Internal Report 2010–04 June 2010

Universiteit Leiden
Opleiding Informatica

Multiobjective Robust Optimization of Water Distribution Networks

Edgar Reehuis

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Due to its practical importance and inherent complexity, the optimization of distribution networks for supplying drinking water has been the subject of extensive study for the past 30 years. The optimization is governed by sizing the pipes in the water distribution network (WDN). The pipe diameters are to be selected from a list of available commercial diameters, making WDN optimization a pure discrete optimization problem. The size of the search space is determined by the number of available diameters, raised to the power of the number of pipes (e.g., 6^{34}, 16^{21}). Originally being a search for the least cost solution, subject to delivering pre-defined water demands to the consumer nodes at a minimum required pressure, the optimization model has gradually changed to a multiobjective scheme including objectives for reliability under mechanical failure (e.g., pipe breakage) and hydraulic failure (e.g., increased demands). Also, robust optimization schemes have been suggested, which aim for hydraulically reliable optima by accounting for fluctuating demands at the consumer nodes. We intend to determine the added value of the multiobjective robust approach (*robust*), over the basic multiobjective model (*raw*). Raw and robust results, obtained using the same number of evaluations, are both re-evaluated to approximate their robustness using 2000 randomly generated demand samples per solution. Surprisingly, the results indicate that in terms of robustness, the raw Multiobjective Evolutionary Algorithms used (NSGA-II, SMS-EMOA) outperform their variants that explicitly include extensions for robust evaluation; the standard extensions for robust optimization, namely the Single Evaluation Model (SEM) and Multi Evaluation Model (MEM), were tested.

Acknowledgements

The author would like to thank Johannes Kruisselbrink for the valuable discussions, useful comments, and good suggestions, and Michael Emmerich and André Deutz for their assistance in improving this thesis.

Furthermore, I would like to thank my uncle and aunt Leo and Riete, and especially my mother for their support over the past years.

Contents

Abstract i

Acknowledgements ii

List of Figures vii

List of Tables ix

List of Algorithms xi

Abbreviations xii

Nomenclature xv

Preface xxi

1 Design Optimization of Water Distribution Networks 1

1.1 Problem Definition . . . 2

1.2 Hydraulic Simulator . . . 3

1.2.1 Constants . . . 5

1.2.2 Simulation Variables . . . 6

1.2.3 Decision Variables . . . 7

1.2.4 Hydraulic Equation System . . . 7

1.2.5 Friction Head Loss Formula . . . 9

1.2.6 Units . . . 10

1.3 Objective Functions from Related Work . . . 11

1.3.1 Functions of Cost . . . 12

1.3.1.1 Investment Cost . . . 12

1.3.2 Functions of Reliability . . . 13

1.3.2.1 Maximum Individual Head Deficiency . . . 13

1.3.2.2 Sum of Maximum Individual Head Deficiency . . . 14

1.3.2.3 Total Head Deficit . . . 14

1.3.2.4 Number of Nodes with Head Deficiency . . . 14

1.3.2.5 Minimum Surplus Head . . . 14

1.3.2.6 Total Surplus Head . . . 15

1.3.2.7 Weighted Demand Supply Ratio . . . 15

1.3.2.8 Resilience Index . . . 16

1.3.2.9 Network Resilience . . . 17

1.3.2.10 System Entropy . . . 18

1.4 Multiobjective Optimization Approach . . . 19
1.4.1 Deterministic Approach . . . 20
1.4.2 Stochastic Approach . . . 22
1.4.3 Multiobjective Evolutionary Algorithms . . . 23

2 Multiobjective Robust Optimization 25
2.1 Multiobjective Optimization . . . 25
2.1.1 Hypervolume Measure . . . 27
2.1.1.1 Reference Point for WDN Optimization . . . 28
2.2 Robust Optimization . . . 30
2.2.1 Robust Evaluation Methods . . . 32

3 Multiobjective Evolutionary Algorithms 35
3.1 NSGA-II . . . 35
3.1.1 Variation Operators for WDN Optimization . . . 39
3.1.2 Overview of Parameters . . . 40
3.2 SMS-EMOA . . . 40
3.2.1 SMS-EMOA-dp . . . 43
3.2.2 Choice of Reference Point . . . 44
3.2.3 Overview of Parameters . . . 45
3.3 Mixed-Integer Evolution Strategy . . . 45
3.3.1 Representation . . . 46
3.3.2 Initialization . . . 46
3.3.3 Recombination . . . 47
3.3.4 Mutation . . . 47
3.3.5 Algorithm Outline . . . 49
3.3.6 NSGA-II-sa . . . 50
3.3.6.1 Overview of Parameters . . . 52
3.3.7 SMS-EMOA-sa . . . 52
3.3.7.1 Overview of Parameters . . . 54

4 Experiments 55
4.1 Two Loop . . . 55
4.2 Hanoi . . . 57
4.3 New York City . . . 59
4.4 Other Test Problems . . . 61
4.5 Experimental Setup . . . 62
4.6 Raw Results . . . 66
4.7 Robust Optimization Results . . . 70

5 Conclusion 75

A Raw Attainment Surfaces 79

B Robust Attainment Surfaces 105

Bibliography **131**

List of Figures

1.1 The Multiobjective WDN Optimization Loop 4
1.2 Example Network Layout 6
1.3 Functional Relationship Q_j^{met} 16

2.1 Three-dimensional Hypervolume Example 28
2.2 Choice of Reference Point 29

3.1 Non-dominated Sorting 36
3.2 Crowding-distance Calculation 37
3.3 NSGA-II Explained 39
3.4 Selection Measures Compared 41
3.5 Dominating Points Measure 44

4.1 Two Loop Network Layout 56
4.2 Hanoi Network Layout 57
4.3 New York City Network Layout 60
4.4 Attainment Curve Explained 64
4.5 Attainment Surface Example 64
4.6 Raw Results Overview 66
4.7 Median Raw Fronts Compared 69
4.8 Robust Results Overview 70
4.9 Supposedly Best Fronts Analyzed 72

A.1 Raw – Two Loop – NSGA-II 80
A.2 Raw – Two Loop – NSGA-II-sa 81
A.3 Raw – Two Loop – NSGA-II-sa-sp 82
A.4 Raw – Two Loop – NSGA-II-sa-sp-λ 83
A.5 Raw – Two Loop – SMS-EMOA 84
A.6 Raw – Two Loop – SMS-EMOA-dp 85
A.7 Raw – Two Loop – SMS-EMOA-sa 86
A.8 Raw – Two Loop – SMS-EMOA-sa-dp 87
A.9 Raw – Hanoi – NSGA-II 88
A.10 Raw – Hanoi – NSGA-II-sa 89
A.11 Raw – Hanoi – NSGA-II-sa-sp 90
A.12 Raw – Hanoi – NSGA-II-sa-sp-λ 91
A.13 Raw – Hanoi – SMS-EMOA 92
A.14 Raw – Hanoi – SMS-EMOA-dp 93
A.15 Raw – Hanoi – SMS-EMOA-sa 94
A.16 Raw – Hanoi – SMS-EMOA-sa-dp 95

A.17 Raw – New York City – NSGA-II . . . 96
A.18 Raw – New York City – NSGA-II-sa . . . 97
A.19 Raw – New York City – NSGA-II-sa-sp . . . 98
A.20 Raw – New York City – NSGA-II-sa-sp-λ . . . 99
A.21 Raw – New York City – SMS-EMOA . . . 100
A.22 Raw – New York City – SMS-EMOA-dp . . . 101
A.23 Raw – New York City – SMS-EMOA-sa . . . 102
A.24 Raw – New York City – SMS-EMOA-sa-dp . . . 103

B.1 Robust – Two Loop – NSGA-II . . . 106
B.2 Robust – Two Loop – NSGA-II-sa-sp-lambda . . . 107
B.3 Robust – Two Loop – NSGA-II-sa-sp-λ-SEM . . . 108
B.4 Robust – Two Loop – NSGA-II-sa-sp-λ-MEM . . . 109
B.5 Robust – Two Loop – SMS-EMOA . . . 110
B.6 Robust – Two Loop – SMS-EMOA-sa . . . 111
B.7 Robust – Two Loop – SMS-EMOA-sa-SEM . . . 112
B.8 Robust – Two Loop – SMS-EMOA-sa-MEM . . . 113
B.9 Robust – Hanoi – NSGA-II . . . 114
B.10 Robust – Hanoi – NSGA-II-sa-sp-lambda . . . 115
B.11 Robust – Hanoi – NSGA-II-sa-sp-λ-SEM . . . 116
B.12 Robust – Hanoi – NSGA-II-sa-sp-λ-MEM . . . 117
B.13 Robust – Hanoi – SMS-EMOA . . . 118
B.14 Robust – Hanoi – SMS-EMOA-sa . . . 119
B.15 Robust – Hanoi – SMS-EMOA-sa-SEM . . . 120
B.16 Robust – Hanoi – SMS-EMOA-sa-MEM . . . 121
B.17 Robust – New York City – NSGA-II . . . 122
B.18 Robust – New York City – NSGA-II-sa-sp-lambda . . . 123
B.19 Robust – New York City – NSGA-II-sa-sp-λ-SEM . . . 124
B.20 Robust – New York City – NSGA-II-sa-sp-λ-MEM . . . 125
B.21 Robust – New York City – SMS-EMOA . . . 126
B.22 Robust – New York City – SMS-EMOA-sa . . . 127
B.23 Robust – New York City – SMS-EMOA-sa-SEM . . . 128
B.24 Robust – New York City – SMS-EMOA-sa-MEM . . . 129

List of Tables

4.1 Two Loop Node Data . . . 56
4.2 Two Loop Pipe Data . . . 56
4.3 Two Loop Diameters . . . 57
4.4 Hanoi Node Data . . . 58
4.5 Hanoi Pipe Data . . . 58
4.6 Hanoi Diameters . . . 59
4.7 New York City Diameters . . . 60
4.8 New York City Node Data . . . 61
4.9 New York City Pipe Data . . . 61
4.10 Raw Hypervolume Results . . . 67
4.11 Running Time of Raw Experiments . . . 68
4.12 Robust Hypervolume Results . . . 71
4.13 Running Time of Robust Optimization Experiments . . . 73

List of Algorithms

1 NSGA-II 38

2 SMS-EMOA 42

3 `reduce(`Q`)` 43

4 `reduce-dp(`Q`)` 43

5 `mutate(`$Q_t, \ldots, n_d, \ldots, n_p$`)` 48

6 $T^r_{[a,b]}(x)$ 49

7 Mixed-Integer Evolution Strategy 50

8 `select-NSGA(`P_t`,` Q_t`)` 51

9 `select-SMS(`P_t`,` Q_t`)` 53

Abbreviations

CWS	**C**enter for **W**ater **S**ystems, University of Exeter
EA	**E**volutionary **A**lgorithm
ES	**E**volution **S**trategy
MEM	**M**ulti **E**valuation **M**odel
MIES	**M**ixed-**I**nteger **E**volution **S**trategy
MO	**M**ulti**o**bjective
MOEA	**M**ulti**o**bjective **EA**
NSGA-II	**N**ondominated **S**orting **G**enetic **A**lgorithm **II**
NSGA-II-sa	**NSGA-II** with **s**elf-**a**daptation
NSGA-II-sa-sp	**NSGA-II-sa** with **s**ingle **p**arent offspring
NSGA-II-sa-sp-$\boldsymbol{\lambda}$	**NSGA-II-sa-sp** with $\boldsymbol{\lambda}$ number of offspring
SMS-EMOA	$\boldsymbol{\mathcal{S}}$ **M**etric **S**election **E**volutionary **M**ultiobjective **O**ptimization **A**lgorithm
SMS-EMOA-dp	**SMS-EMOA** with **d**ominating **p**oints selection
SMS-EMOA-sa	**SMS-EMOA** with **s**elf-**a**daptation
SMS-EMOA-sa-dp	**SMS-EMOA-sa** with **d**ominating **p**oints selection
SEM	**S**ingle **E**valuation **M**odel
SO	**S**ingle **o**bjective
WDN	**W**ater **d**istribution **n**etwork

Nomenclature

$\vec{a}$	individuals in MIES, i.e., tuples containing decision variables and strategy parameters
$\mathbf{a}$	vector of environmental parameters, i.e., assuming a deterministic, fixed, environment
b_i	begin node of pipe i
C_i	roughness coefficient of pipe i
C_v	coefficient of variation, i.e., standard deviation divided by mean
$d(s, Q)$	number of points in Q that dominate individual s
$\vec{d}$	discrete nominal decision variables
D_i	diameter code of pipe i
e_i	end node of pipe i
f_i	i-th objective function considered during optimization
f_j	flow continuity around node j
$\mathbf{f}_{\mathit{eff}}$	effective fitness, i.e., the robust objective function values taken as the expected fitness under fluctuating environmental parameters $\boldsymbol{\alpha}$
$\mathbf{f}_{est}$	estimated fitness, i.e., estimation of the effective fitness, used in robust optimization
$\mathbf{f}_{raw}$	raw fitness, i.e., the objective function values as calculated by the original objective functions
$\mathcal{F}_i$	i-th front of solutions, based on non-domination rank
F_{IC}	objective function Investment Cost
F_{mIHD}	objective function Maximum Individual Head Deficiency
F_{mSH}	objective function Minimum Surplus Head
F_{RI}	objective function Resilience Index
F_{SE}	objective function System Entropy

F_{mIHD}	objective function Sum of Maximum Individual Head Deficiency
F_{NHD}	objective function Number of Nodes with Head Deficiency
F_{NR}	objective function Network Resilience
F_{tHD}	objective function Total Head Deficit
F_{tSH}	objective function Total Surplus Head
F_{wDSR}	objective function Weighted Demand Supply Ratio
h_j	hydraulic head at node j
$h_{j,k}$	hydraulic head at node j under demand pattern k
K_i	resistance coefficient of pipe i
l_{d_i}	lower bound of the domain of decision variable d_i
L_i	length of pipe i
m	total number of pipes in the network
m^{exist}	number of *existing* pipes in the network
m^{new}	number of *new* pipes in the network
M	Pareto-optimal set of solutions $\mathbf{x}$, i.e., vectors of decision variables
n	number of internal nodes in the network
n_d	number of discrete nominal decision variables
n_f	number of objective functions considered during optimization
n_p	number of mutation probabilities for discrete nominal decision variables
$N(0,1)$	random number sampled from a normal distribution with mean 0 and standard deviation 1
N_c	constant random number
N^{dp}	number of demand patterns
N^{sample}	robust sample set size
$\vec{p}$	mutation probabilities for discrete nominal decision variables
p_j	pressure at node j
p^{req}	pressure required to satisfy the nodal water demand completely
p^{zero}	pressure corresponding to zero nodal water demand satisfaction
P^{del}	total power delivered throughout network in order to service nodal demands
$P^{del,req}$	total power corresponding to network-wide nodal demand delivery at p^{req}
P^{fric}	total power lost through friction in the network
$P^{fric,req}$	power lost through friction in case of network-wide demand delivery at

	p^{req}
P^{in}	total power put into the network
P_t	parent population at generation t
q_i	water flow in pipe i
q_t	offspring individual at generation t
Q_j	water demand drawn from node j
Q_j^{met}	portion of the water demand met at node j
Q_t	offspring population at generation t
r	number of reservoir nodes in the network
R	robustness of a design, i.e., a solution to a WDN test problem
R_t	combined parent and offspring pool at generation t
s_j	entropy of node j, i.e., extent to which the incoming flows to node j are similar in magnitude
S	entropy of the network, i.e., sum over the nodal entropies s_j, weighted by the magnitude of the total incoming flow per node
$\mathbb{S}$	search space of possible solutions to an optimization problem
$\mathcal{S}(A)$	hypervolume measure or $\mathcal{S}$ metric of a set $A \subseteq \mathbb{S}$
$\mathcal{S}_{eff}$	$\mathcal{S}$ metric based on effective fitness $\mathbf{f}_{eff}$
$\mathcal{S}_{est}$	$\mathcal{S}$ metric based on estimated fitness $\mathbf{f}_{est}$
$\mathcal{S}_{raw}$	$\mathcal{S}$ metric based on raw fitness $\mathbf{f}_{raw}$
$\widetilde{\mathcal{S}}(s, r, P_t)$	expected hypervolume of s, i.e., hypervolume of P_t and s, excluding r
$T^r_{[a,b]}(x)$	transformation function that reflects a real-valued x back into domain $[a, b]$
u_{d_i}	upper bound of the domain of decision variable d_i
$U(a, b)$	random number sampled from a continuous uniform distribution with lower bound a and upper bound b
x_i	i-th decision variable of the optimization problem
X	weighted surplus power of the network, i.e., summation over the nodal weighted surplus powers ${X^{nod}}_j$
X^{max}	maximum surplus power, i.e., weighted surplus power X under the assumption of no power lost through friction
${X^{nod}}_j$	nodal weighted surplus power at node j, i.e., nodal uniformity times surplus power

y_i	objective function value assigned by objective function f_i to a solution $\mathbf{x}$, i.e., vector of decision variables
$\mathbf{y}^{BoundTP}$	boundary point of a test problem, i.e., containing the worst value possible per objective
$\mathbf{y}^{IdealTP}$	boundary point of a test problem, i.e., containing the best value possible per objective
$\mathbf{y}^{ref}$	reference point lying in the objective space, used in calculation of the $\mathcal{S}$ metric
Z_j	elevation of node j

α	exponent of the friction head loss formula
$\boldsymbol{\alpha}_i$	vector of noisy environmental parameters, i.e., the i-th sample
β	exponent of the Hazen-Williams head loss equation
γ	specific weight of water
$\Delta_{\mathcal{S}}(s, \mathcal{F}_\nu)$	hypervolume contribution of individual s in the last front $\mathcal{F}_\nu$
ε	arbitrary small positive value
λ	offspring population size
$\varLambda$	Lebesgue measure
μ	parent population size
ν	number of fronts $\mathcal{F}_i$ in which population P_t is partitioned
ρ	number of parent individuals involved in recombination
τ	global learning rate
τ'	local learning rate
ψ_i	flow loss through leakage in pipe i
ω	Hazen-Williams numerical conversion constant

$convD$	conversion factor from defined unit for $size(D_i)$ to ft
$convQ$	conversion factor from defined unit for q_i to cfs
$defic_j$	head deficiency at node j, i.e., pressure short of the required level p^{req}
$dest(i)$	destination of the water flow in pipe i
in_j	water flow transported towards node j

IN	sum of all incoming flows in the network
$loss_i$	head loss occurring in pipe i
$orig(i)$	origin of the water flow in pipe i
pdf($\mathbf{a}$)	distribution of the variation in the environmental parameters $\mathbf{a}$
$price(D_i)$	unit price of new pipe $i \in \{m^{exist} + 1, \ldots, m^{exist} + m^{new}\}$
$size(D_i)$	diameter size of pipe i
$supplied_j$	total demand supplied by reservoir node j
$surplus_j$	excess pressure available at node j above required level p^{req}
$unif_j$	uniformity of node j, i.e., extent to which the pipes connected to node j are similar in diameter

Preface

The traditional approaches of designing *water distribution networks* (WDNs) by trial and error and using “rules of thumb” are both time-consuming and likely to produce suboptimal solutions. Motivated by reducing the high investment and maintenance costs inherent to these systems, optimization of the design of WDNs has been the subject of study by scientists in different fields for more than thirty years. It must be recognized however that automated optimization can still only assist the engineer in design and that engineering judgment and experience remains required to provide practicable solutions [88].

