permutation optimization: a case study of n-queens problem." *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, 243-246.

Iorio, A. W., and Li, X. (2005). "Solving rotated multi-objective optimization problems using differential evolution." AI 2004: Advances in Artificial Intelligence, Springer, 861-872.

Jayaram, N., and Srinivasan, K. (2008). "Performance Based Optimal Design and Rehabilitation of Water Distribution Networks Using Life-Cycle Costing." *Water Resources Research*, W01417.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). "Efficient global optimization of expensive black-box functions." *Journal of Global Optimization* 13(4), 455-492.

Jourdan, L., Basseur, M., and Talbi, E.-G. (2009). "Hybridizing exact methods and metaheuristics: A taxonomy." *European Journal of Operational Research*, 199, 620-629.

Jourdan, L., Corne, D., Savic, D., and Walters, G. (2004). "Hybridising Rule Induction and Multi-Objective Evolutionary Search for Optimising Water Distribution Systems." Fourth International Conference on Hybrid Intelligent Systems, IEEE, 434-439.

Jourdan, L., Corne, D., Savic, D., and Walters, G. (2005a). "Preliminary Investigation of the 'Learnable Evolution Model' for Faster/Better Multiobjective Water Systems Design." Evolutionary Multi-Criterion Optimization III, C. Coello Coello, A. Hernández Aguirre, and E. Zitzler, eds., Springer-Verlag, Guanajuato, Mexico, 841-855.

Jourdan, L., Corne, D. W., Savic, D. A., and Walters, G. A. (2005b). "Lemmo: Hybridising Rule Induction And NSGA II For Multi-objective Water

Systems Design." Computers and Control in the Water Industry (CCWI 2005), Exeter, UK.

Kalungi, P., and Tanyimboh, T. T. (2003). "Redundancy model for water distribution systems." *Reliability Engineering & System Safety*, 82(3), 275-286.

Kapelan, Z., Savic, D. A., and Walters, G. A. (2005). "Multiobjective design of water distribution systems under uncertainty." *Water Resources Research* 41(W11407).

Keedwell, E. C., and Khu, S. T. (2003). "More choices in water system design through hybrid optimisation." Computing and Control for the Water Industry 2003, C. Maksimovic, D. Butler, and F. A. Memon, eds., Taylor & Francis, London, UK, 257-264.

Kennedy, J., and Eberhart, R. C. (2001). *Swarm intelligence*, Morgan Kaufmann Publishers Inc., San Francisco.

Kim, J. H., Kim, T. G., Kim, J. H., and Yoon, Y. N. (1994). "A study on the pipe network system design using non-linear programming." *Journal of Korean Water Resource Association*, 27(4), 59-67.

Knowles, J. (2005). "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems." *IEEE Transactions on Evolutionary Computation*, 10(1), 50-66.

Knowles, J., and Corne, D. "On metrics for comparing nondominated sets." *Proceedings of the 2002 Congress on Evolutionary Computation*, 711-716.

Kohonen, T. (1990). "The self organizing map." *Proceedings of the IEEE*, 78(9), 1464-1480.

Kollat, J. B., and Reed, P. M. (2005). "The value of online adaptive search: a performance comparison of NSGA-II, e-NSGAII, and e-MOEA." Lecture Notes in Computer Science, C. C. Coello, Aguirre, A. H., Zitzler, E., ed., Springer Verlag, Guanajuato, Mexico, 386-398.

Kollat, J. B., and Reed, P. M. (2006). "Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design." *Advances in Water Resources*, 29(6), 792-807.

Lansey, K. E., Duan, N., Mays, L. W., and Tung, Y.-K. (1989). "Water distribution system design under uncertainties." *Journal of Water Resources Planning and Management* 115(5), 630-645.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). "Combining convergence and diversity in evolutionary multiobjective optimization." *Evolutionary Computation*, 10(3), 263-282.

Lee, S. C., and Lee, S. I. (2001). "Genetic algorithms for optimal augmentation of water distribution networks." *Journal of Korean Water Resource Association*, 34(5), 567-575.

López-Ibáñez, M., Paquete, L., and Stutzle, T. (2010). "Exploratory Analysis of Stochastic Local Search Algorithms in Biobjective Optimization." *TR/IRIDIA/2009-015*, Springer Berlin Heidelberg.

Marchi, A., Salomons, E., Ostfeld, A., et al. (2013). "Battle of the Water Networks II." *Journal of Water Resources Planning and Management* 0(0), 04014009.

Michalski, R. S., Cervon, G., and Kaufman, K. A. (2000). "Speeding up evolution through learning: Lem." Intelligent Information Systems, 243-256.

Moral, R. J., and Dulikravich, G. S. (2008). "Multi-Objective Hybrid Evolutionary Optimization with Automatic Switching Among Constituent Algorithms." *AIAA Journal*, 46(3), 673-681.

Morley, M. S., and Tricarico, C. (2008). "Pressure Driven Demand Extension for EPANET (EPANETpdd)." Centre for Water Systems, University of Exeter, UK, Exeter.

Murphy, L. J., Dandy, G. C., and Simpson, A. R. (1994). "Optimum Design and Operation of Pumped Water Distribution Systems." Hydraulics Working with the Environment National Conference Publication, Institute of Engineers, Brisbane, Australia, 149-155.

Nicolini, M. (2004). "Evaluating Performance Of Multi-objective Genetic Algorithms For Water Distribution System Optimization." Sixth International Conference on Hydroinformatics, P. S-Y, L. K-K, and B. V, eds., World Scientic Publishing Company, Singapore, 850-857.

Norouzi, K., and Rakhshandehroo, G. R. (2011). "A Self Organizing Map Based Hybrid Multi-Objective Optimisation of Water Distribution Networks." *Transactions of Civil and Environmental Engineering*, 35(C1), 105-119.

Olsson, R. J., Kapelan, Z., and Savic, D. A. (2009). "Probabilistic building block identification for the optimal design and rehabilitation of water distribution systems." *Journal of Hydroinformatics*, 11(2), 89-105.

Padhye, N., Bhardawaj, P., and Deb, K. (2013a). "Improving differential evolution through a unified approach." *Journal of Global Optimization* 55(4), 771-799.

Padhye, N., Deb, K., and Mittal, P. (2013b). "An efficient and exclusively-feasible constrained handling strategy for evolutionary algorithms."

Palmes, P. P., Hayasaka, T., and Usui, S. (2005). "Mutation-based genetic neural network." *IEEE Transactions on Neural Networks*, 16(3), 587-600.

Papadimitriou, C. H., and Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity.*, Dover Publications, New York.

Parsopoulos, K. E., and Vrahatis, M. N. (2002). "Particle swarm optimization method in multiobjective problems." Proceedings of the 2002 ACM symposium on Applied computing, ACM, Madrid, Spain.

Perelman, L., Ostfeld, A., and Salomons, E. (2008). "Cross Entropy multiobjective optimization for water distribution systems design." *Water Resources Research*, 44(9), W09413.

Prasad, T. D., and Park, N.-S. (2004). "Multiobjective Genetic Algorithms for Design of Water Distribution Networks." *Journal of Water Resources Planning and Management*, 130(1), 73-82.

Prasad, T. D., and Tanyimboh, T., T. (2008). "Entropy Based Design of "Anytown" Water Distribution Network." Water Distribution Systems Analysis, J. E. Van Zyl, A. A. Ilemobade, and H. E. Jacobs, eds., ASCE, Kruger National Park, South Africa, 450-461.

Purshouse, R. C., and Fleming, P. J. (2007). "On the Evolutionary Optimization of Many Conflicting Objectives." *IEEE Transactions on Evolutionary Computation*, 11(6), 770-784.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc.

Raad, D. N., Sinske, A. N., and van Vuuren, J. H. (2009). "Robust multi-objective optimization for water distribution system design using a meta-metaheuristic." *International Transactions in Operational Research*, 16(5), 595-626.

Raad, D. N., Sinske, A. N., and van Vuuren, J. H. (2010a). "Multiobjective Optimization for Water Distribution System Design Using a Hyperheuristic." *Journal of Water Resources Planning and Management*, 136(5), 592-596.

Raad, D. N. (2011). "Multi-objective optimisation of water distribution systems design using metaheuristics," University of Stellenbosch, Stellenbosch.

Raad, D. N., Sinske, A. N., and van Vuuren, J. H. (2011). "Water distribution systems design optimisation using metaheuristics and hyperheuristics." *ORiON: The Journal of ORSSA*, 27(1), 17-44.