In WDN optimization the water distribution backbone of a city is modeled using *pipes*, *nodes*, and *reservoirs*. In this model the pipes take up a central position as it is the selection of *diameter sizes* for pipe segments that governs the optimization process. Smaller diameters are less expensive to procure, while larger diameters result in higher water pressure at the demand nodes in the network.

The difficulty of optimizing water distribution systems is mainly due to the discrete nature of the *decision variables* and the size of the search space, which can be calculated as the number of possible discrete pipe diameters (the available commercial diameters) to the power of the number of pipes in the network [91]. Optimal design of pipe networks belongs to the class of large *NP-hard* problems [97] that are difficult to solve using classical operational research techniques [75].

Many formal optimization techniques have been proposed for the optimal design of WDNs. Most of these techniques have employed traditional linear and nonlinear programming approaches, where the decision variables were assumed to be continuous [88]. Comprehensive reviews of these techniques are available, e.g., in [49]. From the mid-nineties of the past century on, there has been a growing interest in the application of

Evolutionary Algorithms (EAs) to the design of water distribution systems, since they can directly handle the discrete nature of the problem [69, 76].

In the original problem formulation the design optimization was formulated as a *single objective* (SO) problem with *investment cost* as *objective* and a *constraint* on the minimal *pressure* required to be available at the consumer nodes. The constrained SO approach has the tendency however of eliminating all capacity beyond the minimum level, as this has no value in the model [95]. Surplus capacity at a reasonable cost is actually of high value, because of the great deal of uncertainty in predicting future consumer demands. Criticism on the constrained single objective model by, for instance, Walski [95] is acknowledged by many authors (e.g., [28, 35, 38, 71]), his main points being [35]:

- it is difficult for practitioners to define objective functions and constraints;
- there should not be a single set of demands for which the system is designed;
- optimization fails to account for the fact that a total distribution system is not built all at once;
- optimization tends to reduce costs by reducing the diameter of or completely eliminating pipes, thus leaving the system with insufficient capacity to respond to pipe breaks or demands that exceed design values without failing to achieve required performance levels.

De Neufville et al. [23] presented one of the first studies that recognized that WDN optimization is an inherently *multiobjective* problem. Formiga et al. [43] address some of the issues listed above by approaching WDN optimization as an *unconstrained multiobjective* (MO) problem, using three objectives: *Investment Cost*, *Demand Supply Ratio*, and *System Entropy*. Investment Cost was inherited from the original formulation, while Demand Supply Ratio was included as an objective to maximize the mean nodal pressure. The third objective, System Entropy, was included to increase the reliability of the network by appreciating redundancy in the system.

Following a multiobjective approach has the potential of providing interesting solutions that are ignored when only Investment Cost is taken into account: imagine two solutions, one being slightly more expensive than the other but scoring better on both reliability and mean nodal pressure. One the other hand, cost may be the prime consideration.

The least network cost for satisfying the minimal pressure constraint in the SO model may, for instance, exceed the available budget in developing countries [4]. Using the MO model, suboptimal but cost-efficient solutions can be considered. Multiobjective EAs (MOEAs), due to their general success in this arena, are becoming increasingly favored for WDN optimization [52].

As water distribution networks are basically intended to last indefinitely, subject to future additions and extensions, another approach has been that of maximizing the *robustness* of the WDN design under changing future requirements and component wear. Pipe roughness increases over time as encrustation occurs, and nodal consumer demands are difficult to predict since they vary over short time periods and because they are modeled as lumped at single locations rather than distributed along a pipe [62]. The WDN design can be prepared for this changing environment by optimizing over fluctuating environmental settings such as pipe roughness and consumer demands.

This work aims at determining the added value of the Multiobjective Robust approach (*robust*), over the basic Multiobjective approach (*raw*), using a fixed budget of evaluations by the *hydraulic simulator*. Solutions by the raw and robust approaches can both be tested for actual robustness of design, providing a means of comparing both models on attained robustness levels.

This document is structured as follows:

Chapter 1 lays out the WDN optimization problem;

Chapter 2 introduces and formalizes multiobjective and robust optimization;

Chapter 3 provides multiobjective optimization methods;

Chapter 4 presents the experimental setup and the results obtained;

Chapter 5 concludes and indicates future research directions.

Chapter 1

Design Optimization of Water Distribution Networks

Water distribution networks (WDNs) are complex and require huge investments in their construction and maintenance. For these reasons, a need to improve their efficiency by way of minimizing their cost and maximizing the benefit accrued from them is strongly felt [71]. In the past, design of water distribution networks was based on experience. However, in the last three decades, a significant number of computerized methods have been applied to WDN optmization, ranging from *linear programming*, *dynamic programming*, and *enumeration* techniques in the early years, to more recently *Evolutionary Algorithms* [69, 76], *Tabu Search* [65, 87], *Simulated Annealing* [21, 86], *Ant Colony Optimization* [66], *Harmony Search* [45], *Particle Swarm Optimization* [68], *Shuffled Complex Evolution* [64], and *Dynamically Dimensioned Search* [84].

WDN optimization is driven by the selection of appropriately sized network components. Originally approached as a *single objective constrained* problem considering minimization of cost under required pressure constraint, criticism with respect to the resulting impractical and unreliable designs [95] led to *multiobjective* reformulations of the problem coming into favor, like those by Formiga et al. [43] and Prasad and Park [71]. Complementary, we have seen stochastic approaches emerge that strive for reliability by varying loads laid on the network [5].

Section 1.1 introduces the overall process of WDN optimization, which breaks up in three modules: the *hydraulic simulator*, the *objective functions*, and the *multiobjective optimizer*. These are then addressed in depth in Section 1.2, 1.3, and 2.1 respectively.

1.1 Problem Definition

Provided with a *network definition* consisting of a fixed *network layout* (i.e., the arrangement of the *nodes* and *pipes*) and accompanying *nodal demands*, we define the *WDN design optimization problem* as selecting the "optimal" diameter sizes for the pipes in the network, taken from a table of available *commercial diameters*. Nodal demand is the flow of water required by one or many network users (e.g., residential, industrial, commercial) at a network node [38], which is to be supplied at an adequate *water pressure* [88]. WDN designs should also be prepared to carry incidental large peak flows required for firefighting [38].

The optimal diameters are selected with regard to minimizing *cost* and simultaneously maximizing *reliability*. Reliability is the extent to which the design is able to service demands under *mechanical* and/or *hydraulic failures* [71]. Mechanical failures are malfunctions of network components such as pumps, valves, and pipe breakage; hydraulic failures are changes in demand or pressure, increased friction due to aging of the pipes etc. [67]. In principle the network layout itself could be subjected to optimization as well, but due to the extreme complexity involved and since largely being restricted by the location of roads, this is mostly left out of consideration in pipe network optimization [71].

From a reliability point of view, the largest diameter available is ideally used throughout the network as this will cause the least *friction loss* in the pipes and provide the maximum number of good alternative routes should mechanical failures occur. In our model each pipe gets assigned a roughness coefficient determined by the construction material used per selected diameter, indicating the resistance working on the water being transported. Concerning pipes that are made up out of the same material, larger diameters exercise less friction than smaller ones. However, selecting maximum-sized diameters network-wide will also be the most expensive solution available. Thus in this work multiobjective optimization is used, with one *objective* function expressing cost and two objectives introducing reliability, leading to a set of mutually *non-dominated* solutions from which the civil engineer can select *good compromise* solutions based on preferences regarding the objectives involved.

In literature a distinction is made between two types of WDN design problems: design optimization for a new network, to be built from scratch, and *rehabilitation* of an existing network [33], in the form of adding new and restorating existing pipes. This work examines both types of design problems, but with respect to rehabilitation only *parallel expansion* is considered, i.e., the insertion of new pipes in the network next to operational links.

WDN optimization as applied here is an iterative process, which can be abstracted as a multiobjective optimization loop consisting of three main components (cf. Fig. 1.1):

1. *Hydraulic simulator* (cf. Section 1.2): an optimizer that solves the non-linear hydraulic equation system by minimizing residuals (cf. Section 1.2.4), taking as input the network definition (i.e., network layout and nodal demands) and one of the solutions generated by the *multiobjective optimizer* at a time (i.e., a list of selected diameter per pipe), while providing *flows* and *pressures* as output (to be used as input by the *objective functions*);

2. *Objective functions* (cf. Section 1.3): calculate the fitness of a solution expressed in three objectives, based on the network definition, the flows and pressures received from the hydraulic simulator (depending on the current diameter selection), and the current diameter selection determined by the solution that is being assessed;

3. *Multiobjective optimizer* (cf. Section 2.1): generates new solutions, which are vectors of diameter codes (the codes point to certain size, price, and roughness triples in the table of commercial diameters), and selects the best solutions to proceed to the next *generation* based on their *objective function values.*

In the following sections we go into depth of the three components that make up the optimization loop.

1.2 Hydraulic Simulator

Within the optimization loop the hydraulic simulator models the WDN design it receives as input (i.e., network definition and selected pipe diameters) consisting of various network components, in the benchmark test problems studied in this work (cf. Chapter 4)

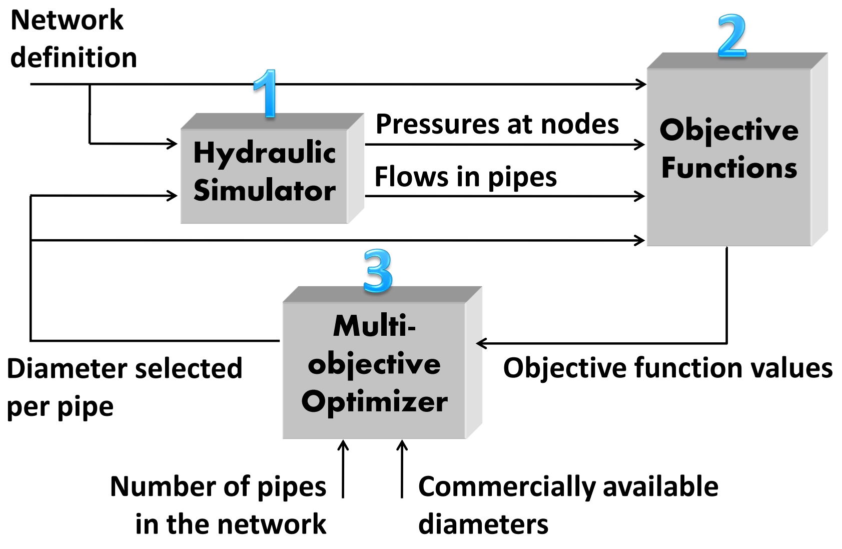

Network definition
1
2
Hydraulic Simulator
Pressures at nodes
Flows in pipes
Objective Functions
3
Multi-objective Optimizer
Diameter selected per pipe
Objective function values
Number of pipes in the network
Commercially available diameters

Figure 1.1: **The Multiobjective WDN Optimization Loop**

limited to *nodes*, *reservoirs*, and *pipes*. Through solving the non-linear hydraulic equation system (cf. Section 1.2.4), it produces two kinds of output: the *hydraulic heads* at the nodes and the *flows* in the pipes. Hydraulic head quantifies the potential energy of the water due to elevation (and added pressure through pumping facilities) [80].

In this study the *EPANET 2* simulator by Rossman [73] is used, specifically the `EPANET-Linux-1.5` port by López-Ibáñez[1]. Starting from a set of known hydraulic heads at the *reservoir* nodes (namely: the elevation of the reservoir), it calculates the hydraulic head at each *internal* node and the flow in each pipe, while compensating for *friction loss* depending on the choice of *diameters* for the pipes. Pressure at an internal node is attained by the relative elevation of the reservoir node, compared to the elevation of the internal node itself.

EPANET is based on the *Gradient Method* by Todini and Pilati [83], and applies one of three *friction head loss formulas* (cf. Section 1.2.5) for determining pressure loss occurring in the pipes during transport. It is nowadays the de facto network solver used at the Center for Water Systems, University of Exeter (an institute that is very active in research on WDN optimization, publications include [5, 6, 28, 34–37, 51–55, 71, 76, 90, 93]), as well as by many other researchers (e.g., [1, 17, 19, 45, 48, 64–66, 84, 85, 91, 94]). EPANET uses the *steady-state flow* model that considers fluid properties at all points in the network to be constant over time. Of the different hydraulic

[1] López-Ibáñez' `EPANET-Linux-1.5` C implementation can be obtained via `http://iridia.ulb.ac.be/~manuel/epanetlinux.html`

models in use, this is the biggest simplification of a real WDN, but also the least difficult to mathematically formulate and the least expensive to solve [53].

We give the formal definition of a *simulated* WDN design, which is the combination of *constants* determined by the network definition, values that are calculated by the hydraulic simulator (*simulator variables*), and values that are set by the multiobjective optimizer (*decision variables*). Furthermore, we derive the hydraulic equation system of the steady-state flow model, address the friction head loss formula, and discuss unit (i.e., of measurement) related technicalities.

1.2.1 Constants

The following values are immutable between different WDN designs (i.e., that are based on the same network definition). They are not changed by the hydraulic simulator or other components of the optimization loop.

m^{exist} number of *existing* pipes in the network (needed in case of a parallel expansion problem, else $m^{exist} = 0$);

m^{new} number of *new* pipes in the network;

m total number of pipes in the network, i.e., $m = m^{exist} + m^{new}$;

r number of *reservoir* nodes in the network;

n number of *internal* nodes in the network;

$r + n$ total number of nodes in the network;

L_i length of pipe i;

b_i begin node of pipe i;

e_i end node of pipe i;

C_i roughness coefficient of pipe i, for use in the *friction head loss formula* (cf. Section 1.2.5); formally not a constant for new pipes, i.e., the pipes $i \in \{m^{exist} + 1, \dots, m^{exist} + m^{new}\}$, but in the examined test problems all available diameter triples have the same roughness (cf. Chapter 4);

Z_j elevation of node $j \in \{1, \dots, r + n\}$;

Q_j water demand drawn from internal nodes, i.e., the nodes $j \in \{r + 1, \dots, r + n\}$;

p^{zero} pressure corresponding to zero nodal water demand satisfaction;

p^{req} pressure required to satisfy the nodal water demand completely.

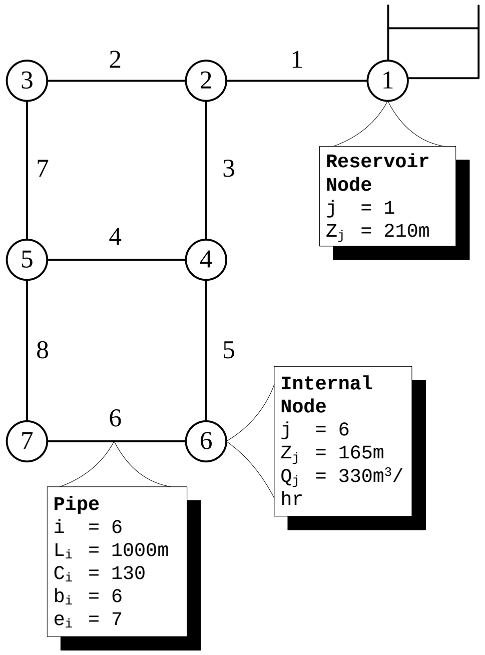


Figure 1.2: **Example Network Layout**
Network layout for test problem Two Loop. Tables of associated node data, pipe data, and available commercial diameters are listed in Chapter 4.

For an illustration of constants in a network layout, see Figure 1.2.

1.2.2 Simulation Variables

The following variables are calculated or can change in the simulation of a WDN design:

p_j effective pressure remaining at internal node $j \in \{r+1, \ldots, r+n\}$;

h_j hydraulic head (i.e., potential energy of the water) at node $j \in \{1, \ldots, r+n\}$, where:

$$h_j = \begin{cases} Z_j & j \leq r \\ Z_j + p_j & \text{else} \end{cases} \tag{1.1}$$

q_i water flow in pipe i, from $orig(i)$ to $dest(i)$;

$orig(i)$ origin of the water flow in pipe i, where:

$$orig(i) = \begin{cases} b_i & q_i \geq 0 \\ e_i & q_i < 0 \end{cases} \tag{1.2}$$

$dest(i)$ destination of the water flow in pipe i, where:

$$dest(i) = \begin{cases} e_i & q_i \geq 0 \\ b_i & q_i < 0 \end{cases} \tag{1.3}$$

1.2.3 Decision Variables

The following variables drive the optimization loop and are varied by the multiobjective optimizer, but are fixed per WDN design:

D_i selected diameter code for new pipe $i \in \{m^{exist} + 1, \ldots, m^{exist} + m^{new}\}$, indicating the size/price/roughness entry from the table of available commercial diameters supplied with the (test) problem (cf. Chapter 4); for existing pipes $i \in \{1, \ldots, m^{exist}\}$ D_i is defined as well, but cannot be varied by the optimizer.

1.2.4 Hydraulic Equation System

Two sets of equations have to be satisfied [43], the first concerning *hydraulic head loss* in the pipes (i.e., pressure loss, in correspondence with the law of conservation of energy [80]) and the second about maintaining *flow continuity* around the nodes (in correspondence with conservation of mass [62]). Hydraulic head loss in all pipes $i \in \{1, \ldots, m\}$ is given by:

$$loss_i = h_{orig(i)} - h_{dest(i)} = K_i |q_i|^\alpha \tag{1.4}$$

where:

K_i resistance coefficient of pipe i, depending on pipe roughness, length, and the selected diameter (i.e., $K_i \equiv K(C_i, L_i, D_i)$); the exact form of this coefficient is determined by the used *friction head loss formula* (cf. Section 1.2.5);

α exponent of the friction head loss formula.

Flow continuity around all internal nodes $j \in \{r + 1, \ldots, r + n\}$ is defined as:

$$f_j = \sum_{i:\, j=dest(i)} |q_i| - \left(\sum_{i:\, j=orig(i)} |q_i| \right) - Q_j = 0 \tag{1.5}$$

This rule basically means that the flow into a node must equal the flow going out of that node.

For a set of known hydraulic heads at the reservoir nodes, the remaining heads and the flows are sought that satisfy Equation 1.4 and 1.5. This can be done by combining these equations and reformulating the system as a minimization problem. First, isolate $|q_i|$ in Equation 1.4:

$$|q_i| = \left(\frac{loss_i}{K_i}\right)^{\frac{1}{\alpha}} = \left(\frac{h_{orig(i)} - h_{dest(i)}}{K_i}\right)^{\frac{1}{\alpha}} \tag{1.6}$$

Then, replace all occurrences of $|q_i|$ in Equation 1.5 by the righthand term of Equation 1.6:

$$f_j = \sum_{i:\, j=dest(i)} \left(\frac{h_{orig(i)} - h_{dest(i)}}{K_i}\right)^{\frac{1}{\alpha}} - \sum_{i:\, j=orig(i)} \left(\frac{h_{orig(i)} - h_{dest(i)}}{K_i}\right)^{\frac{1}{\alpha}} - Q_j = 0 \tag{1.7}$$

Next we define the following system calculating the sum over all n squared errors around zero (i.e., solves Equation 1.7 for every internal node):

$$\sum_{j=r+1}^{r+n} (f_j - 0)^2 \tag{1.8}$$

Now we can express the hydraulic simulator as a minimization problem solving the non-linear equation system by minimizing residuals; a tuple of hydraulic heads at the internal nodes $j \in \{r+1, \ldots, r+n\}$ is sought that minimizes the sum of the n squared errors around zero:

$$\underset{(h_{r+1},\ldots,h_{r+n})\in\mathbb{R}^n}{\arg\min} \sum_{j=r+1}^{r+n} (f_j(h_1, \ldots, h_{r+n}))^2 \tag{1.9}$$

This steady-state flow system is non-linear of order n [43], and can be solved using iterative numerical methods such as *Newton-Raphson* [70] and *Hardy-Cross* [3]. In EPANET the Gradient Method employs a sparse matrix method by George and Liu [46] to solve the non-linear equation system.

Leakage of water flow in the network is an important factor from an economical standpoint [33]. Note that Equation 1.5 assumes no *flow losses* due to leakage occur in the pipes, except losses implicitly included in the nodal demands [88]. A more accurate expression would be to include a *pressure-dependent* leakage term $\psi_i\left(\frac{p_{orig(i)}+p_{dest(i)}}{2}\right)$ [89, 92] accounting for the flow of water through pipe i not arriving at its destination node

$dest(i)$:

$$f_j = \sum_{i:\, j=dest(i)} \left(|q_i| - \psi_i \left(\frac{p_{orig(i)} + p_j}{2} \right) \right) - \sum_{i:\, j=orig(i)} |q_i| - Q_j = 0 \tag{1.10}$$

where p_j is the effective pressure remaining at node j:

$$p_j = h_j - Z_j \tag{1.11}$$

As in EPANET the assumption of no (separate) leakage flow loss occurring is made as well [73], we hold on to the formal definition of the hydraulic simulator given in Equation 1.9.

1.2.5 Friction Head Loss Formula

In an ideal setting, the hydraulic head remains the same at all nodes and does not decrease during transportation of the water through the network. In a real situation however, some potential energy of the water is converted into heat due to friction occurring in the pipes, depending on *pipe roughness*. The friction head loss formula calculates this pressure loss.

With EPANET we have the choice among three head loss formulas. The *Hazen-Williams* formula is the most commonly used head loss formula in the United States. It cannot be used for liquids other than water and was originally developed for *turbulent* flow only. The *Darcy-Weisbach* formula is the theoretically most accurate. It applies over all flow regimes (turbulent and its inverse, *laminar*) and to all liquids. The *Chezy-Manning* formula is more commonly used for *open channel flow* (e.g., through rivers and canals). [73]

In this study the widely adopted, empirically derived Hazen-Williams equation [76] is used:

$$loss_i = \omega \frac{L_i}{C_i^{\alpha} size(D_i)^{\beta}} |q_i|^{\alpha} \tag{1.12}$$

where:

ω numerical conversion constant, depending on the units used (cf. Section 1.2.6);

α	exponent taken to be 1.852 [73];
β	exponent taken to be 4.871 [73];
$\mathit{size}(D_i)$	diameter size corresponding to diameter code D_i.

The resistance coefficient K_i of pipe i (mentioned in Section 1.2.4) then becomes:

$$K_i = \omega \frac{L_i}{{C_i}^{\alpha}\mathit{size}(D_i)^{\beta}} \tag{1.13}$$

Note that in the Hazen-Williams equation, the value of the pipe roughness coefficient C_i is inversely related to the actual pipe roughness, e.g., an increase in pipe roughness leads to a lower value of the pipe roughness coefficient C_i.

1.2.6 Units

In principle arbitrary units may be used in a test problem for each of its properties (cf. Chapter 4). Dependences exist between certain pairs of properties however, with the requirement of using the same unit for both. Whenever a dependency exists but different units are used, values will have to be converted using conversion factors[2]. The following dependences are to be accounted for:

- $Z_j \Rightarrow \begin{cases} h_j \\ L_i \\ p^{zero} \wedge p^{req} \wedge p_j \end{cases}$
- $q_i \Rightarrow Q_j$
- $\mathit{size}(D_i) \wedge q_i \Rightarrow \omega$

Per property we discriminate between the unit used in the calculation (*calculation* unit) and the unit defined in the test problem (*defined* unit). Inputs for which dependences exist (e.g., L_i, Q_j, ω) will have to be converted from the defined unit to the calculation unit prior to calculation; outputs for which dependences exist (e.g., p_j) will have to be converted from the calculation unit to the defined unit after calculation.

[2] For an exhaustive listing, visit `http://www.asknumbers.com`

The numerical conversion constant ω is a special input case as it is unitless. In EPANET ω is defined as 4.727 for $size(D_i)$ in ft and q_i in cfs [73]. The numerical constant can be converted for usage with different units as follows:

$$\omega = \frac{convQ^{\alpha}}{convD^{\beta}} \cdot 4.727 \tag{1.14}$$

where:

$convQ$	conversion factor from defined unit for q_i to cfs;
$convD$	conversion factor from defined unit for $size(D_i)$ to ft.

For example, when q_i is expressed in m^3/s and $size(D_i)$ in m (i.e., the default combination in literature for comparing used ω values [76]), ω becomes:

$$\omega = \frac{convQ^{1.852}}{convD^{4.871}} \cdot 4.727 = 10.667 \tag{1.15}$$

where:

$convQ = 35.31466$	$\text{m}^3/\text{s} \rightarrow$ cfs;
$convD = 3.28083$	m $\rightarrow$ ft.

1.3 Objective Functions from Related Work

We list a selection of objective functions taken from literature, concerning network cost and network reliability. For reliability, functions are included that express the extent to which a network design exceeds the basic requirements concerning reliability, but also functions that express the degree by which the network is unreliable, so that the designer can make a cost-effective trade-off.

The aim for reliable networks can have adverse effects on water quality, as oversized distribution mains and storage tanks will have negative effects on water age due to low flow velocity and little turnover [37]. In practice, water authorities need to satisfy water quality criteria, in addition to delivering nodal demands at the required pressure. However, water quality-based optimization has a much higher computational burden, relative to hydraulics [17]. In this work water quality is left out of consideration, hence we do not include objectives for water quality in this overview.

Not all objective functions considered useful necessarily need to be used in the optimization process. Next to the actual optimization, the design process consists of an elaborate decision phase, in which the results of the optimization need to be analyzed and alternatives weighed. In order to discriminate between solutions, complementary definitions of quality can proof helpful.

1.3.1 Functions of Cost

The expenses for the main hydraulic components of the WDN are divided into investment costs and operational costs [43]. The investment costs, which are distributed along the useful life of the network, cover the acquisition costs and the costs of installation of the pipes, pumps, and tanks. The operational costs depend for a significant part on energy consumption by motor pumps [43, 48, 93].

In this study we only consider investment cost. No pumps and tanks are included in the benchmark test problems, hence investment cost is determined by the expenses related to procurement and installation of the pipes.

1.3.1.1 Investment Cost

Investment Cost is expressed as the lump sum needed for inserting the new pipes in the network, priced in segments of unit length with a certain diameter [43]:

$$F_{IC} = \left[\sum_{i=m^{exist}+1}^{m^{exist}+m^{new}} price(D_i) L_i \right] \rightarrow min \tag{1.16}$$

where:

$price(D_i)$ unit price of new pipe $i \in \{m^{exist}+1, \ldots, m^{exist}+m^{new}\}$; the prices of the diameter sizes are included in the table of available commercial diameters, which is supplied with the test problem.

1.3.2 Functions of Reliability

Whenever there is a mechanical or hydraulic failure, the internal head losses will increase causing failure of the network. These increased head losses during failure conditions can be met, if sufficient excess power is available for internal dissipation [71]. Furthermore, excess capacity is valuable in case of the need to respond to increasing demands in future years. Alternatively, when budget constraints are strict, a means is required to determine the best less than optimal solution with respect to the actual pressures at which demands of certain non-critical nodes are serviced. Next to these two types of pressure-related objective functions, we include two measures of the redundancy available in the network for countering mechanical failure.

1.3.2.1 Maximum Individual Head Deficiency

The head deficiency at node j is the extent to which the available pressure at that node comes short of attaining the required service level p^{req}:

$$defic_j = (Z_j + p^{req}) - h_j \tag{1.17}$$

In case of multiple *demand patterns* (which are defined in some test problems in literature, e.g., peak-average-minimum, 24 one-hour time steps), the largest deficiency over all patterns is selected per node [28]:

$$defic_j = \max_{k \in \{1,\ldots,N^{dp}\}} [(Z_j + p^{req}) - h_{j,k}] \tag{1.18}$$

where:

N^{dp} number of demand patterns;

$h_{j,k}$ hydraulic head at node j under demand pattern k.

Maximum Individual Head Deficiency [28, 35] is then defined as minimization of the head deficiency at the most depressed internal node j:

$$F_{mIHD} = \left[\max_{j \in \{r+1,\ldots,r+n\}} defic_j\right] \rightarrow min \tag{1.19}$$

1.3.2.2 Sum of Maximum Individual Head Deficiency

In [18] a slightly different formulation of Equation 1.19 is used, which we term *Sum of Maximum Individual Head Deficiency.* Instead of dealing with the multiple patterns within the definition of defic_j, as in Equation 1.18, the sum of the Maximum Individual Head Deficiency over all patterns is taken, based on the original Equation 1.17, and which is to be minimized [18]:

$$F_{smIHD} = \left[\sum_{k=1}^{N^{dp}} \max_{j \in \{r+1,\ldots,r+n\}} \mathit{defic}_{j,k} \right] \rightarrow min \tag{1.20}$$

where:

$\mathit{defic}_{j,k}$ head deficiency at node j under demand pattern k (cf. Equation 1.17).

1.3.2.3 Total Head Deficit

Using Equation 1.17 or 1.18, we can express the *Total Head Deficit* [4, 51, 52, 55, 74] as the sum of the head deficiencies at the internal nodes in the network, which is to be minimized:

$$F_{tHD} = \left[\sum_{j=r+1}^{r+n} \mathit{defic}_j \right] \rightarrow min \tag{1.21}$$

1.3.2.4 Number of Nodes with Head Deficiency

Using Equation 1.17 or 1.18, the *Number of Nodes with Head Deficiency* [34, 74], to be minimized, is given by:

$$F_{NHD} = \big[\big| \{ \mathit{defic}_j > 0 \,|\, j \in \{r+1, \ldots, r+n\} \} \big| \big] \rightarrow min \tag{1.22}$$

1.3.2.5 Minimum Surplus Head

The surplus head at node j is the excess pressure available above the required service level p^{req}. This surplus head indicates the available energy for dissipation during failure conditions and is given by:

$$surplus_j = h_j - (Z_j + p^{req}) \tag{1.23}$$

Minimum Surplus Head is then defined as maximization of the surplus head at the most depressed internal node j [71, 82]:

$$F_{mSH} = \left[\min_{j \in \{r+1,\dots,r+n\}} surplus_j \right] \rightarrow max \tag{1.24}$$

1.3.2.6 Total Surplus Head

Based on Equation 1.23, *Total Surplus Head* is a counterpart to Minimum Surplus Head, instead maximizing the summation of the surplus head at each internal node j [71]:

$$F_{tSH} = \left[\sum_{j=r+1}^{r+n} surplus_j \right] \rightarrow max \tag{1.25}$$

1.3.2.7 Weighted Demand Supply Ratio

Weighted Demand Supply Ratio tells us to what extent the water demand at the various internal nodes in the network is met. In order to determine the portion of its water demand a node is capable of servicing, a functional relationship between pressure and water demand at a node needs to be established (cf. Figure 1.3). Let ${Q_j}^{met}$ be the portion of the water demand met at node j [81]:

$${Q_j}^{met} = \begin{cases} 0 & p_j < p^{zero} \\ Q_j \sqrt{\frac{p_j - p^{zero}}{p^{req} - p^{zero}}} & p^{zero} \leq p_j \leq p^{req} \\ Q_j & p_j > p^{req} \end{cases} \tag{1.26}$$

We can then determine the average Demand Supply Ratio [43], weighted by the nodal demands Q_j:

$$F_{wDSR} = \left[\frac{\sum_{j=r+1}^{r+n} {Q_j}^{met}}{\sum_{j=r+1}^{r+n} Q_j} \right] \rightarrow max \tag{1.27}$$

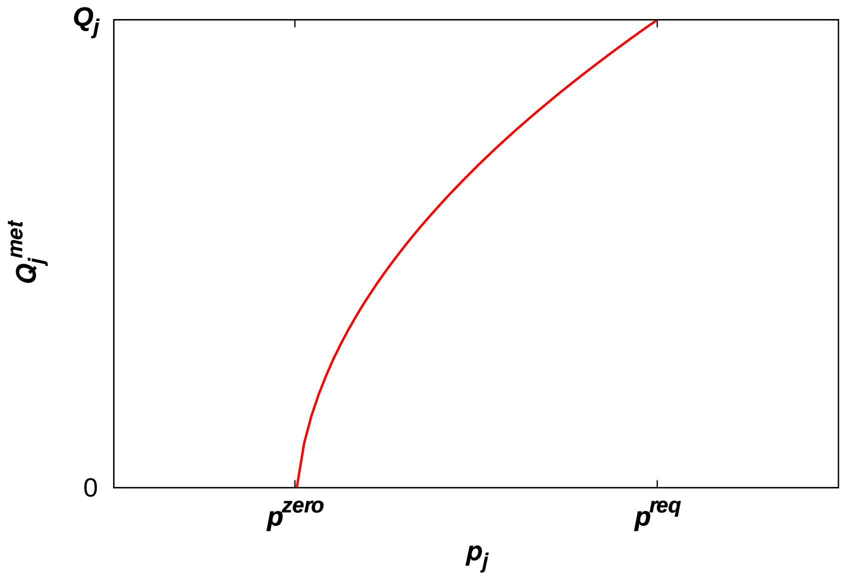


Figure 1.3: **Functional Relationship $Q_j{}^{met}$**
Plot of $Q_j{}^{met}$ for test problem Two Loop (cf. Chapter 4), which defines $p^{zero} = 10$ and $p^{req} = 30$.

1.3.2.8 Resilience Index

In [82] a *Resilience Index* is proposed based on the concept that the total power put into a network is equal to the total power lost through friction plus the total power delivered at the internal nodes to service demands [36, 37, 71]:

$$P^{in} = P^{fric} + P^{del} \tag{1.28}$$

Let $supplied_j$ be the total demand supplied by a reservoir node $j \in \{1, \ldots, r\}$:

$$supplied_j = \sum_{i:\, j=orig(i)} |q_i| \tag{1.29}$$

The total power put into a network is then given by the sum over the products of head and supplied demand at the reservoir nodes $j \in \{1, \ldots, r\}$, where γ is the specific weight of water:

$$P^{in} = \gamma \sum_{j=1}^{r} supplied_j \cdot h_j \tag{1.30}$$

The total power delivered to the internal nodes $j \in \{r+1, \ldots, r+n\}$ is given by:

$$P^{del} = \gamma \sum_{j=r+1}^{r+n} Q_j \cdot h_j = \gamma \sum_{j=r+1}^{r+n} Q_j \cdot (Z_j + p_j) \tag{1.31}$$

The variant of P^{del} corresponding to network-wide delivery at required pressure p^{req} is:

$$P^{del,req} = \gamma \sum_{j=r+1}^{r+n} Q_j \cdot (Z_j + p^{req}) \tag{1.32}$$

We can now define the Resilience Index of a network as follows:

$$F_{RI} = \left[\frac{1 - P^{fric}}{P^{fric,req}} \right] \rightarrow max \tag{1.33}$$

Here P^{fric} can be calculated using Equation 1.28, and $P^{fric,req}$ is defined analogously based on $P^{dem,req}$. Hence the aim is to minimize the total power lost through friction P^{fric}, effectively maximizing the total power delivered to the nodes.

1.3.2.9 Network Resilience

Maximization of surplus head alone is not sufficient for a reliable network. A branched network with sufficient surplus head at the nodes may adjust to increased demands, but a pipe outage will have severe consequences on the downstream nodes. Hence in [71] *Network Resilience* is proposed, incorporating the effects of both surplus power and redundancy through reliable loops. The surplus power at an internal node j is given by:

$$P_j = \gamma \cdot Q_j \cdot surplus_j \tag{1.34}$$

Reliable loops are ensured if the pipes connected to a node are not widely varying in diameter, which we express as the uniformity $unif_j$ of a node j:

$$unif_j = \frac{\sum_{i:\, j=b(i) \vee e(i)} size(D_i)}{|\{i : j = b(i) \vee e(i)\}| \cdot \max[size(D_i)]} \tag{1.35}$$

The uniformity of a node j is taken as the sum over the diameter sizes of all pipes connected to it (i.e., node j is connected to the beginning ($j = b(i)$) or the end of pipe i ($j = e(i)$)), divided by the number of pipes connected to j times the maximum diameter size of such a pipe. If the pipes connected to node j all have the same diameter, or only a single pipe is connected to j, the uniformity of j is equal to 1. If the pipes connected to node j differ in diameter, the value of $unif_j$ is smaller than 1.

We can now define the combined effect of surplus power and nodal uniformity, termed *nodal weighted surplus power*:

$$X^{nod}{}_j = unif_j \cdot P_j \tag{1.36}$$

For the entire network, the *weighted surplus power* X and *maximum surplus power* X^{max} are given by:

$$\begin{aligned} X &= \sum_{j=1}^{r+n} X^{nod}{}_j && (1.37)\\ X^{max} &= P^{in} - P^{del,req} && (1.38) \end{aligned}$$

where:

P^{in} total power put into the network, cf. Equation 1.30;

$P^{del,req}$ total power delivered to the nodes corresponding to network-wide demand delivery at p^{req}, cf. Equation 1.32.

We then get the Network Resilience by maximizing the normalized weighted surplus power:

$$F_{NR} = \left[\frac{X}{X^{max}}\right] \rightarrow max \tag{1.39}$$

When the variables in Equation 1.33 and Equation 1.39 are substituted by the appropriate terms, it can be seen that Network Resilience is equivalent to the Resilience Index, but with surplus power at each node weighted by uniformity [71].

1.3.2.10 System Entropy

The water demand at a given internal node is ideally met using multiple different paths to that node. The required flow should be distributed over these routes as evenly as possible. This way, should a segment of the network fail, alternative routes exist that could still supply a reasonable part of the demand. Let in_j be the water flow transported towards node j by the incoming pipes i connected to it (thus for reservoir nodes $in_j = 0$):

$$in_j = \sum_{i:\, j=dest(i)} |q_i| \tag{1.40}$$

Let IN be the sum of all incoming flows in the network:

$$\mathrm{IN} = \sum_{j=1}^{r+n} in_j \tag{1.41}$$

Let s_j be the *entropy* of node j, defined as:

$$s_j = \sum_{i:\, j=dest(i)} \frac{-|q_i|}{in_j} \ln \frac{|q_i|}{in_j} \tag{1.42}$$

Let S be the entropy of the entire network:

$$S = \sum_{j=1}^{r+n} \frac{in_j}{\mathrm{IN}} (s_j - \ln \frac{in_j}{\mathrm{IN}}) \tag{1.43}$$

System Entropy is then expressed as the exponential of S, increasing the range for easier comparison of different solutions [43]:

$$F_{SE} = [\exp S] \rightarrow max \tag{1.44}$$

1.4 Multiobjective Optimization Approach

The biggest hurdle faced in water distribution network design is predicting future demands [71]. The population of the area supplied by the WDN may increase or shrink in future years, extended periods of drought can occur. Even if these developments could be reasonably well predicted, their exact relation to the increase or decrease of the demands is uncertain [36]. To give an appreciation of the need of preparing for future demand, consider the planning and construction of the New York City Third Tunnel WDN that started in 1969 [23, 77] and is expected to be complete by 2020[3], a 50 years plus horizon for the construction phase only.