Raad, D. N., Sinske, A. N., and van Vuuren, J. H. (2008). "Jumping Genes for Water Distribution System Optimization." Water Distribution Systems Analysis 2008, J. E. Van Zyl, A. A. Ilemobade, and H. E. Jacobs, eds., ASCE, Kruger National Park, South Africa.

Raad, D. N., Sinske, A. N., and van Vuuren, J. H. (2010b). "Comparison of four reliability surrogate measures for water distribution systems design." *Water Resources Research*, 46(5), W05524.

Reca, J., and Martínez, J. (2006). "Genetic algorithms for the design of looped irrigation water distribution networks." *Water Resources Research*, 42(5), W05416.

Reed, P. M., Hadka, D., Herman, J. D., Kasprzyk, J. R., and Kollat, J. B. (2013). "Evolutionary multiobjective optimization in water resources: The past, present, and future." *Advances in Water Resources*, 51(0), 438-456.

Rossman, L. (2007). "Discussion of "Solution for Water Distribution Systems under Pressure-Deficient Conditions" by Wah Khim Ang and Paul W. Jowitt." *Journal of Water Resources Planning and Management* 133(6), 566-567.

Rossman, L. A. (2000). "EPANET 2 USERS MANUAL." U.S. Environment Protection Agency, Cincinnati, Ohio, USA.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989). "Design and analysis of computer experiments (with discussion)." *Statistical Science*, 4(4), 409-423.

Schaake, J., and Lai, D. (1969). "Linear programming and dynamic programming application to water distribution network design." *116*, Dept. of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Schott, J. R. (1995). "Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization," Master's Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Shamir, U. (1974). "Optimal Design and Operation of Water Distribution Systems." *Water Resources Research* 10(1), 27-36.

Shannon, C. (1948). "A Mathematical Theory of Communication." *Bell System Technical Journal*.

Sherali, H., Subramanian, S., and Loganathan, G. V. (2001). "Effective Relaxations and Partitioning Schemes for Solving Water Distribution Network Design Problems to Global Optimality." *Journal of Global Optimization*, 19(1), 1-26.

Siew, C., and Tanyimboh, T. (2012). "Pressure-Dependent EPANET Extension." *Water Resources Management*, 26(6), 1477-1498.

Simpson, A., Dandy, G., and Murphy, L. (1994). "Genetic Algorithms Compared to Other Techniques for Pipe Optimization." *Journal of Water Resources Planning and Management*, 120(4), 423-443.

Simpson, A. R., and Goldberg, D. E. (1994). "Pipeline Optimisation via Genetic Algorithms: From Theory to Practice." Water Pipeline Systems, Mechanical Engineering Publications, London, 309-320.

Singh, H. K., Ray, T., and Sarker, R. (2013). "Optimum oil production planning using infeasibility driven evolutionary algorithm." *Evolutionary Computation* 21(1), 65-82.

Spears, W. (1993). "Crossover or mutation." Proceedings of Foundations of Genetic Algorithms, Morgan Kaufmann, 221-237.

Srinivas, N., and Deb, K. (1994). "Multiobjective optimization using nondominated sorting in genetic algorithms." *Evolutionary Computation*, 2(3), 221-248.

Srivastava, R. P. (2002). "Time continuation in genetic algorithms." Illinois Genetic Algorithm Laboratory.

Storn, R., and Price, K. (1997). "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces." *Journal of Global Optimization*, 11, 341-359.

Talbi, E. G. (2002). "A Taxonomy of Hybrid Metaheuristics." *Journal of Heuristics*, 8, 541-564.

Tanyimboh, T. T., and Templeman, A. B. (1993). "Calculating maximum entropy flows in networks." *Journal of the Operational Research Society*, 383-396.

Tanyimboh, T. T., and Templeman, A. B. (2010). "Seamless pressure-deficient water distribution system model." *J Water Manag*, 163(8), 389-396.

Todini, E. (2000). "Looped water distribution networks design using a resilience index based heuristic approach." *Urban Water*, 2(2), 115-122.

Tolson, B. A., Maier, H. R., Simpson, A. R., and Lence, B. J. (2004). "Genetic algorithms for reliability-based optimization of water distribution systems." *Journal of Water Resources Planning and Management* 130(1), 63-72.

Vamvakeridou-Lyroudia, L., Walters, G., and Savic, D. (2005). "Fuzzy Multiobjective Optimization of Water Distribution Networks." *Journal of Water Resources Planning and Management* 131(6), 467-476.