Thus, a designer would like to provide as much excess head at the demand nodes as possible, subject to monetary constraints [71], but also adhering to tolerance levels [38]. Too high pressure increases leakage [92], component wear, and the likelihood of encountering steady state and water hammer pipe failures. Appropriate surplus head

[3] Information on the current status of the New York City Third Tunnel WDN available on the NRI Water Technology website: http://www.water-technology.net/projects/new_york/

is utilized to overcome the gain in head loss under increasing demand or other failure conditions (cf. Section 1.1). Certain critical nodes may provide more future benefit from excess head than others, depending on the actual changes occurring, e.g., in the demand patterns and with respect to pipe deterioration. Contrary, a designer also wants insight into non-critical nodes that could be supplied at a certain pressure below the required level p^{req} without service being impaired, providing for budgetary flexibility. From a water network design perspective, a slight pressure deficit in some non-strategic nodes (feeding non-sensitive users, e.g., other than hospitals, schools or firefighting hydrants) is often outweighed by the corresponding significant cost reduction [28].

Two approaches for safeguarding reliability of a WDN design are described in literature: through objective functions expressing reliability, next to cost, in a multiobjective optimization model (*deterministic model*), and via the repeated evaluation of WDN designs with varied nodal demands and pipe roughness of the existing pipes (*stochastic model).* While the stochastic model generally requires more evaluations by the hydraulic simulator than the deterministic model in order to converge, it has a better chance of finding the critical and non-critical nodes in the network (i.e., the nodes that benefit most from surplus head and suffer least from head deficiency, respectively), appreciating limited capital resources.

In this work we aim to compare the two models for reliable WDN optimization. This is done by running separate experiments following the deterministic and the stochastic model. In the stochastic model, the solutions in the final result sets are tested for *robustness* by evaluating them on a large sample set of varied environmental parameters (i.e., the nodal demands and roughness of existing pipes), delivering robust approximations of the objectives depending on the environmental parameters. In the same manner, the final result sets of the deterministic experiments can be tested for robustness as well, which provides us with the means of comparing the results obtained using both models. We first lay out the deterministic approach used in this study, and then discuss the stochastic approach.

1.4.1 Deterministic Approach

The deterministic models for reliable WDN optimization encountered in literature are multiobjective schemes applying the cost function Investment Cost (cf. Section 1.3.1) and an author dependent selection of the objective functions available for attaining and/or comparing solutions on reliability (cf. Section 1.3.2). We distinguish between the F_{IC}, F_{NR} approach [71] (reliability through Network Resilience, which we fill refer to as the method *Prasad*) and the F_{IC}, F_{wDSR}, F_{SE} approach [43] (reliability through Weighted Demand Supply Ratio and System Entropy, which we will refer to as the method *Formiga*). These two methods have in common that they express reliability in terms of both pressure and redundancy of pipes in the network.

The two objective method Prasad accounts for hydraulic reliability through surplus head (cf. Section 1.3.2.9), while the three objective method Formiga ignores surpluses above the required level p^{req} and only provides insight into the degree of fulfillment of the total demand, based on the available nodal pressures (cf. Section 1.3.2.7). It should be noted that when using the method Prasad, seemingly efficient in the sense that it combines redundancy and surplus head in one objective (i.e., a smaller number of objectives generally makes for an easier multiobjective optimization problem), it is necessary to enforce a constraint on the nodal pressures p_j, namely, $p_j \geq p^{req}$ [71]. We must make sure that the nodal heads are at least at the required level, otherwise there is the possibility of surpluses at certain nodes canceling out deficits at other nodes, potentially leaving us with consumer nodes that are not serviced at all. The use of a constraint however introduces discontinuities in the search space of possible solutions, thereby increasing the difficulty of the optimization task.

Comparing the two methods concerning mechanical reliability, we find that the entropy of a node (cf. Section 1.3.2.10) is a more convenient formulation than the nodal uniformity (cf. Section 1.3.2.9). First of all, nodal entropy focusses only on the incoming flows to the consumer (i.e., internal) node in question and ignores the outgoing flows, as they are of no particular importance to that node. On the other hand, nodal uniformity treats all pipes connected to a node as being of equal importance, that is, it groups incoming and outgoing pipes together. Furthermore, although this difference is subtle, for measuring redundancy in a network, the equal spread of the incoming water flow over the pipes connected to a node seems more significant than the equal actual diameter size of the pipes transporting the, possibly varying, flows to a node. Namely, we are interested in the part of the demand that would still be serviced should one of the incoming sources fail, and considering the remaining incoming flows is then most straightforward.

Resuming, method Prasad encompasses only two objective functions, where method Formiga defines three, but the former requires enforcing a constraint on the nodal pressures which introduces added complexity. Furthermore, Prasad takes surplus heads into account where Formiga does not, but cannot be used for determining non-critical nodes to be supplied at below required pressures, valuable from an economical point of view. Moreover, the Prasad definition of surplus head does not take adhering to tolerance levels into account (cf. Section 1.3.2.9). Lastly, Formiga provides a better formulation of network redundancy. Hence, although it ignores surplus heads, in this work the unconstrained, three-dimensional method Formiga is adopted for the deterministic approach: Investment Cost (F_{IC}), Weighted Demand Supply Ratio (F_{wDSR}), and System Entropy (F_{SE}).

1.4.2 Stochastic Approach

Stochastic WDN optimization deals with two instances of *environmental* uncertainty (i.e., uncertainty related to changes in the environment, contrary to noise in the decision variables [16]). First of all, there is the inherent uncertainty in predicting future nodal demands; secondly, pipe roughnesses can increase significantly with age depending on the pipe material and corrosive properties of the water transported [5]. Stochastic approaches for WDN optimization can roughly be divided into two classes: methodologies that use some analytically based technique for the propagation of uncertainties to the simulator output values (cf. Figure 1.1) (e.g., [6, 13, 85, 90]), and methodologies that use stochastic simulation, that is, sampling-based techniques to quantify uncertainties (e.g., [47, 54]). Although the sampling-based techniques are typically several orders of magnitude slower than the analytical approaches, the application of stochastic simulation is more universal and straightforward, not depending heavily on the characteristics of the test problem being examined [5]. For this reason, we restrict ourselves to sampling-based techniques.

We consider environmental uncertainty in the consumer demands at the nodes and in the roughness coefficients of the *existing* pipes, hence the latter only concerns parallel expansion problems. The following probability distributions are used [5, 54]:

- consumer demands: Gaussian distribution with mean equal to the given consumer demand, and standard deviation equal to 10% of the mean (i.e., coefficient of variation $C_v = 0.10$);
- roughness coefficients of existing pipes: uniform distribution, defined on an interval of ±10% of the given roughness coefficient.

The stochastic model is often implemented as a two-dimensional multiobjective model, using a second objective next to Investment Cost that expresses the robustness R of a design, as follows [54]:

$$R = \prod_{j=r+1}^{r+n} P(p_j \geq p^{req}) \tag{1.45}$$

Here the design robustness is defined as the probability that the pressure at all internal consumer nodes is equal to or above the required minimum service level p^{req}, where P stands for that probability per node. When a sampling-based technique is used, evaluating and averaging each design over N^{sample} samples, the second objective becomes:

$$\left[\frac{1}{N^{sample}} \sum_{k=1}^{N^{sample}} R \right] \rightarrow max \tag{1.46}$$

In this study, we instead use a natively multiobjective robust approach, building on the deterministic multiobjective Formiga method. Varying environmental parameters does not influence Investment Cost (F_{IC}), like it remains uninfluenced in the stochastic model described above, but in our approach the robustness of a design is expressed in terms of the robust counterparts of Weighted Demand Supply Ratio (F_{wDSR}) and System Entropy (F_{SE}). These are obtained as the average over random samples (cf. Section 2.2), like in Equation 1.46. An interesting property of the method Formiga is that when using varying nodal consumer demands in combination with F_{wDSR}, the latter effectively does account for surplus head at critical nodes through peaks included in the random demand samples (where it does not when applied in the deterministic approach).

1.4.3 Multiobjective Evolutionary Algorithms

In this study Evolutionary Algorithms (EAs) are applied as multiobjective optimizer. EAs are population-based stochastic optimizers, inspired by the principles of biological evolution. The variants used here are specialized for multiobjective problems (cf. Chapter 3). In recent years, EAs have become the standard for WDN optimization, motivated by the fact that they can natively deal with the discrete nature of WDN optimization. The candidate solutions, which are vectors of diameter codes, are taken as the *individuals*, where the *genes* are the pipes in the network and the *alleles* are the possible diameter codes. While here only considering pipes in the WDN optimization problem, other components can easily be added to the vectors of decision variables (i.e., individuals) when using EAs. With regard to the stochastic approach, EAs can be straightforwardly extended with methods for robust evaluation (cf. Section 2.2), optimizing over robust approximations of the fitness instead. Note that instead of referring to the deterministic and stochastic approach (terms commonly used in the field of WDN optimization), we will speak of the multiobjective *raw* and multiobjective *robust* approach (the terms common in the robust optimization field).

Chapter 2

Multiobjective Robust Optimization

As WDN optimization is formulated as a problem involving multiple objectives, we need a method of comparing candidate solutions using multiple, possibly conflicting, expressions of quality. Also, a means of comparing sets of solutions found by different (runs of) *a posteriori* multiobjective optimizers is required. Both topics are addressed in Section 2.1.

Furthermore, as this study aims at comparing the basic multiobjective approach with the multiobjective robust approach of WDN optimization, robust optimization is formalized and methods for robust evaluation of candidate solutions within the framework of an multiobjective optimizer are discussed in Section 2.2.

2.1 Multiobjective Optimization

When dealing with a problem that considers multiple objective functions to compare candidate solutions on quality, we can map the objective functions to a single quality measure, and then let the optimization algorithm search for a single best solution. This *a priori* approach requires preferences to be formulated beforehand regarding the objectives involved. However, stating preferences can be a difficult task in multiobjective optimization, especially if the knowledge about the structure of the set of optimal solutions is incomplete [11]. On the other hand, by adopting the *a posteriori* approach, the decision maker postpones the final decision on preferences between conflicting objectives until having been presented with a set of compromise solutions. Such sets can be obtained by *Pareto optimization* [20, 24]. The aim herein is to approximate the *Pareto-optimal set* that consists of all solutions that cannot be improved in any one objective without degradation in another [99].

In [11] a formal description of Pareto optimization is given, concerning optimization of a problem comprising n_f objective functions $\mathbf{f} = (f_1, \ldots, f_{n_f})$, with $f_i : \mathbb{S} \to \mathbb{R}$ for all $i \in \{1, \ldots, n_f\}$, and $\mathbb{S}$ denoting the *search space* (or: *decision space*) of the problem. The WDN benchmark test problems examined in this study have a search space $\mathbb{S}$ consisting of discrete values, and involve three objective functions, one of which is to be minimized and the rest to be maximized (cf. Chapter 1 for the definition of the WDN optimization model, cf. Chapter 4 for the benchmark test problems). For notational clarity, but without loss of generality, we here consider an all minimization problem, as in [11]. The *dominance relation* ($\prec$), which defines a partial order on the space of objective function value vectors $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^{n_f}$, is then defined as:

$$\mathbf{y} \prec \mathbf{y}' \Leftrightarrow \left(\forall i \in \{1, \ldots, n_f\} : y_i \leq y'_i\right) \wedge \left(\exists j \in \{1, \ldots, n_f\} : y_j < y'_j\right) \tag{2.1}$$

In words, $\mathbf{y}$ is said to *dominate* $\mathbf{y}'$, if and only if 1) none of the objective function values contained in $\mathbf{y}$ are greater than their counterpart contained in $\mathbf{y}'$, 2) at least one objective function value in $\mathbf{y}$ is smaller than its counterpart in $\mathbf{y}'$. For *decision variable* vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{S}$ we write:

$$\mathbf{x} \prec \mathbf{x}' \Leftrightarrow \mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}') \tag{2.2}$$

That is, $\mathbf{x}$ is said to dominate $\mathbf{x}'$, if and only if the objective function value vector that is associated with it dominates the objective function value vector associated with $\mathbf{x}'$. The Pareto-optimal set M of the problem consists of the minimal elements of the dominance relation, given by [30]:

$$M = \left\{\mathbf{x} \in \mathbb{S} \,|\, \nexists \mathbf{x}' \in \mathbb{S} : \mathbf{x}' \prec \mathbf{x}\right\} \tag{2.3}$$

The corresponding image under $\mathbf{f}$ in the *objective space* $\mathbb{R}^{n_f}$ is called the *Pareto front* of the problem. Furthermore, a point in the search space (i.e., a vector of decision variables) is called *non-dominated* within a set $A \subseteq \mathbb{S}$ if there is no point in A dominating it. Hence, the Pareto-optimal set M consists of all points that are non-dominated in $\mathbb{S}$.

2.1.1 Hypervolume Measure

The *hypervolume measure* or $\mathcal{S}$ *metric* is a means to compare the result sets (i.e., sets of decision variable vectors) of different multiobjective optimization runs, on the same problem. It indicates the quality of the Pareto front approximation that a set of decision variable vectors gives rise to. It rewards the convergence towards the actual Pareto front, as well as a representative distribution of points along the front [11]. The $\mathcal{S}$ metric was originally proposed by Zitzler and Thiele [101], who called it the *size of dominated space* [30].

Let Λ denote the *Lebesgue measure* [20], then for a set $A \subseteq \mathbb{S}$, the hypervolume measure is calculated as follows [11, 30]:

$$\mathcal{S}(A) = \Lambda \left(\bigcup_{\mathbf{x} \in A} \left\{ \mathbf{y} \in \mathbb{R}^{n_f} \mid \mathbf{f}(\mathbf{x}) \prec \mathbf{y} \prec \mathbf{y}^{ref} \right\} \right) \tag{2.4}$$

We see the hypervolume measure being defined as the Lebesgue measure of the *union* of the hypercubes that are delimited by the *reference point* $\mathbf{y}^{ref} \in \mathbb{R}^{n_f}$, and the images under $\mathbf{f}$ of the points in A. Overlapping parts of *domination cones* contribute only once (cf. Figure 2.1), the domination cone of $\mathbf{x}$ being defined in the objective space by $\{\mathbf{y} \in \mathbb{R}^{n_f} \mid \mathbf{f}(\mathbf{x}) \prec \mathbf{y} \prec \mathbf{y}^{ref}\}$.

The choice of the problem-specific reference point is arbitrary, as long as the reference point is dominated by the image under $\mathbf{f}$ of each member of the problem-specific Pareto-optimal set M (i.e., $\forall \mathbf{x} \in M : \mathbf{f}(\mathbf{x}) \prec \mathbf{y}^{ref}$) [11]. Furthermore, an accurate calculation of the $\mathcal{S}$ metric requires a normalized and positive objective space [30].

Zitzler et al. [102] state that hypervolume measure is so far the only known unary metric that is capable of detecting that a set of decision variable vectors is not worse than another set. In 2007, the *weighted-integration* hypervolume measure was proposed as a generalization of the hypervolume measure, and shares these properties with the original $\mathcal{S}$ metric [98]. Fleischer [39] proved for discrete spaces that the hypervolume measure is maximized if and only if the set of decision variable vectors contains only Pareto optima. On the other hand, hypervolume has some non undesirable properties too: it is sensitive to the relative scaling of the objectives through the choice of reference point, and to the presence or absence of extremal points in a front [15] (cf. Figure 2.2).

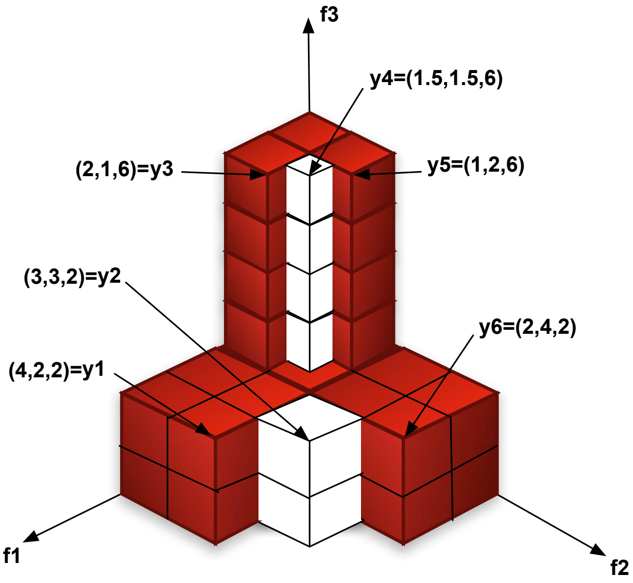
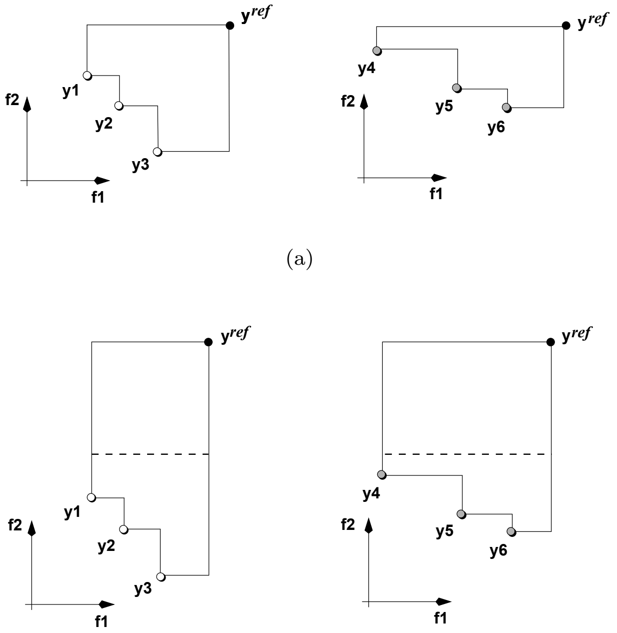


Figure 2.1: **Three-dimensional Hypervolume Example**
Displayed is a maximization problem involving three objectives, with reference point $\mathbf{y}^{ref} = (0, 0, 0)$. Six non-dominated individuals are indicated in the objective space by their objective function value vectors $\mathbf{y}_i$. Exclusively dominated dominated hypervolume per individual $i \in \{1, \ldots, 6\}$ accounts to $\boldsymbol{\Delta}_{\boldsymbol{\mathcal{S}}} = (\Delta_{\mathcal{S}1}, \ldots, \Delta_{\mathcal{S}6}) = (4, 2, 2, 1, 2, 4)$. Total hypervolume for the set $\{1, \ldots, 6\}$ is the sum of the exclusively dominated hypervolumes plus the overlapping parts, $\mathcal{S} = (4{+}2{+}2{+}1{+}2{+}4){+}(4{+}4{+}8){+}(2{+}2{+}4) = 39$. Figure courtesy of [14].