Vamvakeridou-Lyroudia, L. S., Savic, D. A., and Walters, G. A. (2006). "Fuzzy hierarchical decision support system for water distribution network optimization." *Civil Engineering and Environmental Systems*, 23(3), 237-261.

Veldhuizen, D. A. V. (1999). "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," PhD Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Vose, M. D. (1994). "A closer look at mutation in genetic algorithms." *Annals of Mathematics and Artificial Intelligence*, 10(4), 423-434.

Vrugt, J. A., and Robinson, B. A. (2007). "Improved Evolutionary Optimization from Genetically Adaptive Multimethod Search." *Proceedings of the National Academy of Sciences*, 104(3), 708-711.

Walski, T. M. (2001). "The Wrong Paradigm, Why Water Distribution Optimization Doesn't Work." *Journal of Water Resources Planning and Management*, 127(4), 203-205.

Walski, T. M., Brill, J. E. D., Gessler, J., Goulter, I. C., Jeppson, R. M., Lansey, K., Lee, H.-L., Liebman, J. C., Mays, L., Morgan, D. R., and Ormsbee, L. (1987). "Battle of the Network Models: Epilogue." *Journal of Water Resources Planning and Management*, 113(2), 191-203.

Walters, G. A., Halhal, D., Savic, D., and Ouazar, D. (1999). "Improved design of "Anytown" distribution network using structured messy genetic algorithms." *Urban Water*, 1(1), 23-38.

Wang, Q., Guidolin, M., Savić, D., and Kapelan, Z. (2014a). "Two-Objective Design of Benchmark Problems of Water Distribution System via MOEAs: Towards the Best-Known Approximation to the True Pareto Front." *Journal of Water Resources Planning and Management*, accepted.

Wang, Q., Savić, D. A., and Kapelan, Z. (2014b). "Hybrid Metaheuristics for Multi-Objective Design of Water Distribution Systems." *Journal of Hydroinformatics*, 16(1), 165-177.

While, L., Bradstreet, L., and Barone, L. (2011). "A Fast Way of Calculating Exact Hypervolumes." *IEEE Transactions on Evolutionary Computation*, PP(99).

Wolpert, D. H., and Macready, W. G. (1997). "No free lunch theorems for optimization." *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.

Wu, Z., Wang, R., Walski, T., Yang, S., Bowdler, D., and Baggett, C. (2009). "Extended Global-Gradient Algorithm for Pressure-Dependent Water Distribution Analysis." *Journal of Water Resources Planning and Management* 135(1), 13-22.

Xu, C., and Goulter, I. C. (1999). "Reliability-based optimal design of water distribution networks." *Journal of Water Resources Planning and Management* 125(6), 352-362.

Yang, S., Li, M., Liu, X., and Zheng, J. (2013). "A grid-based evolutionary algorithm for many-objective optimization." *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 17(5), 721-736.

Zheng, F., Simpson, A., and Zecchin, A. (2010). "A Method for Assessing the Performance of Genetic Algorithm Optimization for Water Distribution Design." Water Distribution Systems Analysis 2010, 771-785.

Zheng, F., Simpson, A. R., Zecchin, A. C., and Deuerlein, J. W. (2013a). "A graph decomposition based approach for water distribution network optimization." *Water Resources Research* 49(4), 2093-2109.

Zheng, F., Zecchin, A., and Simpson, A. (2013b). "Self-Adaptive Differential Evolution Algorithm Applied to Water Distribution System Optimization." *Journal of Computing in Civil Engineering*, 27(2), 148-158.

Zheng, F., Zecchin, A. C., Simpson, A. R., and Lambert, M. F. (2013c). "Non-crossover dither creeping mutation genetic algorithm for pipe network optimization." *Journal of Water Resources Planning and Management* 140(4).

Zitzler, E. (1999). "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," Swiss Federal Institute of Technology Zurich, Zurich.

Zitzler, E., Deb, K., and Thiele, L. (2000). "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." *Evolutionary Computation*, 8(2), 173-195.

Zitzler, E., Laumanns, M., and Thiele, L. (2002). "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization." Evolutionary Methods for Design, Optimisation and Control, K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, eds., International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, pp. 95-100.

Zitzler, E., and Thiele, L. (1999). "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 3(4), 257-271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and da Fonseca, V. G. (2003). "Performance assessment of multiobjective optimizers: an analysis and review." *IEEE Transactions on Evolutionary Computation*, 7(2), 117-132.