Major pitfall of the $\mathcal{S}$ metric was the time complexity of the original algorithm calculating it, limiting its usefulness for large sets of points or problems with many objectives: $\mathcal{O}(k^{n_f+1})$ [58], with k being the number of objective function value vectors in the examined set and n_f being the number of objective functions of the problem. Better algorithms have been developed since, the current best worst-case time complexity for $n_f = 3$ is $\mathcal{O}(k \log k)$ [10], and the best worst-case time complexity for $n_f > 3$ is $\mathcal{O}(k^{n_f/2} \log k)$ [9].

2.1.1.1 Reference Point for WDN Optimization

For WDN optimization, the choice of a reference point and the requirement of a normalized and positive objective space is handled using vectors of the worst and best possible values per objective, respectively, given a certain WDN test problem, i.e., the *Boundary* and *Ideal* points of the test problem. The Boundary point of a test problem $\mathbf{y}^{BoundTP}$ is calculated by selecting the most expensive pipe diameter for all new pipes in the network, setting the water demand met to 0 for all internal nodes, and taking the entropy



(a)

(b)

Figure 2.2: **Choice of Reference Point**
The relative value of the $\mathcal{S}$ metric depends upon the choice of reference point $\mathbf{y}^{ref}$. Two non-dominated sets, $A = \{1, 2, 3\}$ and $B = \{4, 5, 6\}$, are displayed by their images in the objective space $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$ and $\{\mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6\}$. In (a) $\mathcal{S}(A) > \mathcal{S}(B)$, while in (b) $\mathcal{S}(A) < \mathcal{S}(B)$.
The importance of selecting an undisputed reference point is illustrated. For WDN optimization, this is tackled by calculating the worst possible value per objective, given a certain test problem. Combining these values provides us with the Boundary point of the test problem, which can be used in normalizing the objective space and this way effectively serves as reference point. Figure courtesy of [58].

of the network as 0:

$$\mathbf{y}^{BoundTP} = \begin{bmatrix} F_{IC}\left(\mathbf{D} = \left(\underset{D_i}{\arg\max}\,[price(D_i)]\right)_{i=m^{exist}+1}^{m^{exist}+m^{new}}\right) \\ F_{wDSR}\left(\mathbf{Q}^{met} = (0)^n\right) = 0 \\ F_{SE}\,(S = 0) = 1 \end{bmatrix} \tag{2.5}$$

where:

$$\mathbf{D} = \left(D_{m^{exist}+1}, \ldots, D_{m^{exist}+m^{new}}\right) \tag{2.6}$$

$$\mathbf{Q}^{met} = \left(Q_{r+1}^{met}, \ldots, Q_{r+n}^{met}\right) \tag{2.7}$$

Furthermore, the Ideal point of a test problem $\mathbf{y}^{IdealTP}$ is calculated by selecting the least expensive pipe diameter for all new pipes in the network (i.e., no new pipes are added in case of a parallel expansion problem), setting the water demand met equal to the requested demand for all internal nodes, and taking the maximum entropy of the network by assuming the flows in all pipes equal to each other:

$$\mathbf{y}^{IdealTP} = \begin{bmatrix} F_{IC}\left(\mathbf{D} = \left(\arg\min_{D_i}\left[price(D_i)\right]\right)_{i=m^{exist}+1}^{m^{exist}+m^{new}}\right) \\ F_{wDSR}\left(\mathbf{Q}^{met} = \mathbf{Q}\right) = 1 \\ F_{SE}\left(\mathbf{q} = (1)^m\right) \end{bmatrix} \tag{2.8}$$

where:

$$\mathbf{q} = (q_1, \ldots, q_m) \tag{2.9}$$

$$\mathbf{Q} = (Q_{r+1}, \ldots, Q_{r+n}) \tag{2.10}$$

The objective space is then normalized to $[0,1]^{n_f}$ by

$$\frac{y_i - y_i^{IdealTP}}{y_i^{BoundTP} - y_i^{IdealTP}}, \forall i \in \{1, \ldots, n_f\} \tag{2.11}$$

For WDN optimization this projects the objective space onto $[0,1]^3$ and sets all objectives to be minimized. The following normalized reference point $\mathbf{y}^{ref}$ can then be used:

$$y_i^{ref} = 1 + \varepsilon, \forall i \in \{1, 2, 3\} \tag{2.12}$$

Here ε is an arbitrary (small) positive value to make sure that all solutions in an examined non-dominated set dominate the reference point, and hence contribute to the $\mathcal{S}$ metric score of the set. In this study $\varepsilon = 10^{-7}$ is used.

2.2 Robust Optimization

The *robustness* of a solution is the property of being insensitive to slight changes in the environment, or noise in the decision variables [16]. We here focus on changes in the environment, as the main factors involved in the robustness of a WDN design lie therein. In many real world problems adaptation to a changing environment is not possible, e.g., because the environment changes too quickly, because the environment cannot be monitored closely enough, or because the changes occur after the commitment to a particular solution has been made [16]. The last is the case with WDN optimization. Therefore one might want to take precautions and search for solutions that perform well under all possible future scenarios that can be expected [16]. Due to the uncertainty, the fitness function in those cases is stochastic. Let $f(\mathbf{x}, \mathbf{a})$ be the deterministic fitness function determining the quality of solution $\mathbf{x}$ in an environment $\mathbf{a}$. Its stochastic counterpart, under disturbed environment, then is $f(\mathbf{x}, \boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is a vector of noisy environmental parameters. $\boldsymbol{\alpha}$ is sampled from the problem-specific distribution of the variation in the environmental parameters $\mathbf{a}$, i.e., $\boldsymbol{\alpha} \sim \mathrm{pdf}(\mathbf{a})$ [60].

In the context of robustness, a distinction is made between three fitness concepts [16]:

- *raw fitness* f_{raw}: the objective function value as calculated by the original deterministic objective function, i.e., $f_{raw}(\mathbf{x}) = f(\mathbf{x}) \equiv f(\mathbf{x}, \mathbf{a})$;
- *effective fitness* f_{eff}: the robust objective function value, taken as the *expected fitness* accounting for fluctuating environmental parameters, i.e., [60]:

  $$f_{eff}(\mathbf{x}) = \int_{-\infty}^{\infty} f(\mathbf{x}, \boldsymbol{\alpha}) d\boldsymbol{\alpha} \tag{2.13}$$

  However, for most problems it is impossible to find closed form expressions for f_{eff} (e.g., when given in the form of a simulation model [16], as is the case with WDN optimization), and it is commonly approximated using *Monte Carlo integration* [61]:

  $$f_{eff}(\mathbf{x}) = \frac{1}{N^{sample}} \sum_{i=1}^{N^{sample}} f(\mathbf{x}, \boldsymbol{\alpha}_i) \tag{2.14}$$

  $$\boldsymbol{\alpha}_i \sim \mathrm{pdf}(\mathbf{a}) \tag{2.15}$$

  where:

  N^{sample} number of samples used for robustness evaluation, 2000 in this study;
  $\mathbf{x}$ vector of decision variables;
  $\boldsymbol{\alpha}_i$ vector of noisy environmental parameters, i.e., sample i;
  $\mathrm{pdf}(\mathbf{a})$ distribution of the variation in the environmental parameters $\mathbf{a}$.

- *estimated fitness* f_{est}: estimation of the robust objective function value, an approximation of the effective fitness that is used during optimization by the *robust evaluation method*, as calculating the effective fitness is too costly in terms of computation time. The way in which the estimated fitness is calculated depends on the robust evaluation method used.

The concepts given above apply to the single objective optimization model, in which a single objective function f is used for expressing solution fitness, hence the aim of robust optimization is, considering a minimization problem [25]:

$$\left.\begin{array}{rc} \textit{Minimize} & f_{\mathit{eff}}(\mathbf{x}) \\ \textit{subject to} & \mathbf{x} \in \mathbb{S} \end{array}\right\} \tag{2.16}$$

They can readily be generalized to the multiobjective optimization model, with objective functions f_i with $i \in \{1, \ldots, n_f\}$ expressing solution quality. The goal of *multiobjective robust optimization* is, considering an all minimization problem [25]:

$$\left.\begin{array}{rc} \textit{Minimize} & \mathbf{f}_{\mathit{eff}}(\mathbf{x}) = (f_1^{\mathit{eff}}(\mathbf{x}), \ldots, f_{n_f}^{\mathit{eff}}(\mathbf{x})) \\ \textit{subject to} & \mathbf{x} \in \mathbb{S} \end{array}\right\} \tag{2.17}$$

2.2.1 Robust Evaluation Methods

Two straightforward robust evaluation schemes that can be applied within the (multiobjective) optimizer to estimate the effective fitness of candidate solutions, instead of optimizing over their raw fitness, are the Single Evaluation Model (SEM) and Multi Evaluation Model (MEM) [61]. In SEM the estimated fitness is calculated as a single randomly perturbed evaluation:

$$f_{est}(\mathbf{x}) = f(\mathbf{x}, \boldsymbol{\alpha}) \tag{2.18}$$

Its MEM counterpart applies Monte Carlo integration, like in the approximation of the effective fitness, but using at least 2 samples:

$$f_{est}(\mathbf{x}) = \frac{1}{N^{sample}} \sum_{i=1}^{N^{sample}} f(\mathbf{x}, \boldsymbol{\alpha}_i) \tag{2.19}$$

In this study, $N^{sample} = 10$ is used with MEM.

With regard to MEM, in [16] it was found that de-randomizing the way different samples within one sample set are generated is beneficial, using *Latin Hypercube sampling* instead of *Monte Carlo sampling* (i.e., pure random sampling governed by the probability density function). Furthermore, when using a *population-based* (i.e., multiset) algorithm, it is beneficial to use the same sample set in evaluating all the candidate solutions per *generation* (i.e., round or iteration of the algorithm loop). Both approaches are used with MEM in this study.

Concerning SEM, which evaluates candidate solutions using a single random sample of (in the context of WDN optimization) environmental parameters, the following can be stated [12]: due to selection (i.e., comparing and discarding of candidate solutions on, in this case, estimated fitness), the robustness of a design is *not* tested with respect to the density function $\mathrm{pdf}(\mathbf{a})$. Selection is instead driven by beneficial realizations of the single sample α. Designs that are by chance paired with an advantageous sample of environmental parameters will get preferred. To prevent this effect in SEM, we will use the same single random sample of environmental parameters in evaluating all the candidate solutions per generation.

Chapter 3

Multiobjective Evolutionary Algorithms

Evolutionary Algorithms (EAs) are *population-based* (i.e., multiset) *stochastic global optimizers*. Guided by abstractions of the Darwinian principles of organic evolution (e.g., *recombination*, *mutation*, *selection*) these algorithms converge to the (supposedly) *global optima* in the *search space* of possible solutions.

In the context of WDN optimization various multiobjective EAs (MOEAs) are applied in literature, the most popular being *NSGA-II* by Deb et al. [27], *SPEA2* by Zitzler et al. [100], and *MOGA* by Fonseca and Fleming [40]. In this study NSGA-II is used, featuring multiobjective selection based on *crowding-distance* (cf. Section 3.1). *SMS-EMOA* by Emmerich et al. [30] was partly derived from NSGA-II, but is centered instead around multiobjective selection based on *dominated hypervolume contribution*. As SMS-EMOA was shown to outperform NSGA-II and SPEA2 on standard two objective and three objective benchmarks (i.e., ZDT, DTLZ) [11], it is considered in this study (cf. Section 3.2).

Next to the standard NSGA-II and SMS-EMOA, *self-adaptive* variants of these algorithms are presented (cf. Section 3.3). Following [72], the self-adaptive variation operators of the *Mixed-Integer Evolution Strategy* [63] are combined with the crowding-distance and hypervolume contribution multiobjective selection schemes. For obtaining a *steady state* self-adaptive SMS-EMOA, the approach of [72] is extended with ideas from Klinkenberg et al. [56].

3.1 NSGA-II

The *Nondominated Sorting Genetic Algorithm II* (NSGA-II) [27] is a *generational* MOEA (involving *parent* and *offspring* populations of size μ) that aims at approximating the

Pareto front for a given problem, while keeping high diversity in its result set. It builds on three main modules:

- `fast-non-dominated-sort`: partitions the population P_t in *fronts* $\mathcal{F}_i$, with index i indicating the non-domination rank shared by all individuals contained in such a front. The first front $\mathcal{F}_1$ is the actual non-dominated front, i.e., it consists of all non-dominated solutions in population P_t at a certain *generation* t. The second front $\mathcal{F}_2$ consists of all individuals that are non-dominated in the set $P_t \setminus \mathcal{F}_1$, i.e., each member of $\mathcal{F}_2$ is dominated by at least one member of $\mathcal{F}_1$. Generally, front $\mathcal{F}_k$ comprises all individuals that are non-dominated if the individuals in fronts $\mathcal{F}_j$ with $j < k$ were to be removed from P_t (cf. Fig. 3.1). [11];

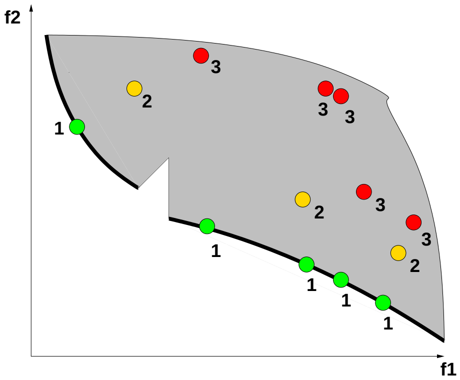


Figure 3.1: **Non-dominated Sorting**
The population of individuals is partitioned into fronts. 13 individuals are indicated in the objective space by their objective function value vectors. The numbers attached to the points indicate the front that the individuals belong to, i.e., their non-domination rank. Figure courtesy of [29].

- `crowding-distance-assignment`: calculates a *crowding-distance* value for each individual within a certain front $\mathcal{F}_i$ as the difference in objective function values between the nearest neighbors at each side of the individual, summed up over all objectives (cf. Fig. 3.2). Extremal solutions (i.e., solutions with the smallest and largest function values occurring within the front) are assigned an infinite distance value, which, motivated by the pursuit of diversity, effectively preserves them into the next generation should the front in which they are contained be partially discarded when a new population P_{t+1} is formed. [27];

- *crowded-comparison* operator ($\prec_n$): guides the selection process by defining a ordering on P_t. Each individual has two attributes, a non-domination rank and

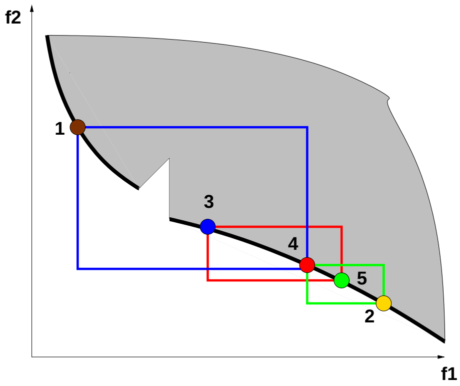


FIGURE 3.2: **Crowding-distance Sorting**
Five individuals, all belonging to the first front $\mathcal{F}_1$, are indicated in the objective space by their objective function value vectors. The crowding-distance value of an individual is determined by its nearest neighbor on each side, and individuals on the boundaries are assigned infinite values. Hence, extremal points and points whose neighbors lie far apart are favored. Figure courtesy of [29].

a crowding-distance value. Between two individuals with with differing non-domination ranks, we prefer the individual with the lower rank. Otherwise, with both individuals belonging to the same front, we prefer the individual that is located in the lesser crowded region (i.e., with higher crowding-distance value). [27]

The main loop of NSGA-II is given in Algorithm 1. For algorithmic descriptions of procedures `fast-non-dominated-sort` and `crowding-distance-assignment` the reader is referred to [27]. The exact order in which the operations are listed in Algorithm 1 differs from the main loop in [27], but this is because we follow Deb's `nsga2-v1.1` implementation[1].

The algorithm proceeds as follows: initially, a random parent population P_0 of size μ is generated. Objective functions values are assigned to the individuals in P_0 by sequentially feeding them to the project simulator (cf. Chapter 1). For the variation operators that employ the crowded-comparison operator in generating offspring individuals at the start of the main loop, it is necessary that the parent individuals in P_0 be assigned a non-domination rank and crowding-distance value.

[1] Deb's `nsga2-v1.1` C implementation can be obtained via `http://www.iitk.ac.in/kangal/codes.shtml`.

Algorithm 1 NSGA-II

```
/* Initialization */
$P_0 \leftarrow$ initialize();   // Initialize random population of $\mu$ individuals
evaluate($P_0$);   // Evaluate individuals in initial population through simulation
$\{\mathcal{F}_1, \ldots, \mathcal{F}_\nu\} \leftarrow$ fast-non-dominated-sort($P_0$);   // All $\nu$ fronts of $P_0$
for all $i \in \{1, \ldots, \nu\}$ do
    crowding-distance-assignment($\mathcal{F}_i$);   // Calc. crowd.-dist. of indiv. in $\mathcal{F}_i$
end for
$t \leftarrow 0$;
/* The evolution loop */
while not terminate do
    $Q_t \leftarrow$ make-new-pop($P_t$);   // Generate $\mu$ offspring individuals by variation
    evaluate($Q_t$);   // Evaluate individuals in $Q_t$ through simulation
    $R_t \leftarrow P_t \cup Q_t$;   // Combine parent and offspring populations in $R_t$
    $\{\mathcal{F}_1, \ldots, \mathcal{F}_\nu\} \leftarrow$ fast-non-dominated-sort($R_t$);   // All $\nu$ fronts of $R_t$
    $P_t \leftarrow \emptyset$;  $i \leftarrow 1$;
    repeat
        crowding-distance-assignment($\mathcal{F}_i$);   // Calc. crowd.-dist. of indiv. in $\mathcal{F}_i$
        $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$;   // Include individuals from the i-th front in $P_{t+1}$
        $i \leftarrow i + 1$;
    until $(|P_{t+1}| + |\mathcal{F}_i| \geq \mu)$;   // Add fronts until the size of $\{P_{t+1} \cup \mathcal{F}_i\}$ exceeds $\mu$
    sort($\mathcal{F}_i$,$\prec_n$);  // Sort $\mathcal{F}_i$ in descend. order using crowded-comparison operator
    $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i[1 : (\mu - |P_{t+1}|)]$;  // Best $\mu$ elements form the new parent pop.
    $t \leftarrow t + 1$;   // Increment the generation counter
end while
```

The termination criterion for the evolution loop is exceedance of a certain amount of generations, depending on the available number of *evaluations* (cf. Chapter 4). Binary tournament parent selection based on $\prec_n$ and problem-specific variation operators (cf. Section 3.1.1) are used to create a child population Q_t of size μ. After generating the offspring population, it gets combined with the parent population and this 2μ set R_t is partitioned into fronts according to non-domination rank. Following the new parent population P_{t+1} is to be formed. Front-wise, the best μ individuals in R_t proceed to the next generation, a procedure in which the last front added is (potentially) partly discarded after sorting its members on crowding-distance value (cf. Fig. 3.3). To this end `nsga2-v1.1` includes an instance of the 1962 `Quicksort` algorithm by Hoare [50], which in our case has a time complexity of $\mathcal{O}(\mu \log \mu)$. Then the generation counter is incremented, and a new iteration starts, provided the termination criterion is not satisfied. The overall worst-case time complexity of NSGA-II is governed by `fast-non-dominated-sort`, being $\mathcal{O}(\mu^2)$ for the three-dimensional WDN optimization problem [27].

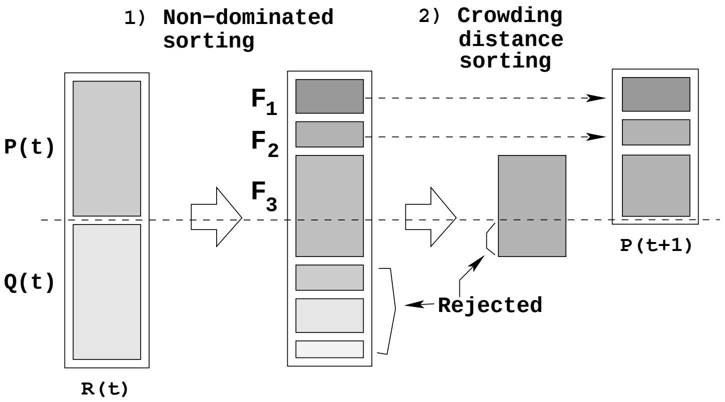


FIGURE 3.3: **NSGA-II Explained**
In step 1 we see the parent population P_t and offspring population Q_t, combined in R_t, going through non-dominated sorting. In generating the new parent population P_{t+1} (step 2), individuals are processed front-wise, assigning them crowding-distance values and moving them to the new population. Enforcing the population size of μ individuals, part of the individuals in the last front selected is discarded, governed by crowding-distance value. Figure courtesy of [26].

3.1.1 Variation Operators for WDN Optimization

Variation for the discrete WDN optimization problem can be approached in two ways. The decision variables are the diameter codes selected per pipe, pointing to a certain size/price/roughness triple in the table of available commercial diameters (cf. Chapter 1). In principle, as these diameter triples are listed in increasing size and price order (cf. Chapter 4), we could interpret the diameter codes as integers and exploit this ordering to guide the variation process. However, as typically only a very limited set of diameters is available per test problem (e.g., 6–15 diameters, cf. Chapter 4), we find that this approach is not feasible. Instead the diameter codes are regarded as nominal values, and special nominal variation operators have been implemented for use with

NSGA-II: nominal two-point crossover (i.e., the recombination operator) and a nominal mutation procedure, which per nominal vector component decides whether to change it to a uniform randomly selected new value.

3.1.2 Overview of Parameters

The configuration parameters that need to be set for NSGA-II are, when applied to WDN optimization:

- parent population size μ: in this study $\mu = 100$ is used;
- crossover probability: in this study 0.8 is used, which determines that two-point nominal crossover is applied for 80% of the generated offspring, and the remaining 20% consists of direct copies of parent individuals.

The configuration parameters for NSGA-II for which fixed recommended values exist are, when applied to WDN optimization:

- mutation rate: $\frac{1}{n_d}$ [27], where n_d is the number of discrete nominal decision variables (i.e., the diameters D_i to be selected for new pipes, hence $n_d = m^{new}$ (cf. Chapter 1)), which means that per individual one decision variable is mutated on average.

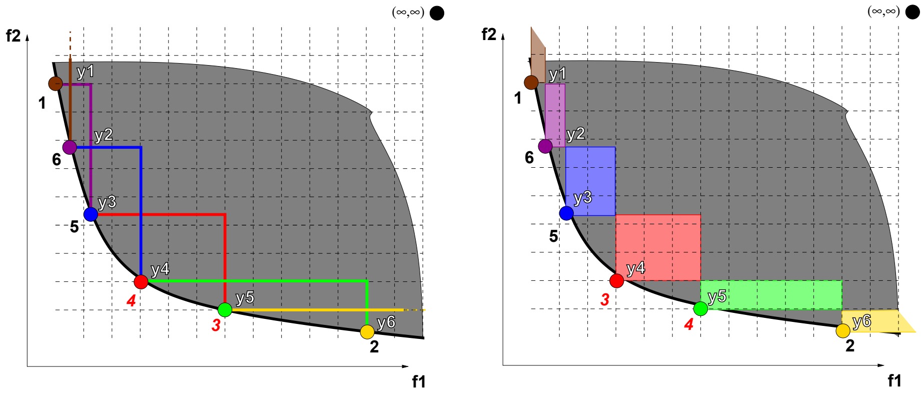
3.2 SMS-EMOA

The $\mathcal{S}$ *Metric Selection Evolutionary Multiobjective Optimization Algorithm* [11, 30] is a *steady-state* MOEA (generating only one offspring individual per generation) and was designed to cover a maximal *hypervolume* (also referred to as the $\mathcal{S}$ *metric*) with a limited number of points. Extending on a combination of ideas borrowed from NSGA-II and archiving strategies used in [59], it is founded on two pillars:

- *non-dominated sorting*: is used as a ranking criterion for the parent and offspring individuals at a certain generation t (cf. Section 3.1);

- *contributing hypervolume*: is applied as a selection criterion to discard the individual from the combined parent and offspring pool $P_t \cup \{q_t\}$ that contributes the least hypervolume to the worst-ranked front $\mathcal{F}_\nu$.

Using selection based on contributing hypervolume instead of crowding-distance is motivated among others by the observation that this makes that SMS-EMOA concentrates on good compromise solutions with fair trade-offs, which is beneficial from a practical point of view (cf. Figure 3.4) [30]. As opposed to NSGA-II certain kinds of divergence, away from a non-dominated set of solutions to solutions that are strictly worse, cannot occur within SMS-EMOA [31]. Also, the performance indicator that is used to measure the quality of the result sets of the multiobjective optimizers (i.e., the $\mathcal{S}$ metric) is now directly integrated in the optimizer algorithm itself.



(a) Crowding-distance

(b) Hypervolume Contribution

FIGURE 3.4: **Selection Measures Compared**
Six individuals, all belonging to front $\mathcal{F}_1$, are indicated in the objective space by their objective function value vectors $\mathbf{y}_i$. Using crowding-distance, individual 5 would be preferred over individual 4. On the other hand, $\Delta_{\mathcal{S}}(4, \mathcal{F}_1) > \Delta_{\mathcal{S}}(5, \mathcal{F}_1)$. This indicates that good compromise solutions, which are located near knee-points of convex parts of the Pareto front, are given better ranks in SMS-EMOA than in NSGA-II. Figure courtesy of [30].

The main loop of SMS-EMOA is given in Algorithm 2. We start from an initial population P_0 of μ individuals. Evaluation, termination, and variation proceed as in NSGA-II (cf. Section 3.1), but using binary tournament parent selection based on non-domination rank only. Should competing individuals share the same rank, a random choice between them is made. Furthermore, variation generates only one offspring individual. The `reduce` procedure, given in Algorithm 3, then discards the worst individual from the combined population $P_t \cup \{q_t\}$. This is the individual s in the last front $\mathcal{F}_\nu$ that minimizes $\Delta_{\mathcal{S}}(s, \mathcal{F}_\nu)$, which is termed the *exclusive contribution* of s to the hypervolume of $\mathcal{F}_\nu$ [11]:

$$\Delta_{\mathcal{S}}(s, \mathcal{F}_\nu) := \mathcal{S}(\mathcal{F}_\nu) - \mathcal{S}(\mathcal{F}_\nu \setminus \{s\}) \tag{3.1}$$

The remaining individuals form the new parent population P_{t+1}, and a new iteration starts.

Algorithm 2 SMS-EMOA

```
$P_0 \leftarrow$ initialize();                 // Initialize random population of $\mu$ individuals
evaluate($P_0$);      // Evaluate individuals in initial population through simulation
$t \leftarrow 0$;
while not terminate do
    $q_t \leftarrow$ generate($P_t$);           // Generate one offspring individual by variation
    evaluate($q_t$);                        // Evaluate individual $q_t$ through simulation
    $P_{t+1} \leftarrow$ reduce($P_t \cup \{q_t\}$);                  // Select the $\mu$ best individuals
    $t \leftarrow t + 1$;
end while
```

Directly using the $\mathcal{S}$ metric, which was originally intended as a quality measure for comparing the results of multiobjective optimizers [101], in a selection criterion holds drawbacks concerning running time. A new algorithm has been devised, however, that computes the hypervolume of a three-dimensional set in $\mathcal{O}(\mu \log \mu)$ [10]. Through incorporation of this algorithm (via the `hv-1.3` implementation[2] by Fonseca et al. [42]) the worst-case time complexity of SMS-EMOA becomes $\mathcal{O}(\mu^2 \log \mu)$ for a three-dimensional problem. This is still greater than the $\mathcal{O}(\mu^2)$ of NSGA-II. Moreover, the evolution loop continuing for more generations due to the steady state model contributes significantly to a larger actual running time of SMS-EMOA compared to that of NSGA-II (cf. Chapter 4).

[2] The `hv-1.3` C implementation can be obtained via `http://iridia.ulb.ac.be/~manuel/hypervolume`

Algorithm 3 `reduce(`Q`)`

```
$(\{\mathcal{F}_1, \dots, \mathcal{F}_\nu\} \leftarrow$ fast-non-dominated-sort($Q$);        // All $\nu$ fronts of $Q$
$r \leftarrow \arg\min_{s \in \mathcal{F}_\nu}[\Delta_\mathcal{S}(s, \mathcal{F}_\nu)]$;        // Select $s \in \mathcal{F}_\nu$ with lowest $\Delta_\mathcal{S}(s, \mathcal{F}_\nu)$
return $(Q \setminus \{r\})$;        // Eliminate element $r$
```

3.2.1 SMS-EMOA-dp

In an attempt to improve the running time of the basic SMS-EMOA, a new variant was contrived that applies the *number of dominating points* $d(s, Q)$ as primary selection measure and hence is termed *SMS-EMOA-dp* [11]. The alternative selection method is implemented through a modified `reduce-dp` procedure, which is given in Algorithm 4.

Algorithm 4 `reduce-dp(`Q`)`

```
$(\mathcal{F}_1, \dots, \mathcal{F}_\nu) \leftarrow$ fast-non-dominated-sort($Q$);  // All $\nu$ non-domin. fronts of $Q$
if $\nu > 1$ then  /* If $Q$ contains dominated individuals (i.e., in latter front(s)): */
    $r \leftarrow \arg\max_{s \in \mathcal{F}_\nu}[d(s, Q)]$;        // Select $s \in \mathcal{F}_\nu$ with highest $d(s, Q)$
else   /* If $Q$ contains only non-dominated individuals (i.e., all in one front): */
    $r \leftarrow \arg\min_{s \in \mathcal{F}_\nu}[\Delta_\mathcal{S}(s, \mathcal{F}_\nu)]$;        // Select $s \in \mathcal{F}_1$ with lowest $\Delta_\mathcal{S}(s, \mathcal{F}_1)$
end if
return $(Q \setminus \{r\})$;        // Eliminate element $r$
```

The value $d(s, Q)$ stands for the total amount of individuals in the set Q that dominate individual s. The $d(s, Q)$ measure can therefore only be used to discard dominated individuals. In case $P_t \cup \{q_t\}$ (represented by Q in `reduce-dp`) contains only non-dominated individuals (with all individuals belonging to the first front $\mathcal{F}_1$), the standard $\Delta_\mathcal{S}(s, \mathcal{F}_\nu)$ selection is applied. Next to the smaller runtime complexity, the idea behind applying the $d(s, Q)$ measure is to favor individuals located in those areas where the better fronts are sparsely populated, potentially giving rise to new individuals in later generations located exactly in those vacancies *and* on the better fronts (cf. Figure 3.5) [11].

Determining the $d(s, Q)$ values requires comparing the individuals in the last front $\mathcal{F}_\nu$ with all individuals in the dominating fronts. However, `fast-non-dominated-sort`

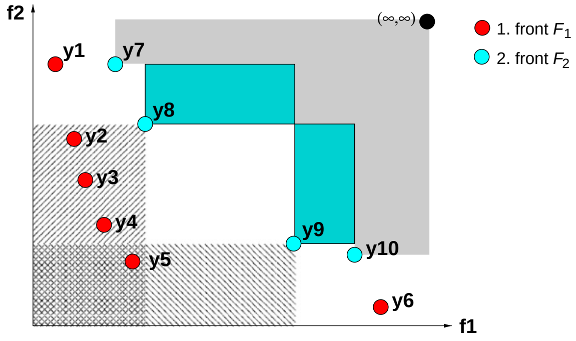


Figure 3.5: **Dominating Points Measure**
Ten individuals are indicated in the objective space by their objective function value vectors $\mathbf{y}_i$. The hypervolume dominated by front $\mathcal{F}_2 = \{7, \ldots, 10\}$ is displayed light-colored and the hypervolume contributions of individuals 8 and 9, $\Delta_{\mathcal{S}}(8, \mathcal{F}_2)$ and $\Delta_{\mathcal{S}}(9, \mathcal{F}_2)$, are visualized by dark rectangles attached to their objective function value vectors. The hatched areas correspond to the regions potentially containing points that dominate individual 8 or 9. Using dominating points selection, individual 9 would get selected over individual 8, as $d(9, P_t \cup \{q_t\}) = 1 < d(8, P_t \cup \{q_t\}) = 4$. Hypervolume contribution selection will however prefer individual 8, because $\Delta_{\mathcal{S}}(8, \mathcal{F}_2) > \Delta_{\mathcal{S}}(9, \mathcal{F}_2)$. Figure courtesy of [11].

already keeps count per individual of the number of individuals dominating it when comparing all individuals amongst each other [27], so these values can be reused. The worst-case time complexity of SMS-EMOA-dp equals that of the basic SMS-EMOA (as the standard $\Delta_{\mathcal{S}}(s, \mathcal{F}_\nu)$ selection is used if the entire population lies on one front), but the average case complexity is assumed to be better [11].

3.2.2 Choice of Reference Point

A point for attention that applies to the $\mathcal{S}$ metric calculation is the choice of *reference point* $\mathbf{y}^{ref}$. In comparing multiple sets of points, the choice of reference point can influence the relative ordering of the sets (cf. Chapter 2). For hypervolume contribution-based selection however, this necessarily careful choice can be omitted by using a *dynamic* reference point. As here points are compared that all belong to *one* set, namely the worst ranked front $\mathcal{F}_\nu$ of the combined population $P_t \cup \{q_t\}$, per generation the dynamic reference point can be taken equal to the *Nadir point* of $\mathcal{F}_\nu$: the vector consisting of the worst objective function values in $\mathcal{F}_\nu$.

A consequence of using the Nadir point is that the extremal points in $\mathcal{F}_\nu$ (i.e., the points containing a worst objective function value) do not contribute to the hypervolume at all as the product of their difference to the reference point equals zero. In [11] it is suggested to use the worst objective function values increased by 1.0 (assuming a minimization problem). This makes that for the hypervolume contribution, extremal points depend solely on their remaining non-extreme objective function values, as the difference of the extreme objective function values to the reference point equals 1.0.

Since it is straightforward to obtain the undisputed global reference point for a WDN test problem, in this study we do not employ the dynamic scheme but use a fixed reference point in determining the hypervolume contributions. Hypervolume calculation proceeds as described in Chapter 2, using the Boundary and Ideal points of the test problem (i.e., vectors comprising, respectively, the worst and best objective function values possible for the test problem).

3.2.3 Overview of Parameters

In order to assess the effect of the selection scheme in a controlled way, the settings of the configuration parameters for SMS-EMOA and SMS-EMOA-dp, when applied to WDN optimization, are chosen the same as for NSGA-II (i.e., to be set: $\mu = 100$, crossover probability of 0.8; fixed: mutation rate of $\frac{1}{n_d}$ (cf. Section 3.1.2)).

3.3 Mixed-Integer Evolution Strategy

Compared to other types of EAs, the *Evolution Strategy* (ES) stands out by using *endogenous mutation* parameters included in each individual. Instead of applying exogenous mutation control with a fixed mutation rate, globally for all individuals, this allows the individuals in the population of an ES to *self-adapt* to their optimal setting(s) [7, 79]. An ES typically operates on real-valued individuals, i.e., candidate solutions consisting of real-valued input parameters for the simulator or mathematical function under investigation. The *Mixed-Integer Evolution Strategy* (MIES) variant is specialized for working with solutions comprising real-valued, integer, and/or discrete nominal variables [63, 72].

In this study we use MIES to optimize the discrete WDN optimization problem. Like with NSGA-II and SMS-EMOA, instead of interpreting the decision variables as integer values, we process them as nominal values (cf. Section 3.1.1). This is motivated by the limited number of commercial diameters available per test problem. Therefore only the discrete nominal part of MIES is discussed here. For the complete listing, the reader is referred to [63, 72].

In order to use MIES for WDN optimization it is extended with multiobjective selection borrowed from NSGA-II and SMS-EMOA, leading to self-adaptive variants of these algorithms: *NSGA-II-sa* and *SMS-EMOA-sa*. After introducing the shared MIES operators and evolution loop in Sections 3.3.1 through 3.3.5, the specific details of NSGA-II-sa and SMS-EMOA-sa are addressed in Section 3.3.6 and 3.3.7.

3.3.1 Representation

Individuals in MIES are tuples $\vec{a}$ of the following form:

$$\vec{a} = (\ldots, \vec{d}, \ldots, \vec{p}) \tag{3.2}$$

where:

$\vec{d} = (d_1, \ldots, d_{n_d})$ n_d discrete nominal decision variables;

$\vec{p} = (p_1, \ldots, p_{n_p})$ n_p mutation probabilities for the discrete nominal decision variables.

The probabilities $\vec{p}$ are the *strategy parameters* of an individual and are used in the mutation of its discrete nominal decision variables (cf. Section 3.3.4).

3.3.2 Initialization

For the μ initial individuals in population P_0, the decision variable values and strategy parameter values are obtained as follows:

$$\vec{d} = (d_i \in [l_{d_i}, u_{d_i}])_{i=1}^{n_d} \tag{3.3}$$

$$\vec{p} = \left(p_i = U\left(\frac{1}{n_d}, 0.5\right)\right)_{i=1}^{n_p} \tag{3.4}$$

where:

$U(a, b)$ random number sampled from a continuous uniform distribution with lower bound a and upper bound b;

l_{d_i} lower bound of the domain of decision variable d_i;

u_{d_i} upper bound of the domain of decision variable d_i.

The decision variables in $\vec{d}$ are initialized uniform randomly to values in their allowed domains. The upper bound of 0.5 for the probabilities in $\vec{p}$ is motivated by the observation that mutation loses its causality once its application probability exceeds ca. 50% [63]. A lower bound of $\frac{1}{n_d}$ is selected, which assures a minimum of one discrete mutation in every application of the mutation operator. Furthermore, a setting of $n_p = 1$ is recommended. When using a single mutation probability, for each position in the discrete subvector $\vec{d}$ it is decided independently whether to mutate it, but with equal probability for all positions. Evidence that discrete nominal self-adaptation with individual stepsizes for each nominal decision variable (i.e., $n_p = n_d$) does not work properly was provided by Schütz [78].

3.3.3 Recombination

For each to be created offspring, ρ individuals are randomly picked from the current population P_t. Recombination is then applied to the decision variables and probabilities, with the choice between two approaches: *discrete recombination* or *intermediate recombination*. The first approach means a random parent from the ρ individuals is chosen per variable or strategy parameter, in the second approach the ρ values are averaged. The procedure is repeated λ times for generating the same number of offspring. Discrete recombination is recommended for the decision variables, whilst intermediate recombination is for the strategy parameters [7].

3.3.4 Mutation

The mutation operator gives MIES (and Evolution Strategies in general) the self-adaptive properties (together with selection, i.e., λ ¿ 1 is required [56]). The endogenous strategy parameters are prone to mutation, next to the decision variables, and hence to the evolutionary process. The strategy parameters determine the distance that the mutated decision variables will lie from the original variables in the search space. The essential difference between the standard ES and MIES is the extended mutation operator of the latter.

The mutation procedure in MIES for newly generated individuals is given in Algorithm 5. First the probabilities are mutated. A new value for each decision variable is then obtained by applying an *update rule* governed by the mutated accompanying probability and the current value of the decision variable.

Algorithm 5 `mutate(`$Q_t, \dots, n_d, \dots, n_p$`)`

```
Input:  $\{\vec{a} = (\dots, \vec{d}, \dots, \vec{p}) \mid \vec{a} \in Q_t\}$        // Original individuals contained in $Q_t$
Output: $\{\vec{a}' = (\dots, \vec{d}', \dots, \vec{p}') \mid \vec{a}' \in Q_t\}$    // Mutated versions of individuals in $Q_t$

 1: /* Discrete nominal case: setup */
 2: $N_c \leftarrow N(0,1)$;                                   // Normally distributed random number
 3: $\tau \leftarrow \frac{1}{\sqrt{2 \cdot n_d}}$; $\tau' \leftarrow \frac{1}{\sqrt{2\sqrt{n_d}}}$;                 // Global / local learning rate
 4: /* Discrete nominal case: probability mutation */
 5: if $n_p = 1$ then                                        // Single probability mode
 6:   $p'_1 \leftarrow T^r_{[1/n_d, 0.5]}\left(\left(1 + \frac{1-p_1}{p_1} \cdot \exp(-\tau \cdot N_c)\right)^{-1}\right)$;
 7: else                                                     // Multiple probability mode
 8:   for all $i \in \{1, \dots, n_p\}$ do
 9:     $p'_i \leftarrow \left(1 + \frac{1-p_i}{p_i} \cdot \exp(-\tau \cdot N_c - \tau' \cdot N(0,1)\right)^{-1}$;
10:     $p'_i \leftarrow T^r_{[1/n_d, 0.5]}(p'_i)$;
11:   end for
12: end if
13: /* Discrete nominal case: decision variable mutation */
14: for all $i \in \{1, \dots, n_d\}$ do
15:   if $U(0,1) < p'_{\min(n_p, i)}$ then
16:     $d'_i \in [l_{d_i}, u_{d_i}] \setminus \{d_i\}$          // Choose a new value, uniformly distributed
17:   end if
18: end for
```

In lines 1–3 we see a constant random number N_c, global learning τ, and local learning rate τ' getting defined. $N(0,1)$ represents a random number sampled from a normal distribution with mean 0 and standard deviation 1. These variables are used in the mutation of the probabilities in lines 4–12: the global factor $\exp(\tau \cdot N_c)$ allows for an overall change of the mutability, while local factor $\exp(\tau' \cdot N(0,1))$ enables individual changes to probabilities p_i. The logistic function in line 6 and 9 ensures that increments to the probabilities are as likely as decrements. It maps values from [0,1] to [0,1], after which the probabilities are transformed to remain within $[1/n_d, 0.5]$. Then in lines 13–18 the decision variables d_i get updated to a new value using the appropriate probability, which is (repeatedly) the last p_{n_p} in case of $n_d > n_p$, and these new values are selected uniform randomly from the allowed domains.

To keep probabilities $\vec{p}\,'$ within the allowed domains, a transformation function $T^r_{[a,b]}(x)$ is applied that reflects a value back with equal likelihood for each position within the allowed interval [63], listed in Algorithm 6.

Algorithm 6 $T^r_{[a,b]}(x)$

```
Input:  $x$                              // Original value
Output: $x'$     // Value checked or transformed to be in $[a, b]$
1: $y \leftarrow (x - a)/(b - a)$;
2: if $\lfloor y \rfloor \bmod 2 = 0$ then
3:     $y' \leftarrow |y - \lfloor y \rfloor|$;
4: else
5:     $y' \leftarrow 1 - |y - \lfloor y \rfloor|$;
6: end if
7: $x' \leftarrow a + (b - a) \cdot y'$;
```

Analyzing $T^r_{[a,b]}(x)$ for an $x \in [a, b]$, we see that $y \in [0, 1]$, after which $y' = y$ (via line 3 if $x \neq b$, else via line 5) and thus $x' = x$. Hence the transformation function does not change an already feasible x. An $x \notin [a, b]$ will be reflected to a position depending on the modulus of the remainder after substraction of a and division by $b - a$, in the left or right half of the interval depending on $\lfloor y \rfloor \bmod 2$.

3.3.5 Algorithm Outline

Combining its operators, the evolution loop of MIES is given in Algorithm 7. First the initial population of size μ is generated (cf. Section 3.3.2) and evaluated by the project simulator (cf. Chapter 1), which assigns output scores to each individual. The termination criterion for the evolution loop is, like in the basic NSGA-II and SMS-EMOA, exceedance of a certain amount of generations, based on the available number of evaluations. The main loop starts with λ offspring being generated. If the number of parents involved ρ is greater than 1, per offspring individual ρ parents are randomly selected from P_t and recombined to generate it (cf. Section 3.3.3). Otherwise, if ρ equals 1, the λ generated offspring are direct copies of a *single*, randomly selected parent r. This single parent r is swapped with the individual at the first position in the parent population P_t (SMS-EMOA-sa requires $P_t[1] = r$, cf. Section 3.3.7). Thus, this leaves us with λ *distinct* individuals in case $\rho \geq 2$, or λ *equal* individuals in case $\rho = 1$. Either way, the λ offspring is subjected to self-adaptive mutation.

Algorithm 7 Mixed-Integer Evolution Strategy

```
$P_0 \leftarrow$ initialize();                 // Initialize random population of $\mu$ individuals
evaluate($P_0$);        // Evaluate individuals in initial population through simulation
$t \leftarrow 0$;
while not terminate do
    $Q_t \leftarrow$ generate($P_t$, $\rho$, $\lambda$);                  // Generate $\lambda$ offspring individuals
    $Q_t \leftarrow$ mutate($Q_t, \ldots, n_d, \ldots, n_p$);
    evaluate($Q_t$);                     // Evaluate individuals in $Q_t$ through simulation
    $P_{t+1} \leftarrow$ select($P_t$, $Q_t$); // Sel. $\mu$ ind. from $P_t$ and $Q_t$ to form new pop. $P_{t+1}$
    $t \leftarrow t + 1$;
end while
```

The `select` procedure is then where NSGA-II-sa and SMS-EMOA-sa come in. They provide the multiobjective schemes to select the best μ individuals, originating from the parent and offspring populations P_t and Q_t, in order for a new iteration to start.

3.3.6 NSGA-II-sa

Self-adaptive NSGA-II (NSGA-II-sa) is a combination of self-adaptive MIES operators and multiobjective selection based on *non-dominated sorting* and *crowding-distance value* (cf. Section 3.1). The main algorithm loop is that of MIES (given in Algoritm 7), and selection is handled by the `select-NSGA` procedure listed in Algorithm 8. The `select-NSGA` routine performs selection as in the basic NSGA-II (lines 13–22 of Algorithm 1): the combined parent and offspring pool R_t is divided into fronts based on non-domination, and the best ranked fronts deliver the individuals for the new parent population P_{t+1}.

NSGA-II-sa is similar to the $(\mu + \lambda)$-NSGA-II defined in [29, 32], but the latter is based on a canonical ES for real-valued search spaces, where MIES-based NSGA-II-sa can be applied to real-valued, pure integer, and mixed-integer problems.

Algorithm 8 `select-NSGA(`P_t`, `Q_t`)`

```
$R_t \leftarrow P_t \cup Q_t$;                        // Combine parent and offspring populations in $R_t$
$\{\mathcal{F}_1, \ldots, \mathcal{F}_\nu\} \leftarrow$ fast-non-dominated-sort($R_t$);        // All $\nu$ fronts of $R_t$
$P_t \leftarrow \emptyset$;  $i \leftarrow 1$;
repeat
    crowding-distance-assignment($\mathcal{F}_i$);      // Calc. crowd.-dist. of indiv. in $\mathcal{F}_i$
    $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$;               // Include individuals from the i-th front in $P_{t+1}$
    $i \leftarrow i + 1$;
until $(|P_{t+1}| + |\mathcal{F}_i| \geq \mu)$;      // Add fronts until the size of $\{P_{t+1} \cup \mathcal{F}_i\}$ exceeds $\mu$
sort($\mathcal{F}_i$,$\prec_n$);      // Sort $\mathcal{F}_i$ in descend. order using crowded-comparison operator
$P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i[1 : (\mu - |P_{t+1}|)]$;     // Best $\mu$ elements form the new parent pop.
return $P_{t+1}$;
```

Three variants of NSGA-II-sa were tested (cf. Chapter 4):

- NSGA-II-sa: using $\lambda = \mu$ as prescribed by Deb et al. [27];
- NSGA-II-sa-sp: using a fixed $\rho = 1$, which leads to the offspring being μ mutated copies of a *single parent* (sp) per generation;
- NSGA-II-sa-sp-λ: using a fixed $\rho = 1$, and a user-defined number of offspring λ (instead of a fixed $\lambda = \mu$).

3.3.6.1 Overview of Parameters

The configuration parameters that need to be set for the NSGA-II-sa variants are, when applied to WDN optimization:

- parent population size μ: in this study $\mu = 100$ is used;
- number of parents ρ: in this study $\rho = 2$ is used for NSGA-II-sa, and $\rho = 1$ for NSGA-II-sa-sp and NSGA-II-sa-sp-λ;
- offspring population size λ: in this study $\lambda = \mu$ is used for NSGA-II-sa and NGSA-II-sa-sp, and $\lambda = 10$ for NSGA-II-sa-sp-λ.

The configuration parameters for the NSGA-II-sa variants for which fixed recommended values exist are, when applied to WDN optimization:

- number of mutation probabilities n_p: $n_p = 1$ [63], i.e., a single probability is included in each individual (cf. Section 3.3.2);
- lower bound for mutation probabilities: $\frac{1}{n_d}$ [63] (cf. Section 3.3.2);
- upper bound for mutation probabilities: 0.5 [63] (cf. Section 3.3.2);
- recombination approach for decision variables: discrete [7] (cf. Section 3.3.3);
- recombination approach for mutation probabilities: intermediate [7] (cf. Section 3.3.3).

3.3.7 SMS-EMOA-sa

Self-adaptive SMS-EMOA (SMS-EMOA-sa) is a combination of self-adaptive MIES operators and multiobjective selection based on *hypervolume contribution* (cf. 3.2) and a new measure of *expected hypervolume*. The main algorithm loop is that of MIES (given in Algoritm 7), and selection is handled by the `select-SMS` procedure listed in Algorithm 9.

Given an offspring individual $s \in Q_t$, with all offspring being derived from the same single parent $r \in P_t$, the expected hypervolume $\widetilde{\mathcal{S}}(s, r, P_t)$ is the hypervolume dominated by the set consisting of current population P_t and offspring individual s, while excluding the single parent r from P_t:

$$\widetilde{\mathcal{S}}(s, r, P_t) := \mathcal{S}(\{P_t \setminus \{r\}\} \cup \{s\}) \tag{3.5}$$

Maximization of the expected hypervolume is used to perform a pre-screening on the λ generated offspring, returning one individual that serves as the single offspring q_t at a certain generation t, such that the steady state model of the basic SMS-EMOA is effectively preserved. Parent r is excluded from the $\widetilde{\mathcal{S}}$ calculations because the offspring are likely to be similar to it. If r were included, we would end up comparing each offspring s with r, instead of the offspring against each other.

Algorithm 9 `select-SMS(`P_t`, `Q_t`)`

```
$r \leftarrow P_t[1]$;                                  // Retrieve the single parent of the $\lambda$ offspring
/* Select offspring individual $s \in Q_t$ which leads to the highest exp. $\mathcal{S}$ metric */
$q_t \leftarrow \arg\max_{s \in Q_t}[\widetilde{\mathcal{S}}(s, r, P_t)]$;
$P_{t+1} \leftarrow$ reduce($P_t \cup \{q_t\}$);        // Select the $\mu$ best indiv. from $P_t$ and offspring $q_t$
return $P_{t+1}$;
```

This approach of using self-adaptation with a single parent, multiple offspring, and pre-screening to maintain the steady state model of SMS-EMOA was derived from [56], where it is called $(\mu + (1, \lambda))$-SMS-EMOA+sa. It may seem elaborate, but in order for self-adaptive mutation to work, it is required that more than one offspring individual be created per generation (i.e., $\lambda > 1$). Here we use pre-screening governed by $\widetilde{\mathcal{S}}(s, r, P_t)$ instead of a pre-screening filter based on *Kriging* as in [56]. Generally, expected hypervolume is more straightforward to apply than Kriging, as $\widetilde{\mathcal{S}}$ is entirely based on the hypervolume measure. Choices concerning reference point are inherited. Another difference between SMS-EMOA-sa and $(\mu + (1, \lambda))$-SMS-EMOA+sa is, like with NSGA-II-sa and $(\mu + \lambda)$-NSGA-II (cf. Section 3.3.6), that the former is based on MIES and the latter on a canonical ES.

Note that `select-SMS` requires the single parent used to generate the offspring to be at a pre-specified position in the datastructure used to implement P_t (e.g., position

1), which is taken care of in the `generate` procedure (cf. Algorithm 7). After applying $\widetilde{\mathcal{S}}(s, r, P_t)$ pre-screening, the `reduce` routine is employed. As in SMS-EMOA, we have the choice between `reduce` based solely on hypervolume contribution (cf. Algorithm 3), and `reduce-dp` combining hypervolume contribution with the dominating points measure (cf. Algorithm 4). This gives rise to two self-adaptive variants of SMS-EMOA: SMS-EMOA-sa and SMS-EMOA-sa-dp, the latter employing self-adaptation and dominating points selection.

3.3.7.1 Overview of Parameters

The configuration parameters that need to be set for SMS-EMOA-sa and SMS-EMOA-sa-dp are, when applied WDN optimization:

- parent population size μ: in this study $\mu = 100$ is used;
- offspring population size λ: in this study $\lambda = 10$ is used.

The configuration parameters for SMS-EMOA-sa and SMS-EMOA-sa-dp for which fixed recommended values exist are, when applied to WDN optimization:

- number of mutation probabilities n_p: $n_p = 1$ [63], i.e., a single probability is included in each individual (cf. Section 3.3.2);
- lower bound for mutation probabilities: $\frac{1}{n_d}$ [63] (cf. Section 3.3.2);
- upper bound for mutation probabilities: 0.5 [63] (cf. Section 3.3.2);
- recombination approach for decision variables: discrete [7] (cf. Section 3.3.3);
- recombination approach for mutation probabilities: intermediate [7] (cf. Section 3.3.3);
- (number of parents ρ: required to be fixed at $\rho = 1$, i.e., the SMS-EMOA-sa model is based on the mutation of a single parent).

Chapter 4

Experiments

A recurring set of problems is commonly used to test and compare methods tailored for WDN optimization. Test problems *Two Loop* and *Hanoi* deal with the design of a new water distribution network, while *New York City* is a parallel expansion problem. These benchmark problems feature a high level of abstraction, involving only the primary network components, and a single pattern of demands. Next to a higher number of pipes and nodes, more realistic problems include tanks, mechanical pumps, and disinfection facilities that have to be calibrated. Furthermore, the optimization is to be performed over multiple demands patterns, for instance peak-average-minimum or 24 one-hour time steps.

Only the three relatively simple problems are optimized in this work, in order to allow for a comparison between the multiobjective *raw* and *robust* model using multiple different methods applying these models. The focus of this work is primarily on the comparison of algorithms and models, and less on problem complexity. First the three test problems are defined in Section 4.1 through 4.3, followed by descriptions of other more complex problems in Section 4.4, then the experimental setup is discussed in Section 4.5, and finally the results of the multiobjective raw and multiobjective robust experiments are given and analyzed in Section 4.6 and 4.7.

4.1 Two Loop

Two Loop is a fictional network, defined in 1977 by Alperovitz and Shamir [2], and has a relatively low complexity. The problem is to determine the optimal design for a new water distribution network, i.e., there is no existing infrastructure. The Two Loop network layout consists of 8 pipes, 6 nodes, and one reservoir (cf. Figure 4.1). No pumping facilities are used, the internal nodes are fed by gravity. 14 commercial

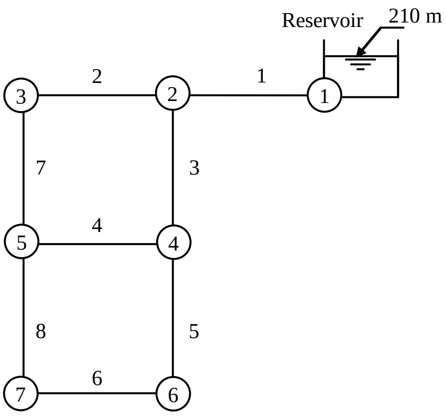


Figure 4.1: **Two Loop Network Layout**
Figure courtesy of [86].

diameters are available, sharing the same Hazen-Williams roughness coefficient (i.e., all diameters are made from the same material). The search space of Two Loop consists of $14^8 = 1.48 \times 10^9$ possible solutions (i.e., the number of available commercial diameters raised to the power of the number of pipes in the network). The maximum nodal pressure leading to zero demand satisfaction p^{zero} is taken to be 10 m [1] and the pressure required for full demand satisfaction p^{req} is 30 m. Node data is given in Table 4.1, pipe data in Table 4.2, and the commercial diameters are listed in Table 4.3. Two Loop defines diameter sizes $size(D_i)$ in inch and demands q_i in m^3/h. This combination is not available in EPANET (i.e., the hydraulic simulator used, cf. Section 1.2), so the combination of q_i in m^3/h and $size(D_i)$ in mm was used. The diameter sizes were converted from inch to mm prior to calculation (as prescribed in Section 1.2.6) using a conversion factor of 25.4.

Table 4.1: **Two Loop Node Data**

Node	**Elevation (m)**	**Demand (m^3/h)**
1	210	(reservoir)
2	150	100
3	160	100
4	155	120
5	150	270
6	165	330
7	160	200

Table 4.2: **Two Loop Pipe Data**

Pipe	**Begin Node**	**End Node**	**Length (m)**
1	1	2	1000
2	2	3	1000
3	2	4	1000
4	4	5	1000
5	4	6	1000
6	6	7	1000
7	3	5	1000
8	5	7	1000

[1] Value communicated in 2008 by Klebber Formiga, Federal University of Gois (UFG) to have been used in [43].

TABLE 4.3: **Two Loop Diameters**

Diam. Code	**Size (in.)**	**Price (m^{-1})**	**H-W coef.**
0	1	2	130
1	2	5	130
2	3	8	130
3	4	11	130
4	6	16	130
5	8	23	130
6	10	32	130
7	12	50	130
8	14	60	130
9	16	90	130
10	18	130	130
11	20	170	130
12	22	300	130
13	24	550	130

4.2 Hanoi

Test problem *Hanoi* concerns the water distribution trunk layout that was to be realized in Hanoi, Vietnam, first published in 1990 by Fujiwara and Khang [44]. Like with Two Loop, the optimal design for a new water distribution network is to be determined. The network layout consists of 34 pipes, 31 nodes, and one reservoir, organized in three loops (cf. Figure 4.2). No pumping facilities are considered. 6 commercial diameters are

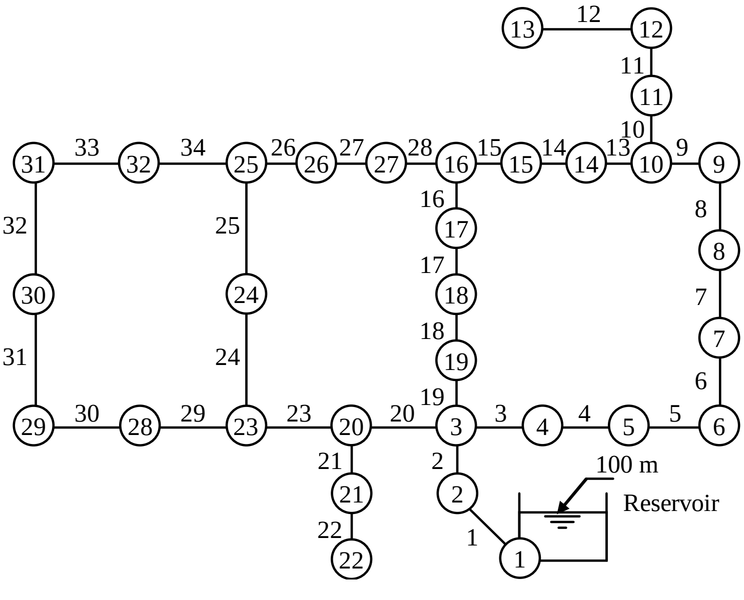


FIGURE 4.2: **Hanoi Network Layout**
Figure courtesy of [86].

Table 4.4: **Hanoi Node Data**

Node	Elevation (m)	Demand (m^3/h)
1	100	(reservoir)
2	0	890
3	0	850
4	0	130
5	0	725
6	0	1005
7	0	1350
8	0	550
9	0	525
10	0	525
11	0	500
12	0	560
13	0	940
14	0	615
15	0	280
16	0	310
17	0	865
18	0	1345
19	0	60
20	0	1275
21	0	930
22	0	485
23	0	1045
24	0	820
25	0	170
26	0	900
27	0	370
28	0	290
29	0	360
30	0	360
31	0	105
32	0	805

Table 4.5: **Hanoi Pipe Data**

Pipe	Begin Node	End Node	Length (m)
1	1	2	100
2	2	3	1350
3	3	4	900
4	4	5	1150
5	5	6	1450
6	6	7	450
7	7	8	850
8	8	9	850
9	9	10	800
10	10	11	950
11	11	12	1200
12	12	13	3500
13	10	14	800
14	14	15	500
15	15	16	550
16	17	16	2730
17	18	17	1750
18	19	18	800
19	3	19	400
20	3	20	2200
21	20	21	1500
22	21	22	500
23	20	23	2650
24	23	24	1230
25	24	25	1300
26	26	25	850
27	27	26	300
28	16	27	750
29	23	28	1500
30	28	29	2000
31	29	30	1600
32	30	31	150
33	32	31	860
34	25	32	950

available, sharing the same Hazen-Williams roughness coefficient. The search space of Hanoi consists of $6^{34} = 2.87 \times 10^{26}$ possible solutions. p^{zero} is taken to be 15 m [2] and p^{req} is 30 m. Node data is given in Table 4.4, pipe data in Table 4.5, and the commercial diameters are listed in Table 4.6. Like with Two Loop, the diameter sizes were converted from inch to mm prior to calculation using a conversion factor of 25.4.

[2] Value communicated in 2008 by Klebber Formiga, Federal University of Gois (UFG) to have been used in [43].

Table 4.6: **Hanoi Diameters**

Diam. Code	**Size (in.)**	**Price ($/m)**	**H-W coef.**
0	12	45.73	130
1	16	70.40	130
2	20	98.39	130
3	24	129.33	130
4	30	180.75	130
5	40	278.28	130

4.3 New York City

Test problem *New York City* was first examined in 1969 by Schaake and Lai [77] and is about the rehabilitation of the primary water distribution system of the city of New York. The problem is to determine the optimal design for additions to the then-existing system of water supply tunnels, reinforcing the network by constructing tunnels parallel to the existing tunnels (i.e., parallel expansion). Because of age and increased demands the existing gravity flow tunnels were found to be inadequate to meet the pressure requirements (at nodes 16, 17, 18, 19, and 20) for the projected consumption level [76]. The existing network layout consists of 21 pipes, 19 nodes, and one reservoir (cf. Figure 4.3). All existing 21 pipes are considered for duplication, using arbitrary diameters for the new pipes, to be selected from 15 commercial diameters with an equal Hazen-Williams roughness coefficient of 100. Accounting for the extra "do nothing" option, the search space of New York City consists of $16^{21} = 1.93 \times 10^{25}$ possible solutions. The existing pipes have an Hazen-Williams roughness coefficient of 100 as well[3]. p^{zero} is taken to be 50 ft, which is equal to the 15 m used for Hanoi, and p^{req} is 93 ft, equal to 40 psi as reported in [23]. Node data is given in Table 4.8, pipe data in Table 4.9, and the commercial diameters are listed in Table 4.7, taken from [22].

[3] One would expect the existing pipes to have an increased roughness caused by corrosion, which would be reflected in a lower Hazen-Williams roughness coefficient (cf. Section 1.2.5) than of the new pipes. Furthermore, certain pipes are likely to have been worn out more than others, depending on the stress exercised by the actual flow of water. The robust (or stochastic) WDN optimization model accounts for this issue (cf. Section 2.1).

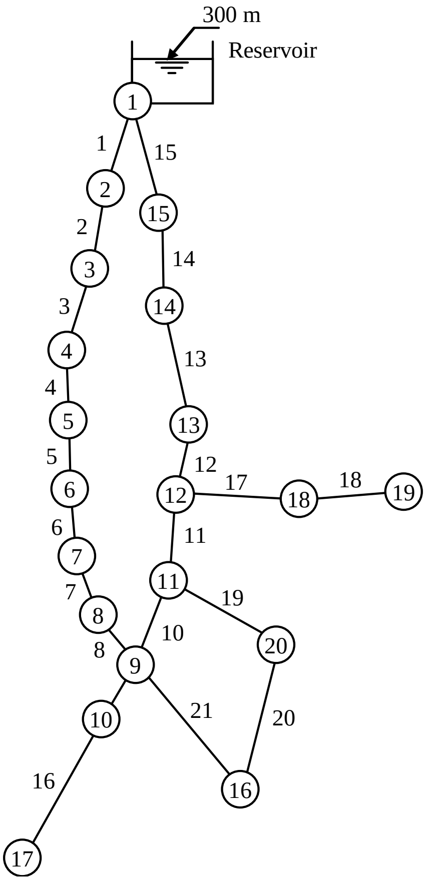


Figure 4.3: **New York City Network Layout**
Figure courtesy of [86].

Table 4.7: **New York City Diameters**

Diam. Code	**Size (in.)**	**Price ($/ft)**	**H-W coef.**
0	36	93.5	100
1	48	134	100
2	60	176	100
3	72	221	100
4	84	267	100
5	96	316	100
6	108	365	100
7	120	417	100
8	132	469	100
9	144	522	100
10	156	577	100
11	168	632	100
12	180	689	100
13	192	746	100
14	204	804	100

TABLE 4.8: **New York City Node Data**

Node	Elevation (ft)	Demand (ft^3/s)
1	300	(reservoir)
2	162	92.4
3	162	92.4
4	162	88.2
5	162	88.2
6	162	88.2
7	162	88.2
8	162	88.2
9	162	170.0
10	162	1.0
11	162	170.0
12	162	117.1
13	162	117.1
14	162	92.4
15	162	92.4
16	167	170.0
17	179.8	57.5
18	162	117.1
19	162	117.1
20	162	170.0

TABLE 4.9: **New York City Pipe Data**

Pipe	Begin Node	End Node	Length (ft)	Exist. Diam. (in.)
1	1	2	11600	180
2	2	3	19800	180
3	3	4	7300	180
4	4	5	8300	180
5	5	6	8600	180
6	6	7	19100	180
7	7	8	9600	132
8	8	9	12500	132
9	9	10	9600	180
10	11	9	11200	204
11	12	11	14500	204
12	13	12	12200	204
13	14	13	24100	204
14	15	14	21100	204
15	1	15	15500	204
16	10	17	26400	72
17	12	18	31200	72
18	18	19	24000	60
19	11	20	14400	60
20	20	16	38400	60
21	9	16	26400	72

4.4 Other Test Problems

Other test problems of which the definition is publicly available are *Apulia* [47], *Anytown* [96] and *EXNET* [34]. The first is a medium-size municipal network in Apulia (Southern Italy), gravity-fed, and with a single demand pattern. It consists of one reservoir, 23 internal nodes, 34 pipes, and 9 available diameters, making the search space of size $9^{34} = 2.78 \times 10^{32}$. The Anytown WDN was set up as a realistic benchmark and has features and problems typical of those found in many real systems, such as sizing and locating of pumps and tanks, and 5 demand patterns [36]. The EXNET is another artificial benchmark that includes real system features, such as tank sizing, operation schedules for pumps, and pipe rehabilitation decisions. It was published by the Center for Water Systems (CWS), University of Exeter (cf. Section 1.2) and inspired by the New York City and Anytown WDNs. It aims at clear definition of decision variables, objective functions, and constraints.

A recurring problem in CWS publications, but of which the definition is not publicly available, is the BW network (North of UK) [28, 51, 55]. It spans over approximately 39 kilometers. Though BW is gravity-fed and only involves sizing of pipes under a single pattern of demands, it has a large number of network components: 535 internal nodes and one reservoir, connected by 632 pipes. Together with the 20 commercial candidate pipe diameters, this gives rise to a search space of $20^{632} = 1.78 \times 10^{822}$.

4.5 Experimental Setup

The aim of this study is to compare the basic multiobjective approach (*raw*) and the multiobjective robust approach (*robust*) for WDN optimization (cf. Chapter 1). In a first stage raw experiments were performed to determine the best raw NSGA-II and SMS-EMOA variant (cf. Chapter 3). These best raw variants were then extended with the robust evaluation methods SEM and MEM (cf. Chapter 2) for running the robust experiments. By determining the hypervolume based on effective fitness ($\mathcal{S}_{\mathit{eff}}$) of the result sets obtained by the raw and robust experiments, we are able to compare the raw and robust approach for WDN optimization in terms of robustness of the found sets of solutions.

General settings used in all experiments are:

- parent population size μ of 100;
- 20,000 evaluations by the hydraulic simulator, i.e., of candidate solutions;
- 30 runs per algorithm variant, motivated by the assumption that mean values and standard deviations evaluated on, at least, 30 samples are sufficiently statistically reliable [8].

For the algorithm variants' specific configuration parameters, refer to Chapter 3.

The results of the raw experiments are compared using the hypervolume measure or $\mathcal{S}$ metric. For calculation of the $\mathcal{S}$ metric in the context of WDN optimization, refer to Chapter 2. The results of the robust experiments and best raw experiments are compared using hypervolume based on the effective fitness of the solutions in the result sets (i.e., their robust objective function values). The effective fitness of a solution is approximated using Monte Carlo integration over 2000 samples of randomly generated environmental parameters (cf. Section 2.2). In order to make a clear distinction between the hypervolume based on raw fitness and the hypervolume based on effective fitness, these are denoted by $\mathcal{S}_{raw}$ and $\mathcal{S}_{eff}$, respectively. A comparison with results of the experiments performed in this work and those found in literature is not possible, as the problem formulation Formiga (cf. Chapter 1) used in this study is not applied by other authors than Formiga et al. [43]. Furthermore, Formiga et al. [43] used an extended set of available commercial diameters for test problem Hanoi, and a different hydraulic simulator.

Next to these hypervolume values, three-dimensional *attainment surface* plots are used to compare the result sets, which represent Pareto front approximations of the problem being optimized. An *attainment curve* (in case of two objectives) or attainment surface (for three objectives) [41] is a boundary in the objective space separating those points that are dominated by at least one of the objective function value vectors in a Pareto front approximation, from those that no objective function value vector dominates or equals [57]. This boundary is “the family of tightest goals known to be attainable as a result of the optimization run” [41]. A two-dimensional example of an attainment surface is shown in Figure 4.4. The concept of an attainment surface is closely related to the hypervolume dominated by an approximation set, where the dominated hypervolume is bounded on one side by the attainment surface of the approximation set, and a reference point on the other side. However, in order to scale up the differences between the attainment surface plots, the range of each axis is determined by the extremal values occurring per test problem, taken over all result sets per problem, instead of using the reference point that lies (much) further away from the approximation sets in the objective space. For an example of an attainment surface plot of a WDN result set, see Figure 4.5. In this plot the best solution found by the constrained single objective approach of WDN optimization is indicated. For the best single objective (SO) solution per test problem we adopt the following three solutions:

- Two Loop: 420,000 units, found in 1997 by Savic and Walters [76];
- Hanoi: 6.195 million dollar, found in 1997 by Savic and Walters [76];
- New York City: 38.8 million dollar, found in 1993 by Murphy et al. [69].

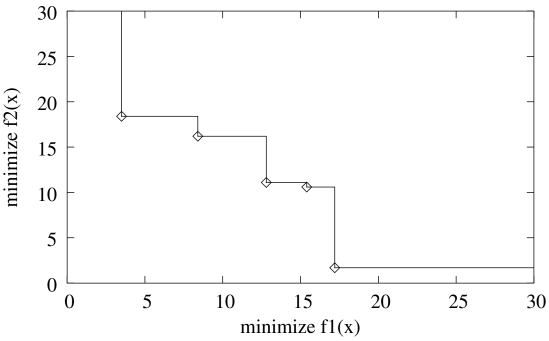


Figure 4.4: **Attainment Curve Explained**
The attainment curve of a two objective Pareto approximation set is the family of tightest goals that has been attained by the approximation set defining it. Figure courtesy of [57].

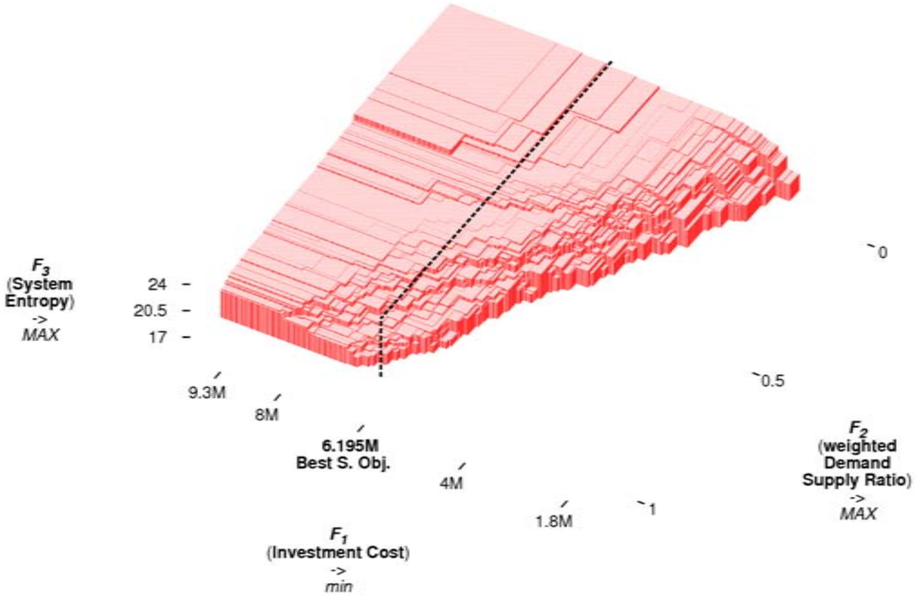
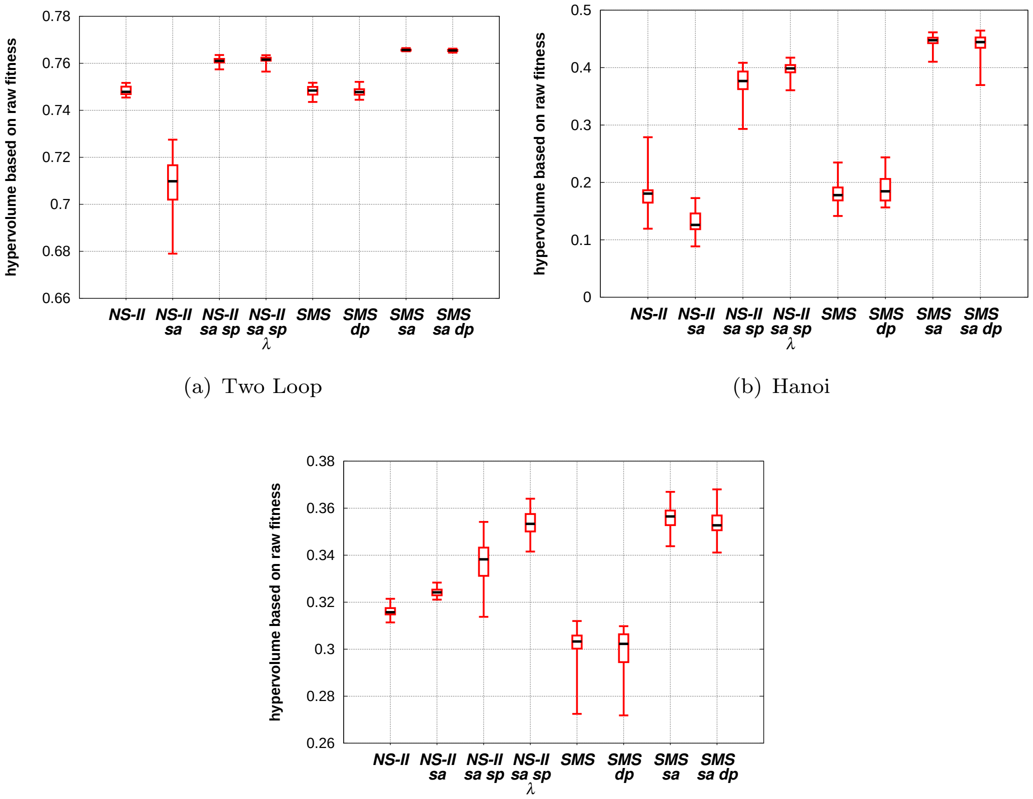


Figure 4.5: **Attainment Surface Example**
Example of an attainment surface defined by a Pareto front approximation set, for test problem Hanoi. The surface displayed belongs to the test run scoring best on raw hypervolume, which was found by method SMS-EMOA-sa-dp.
The axes are labeled by the three objective functions defined in the multiobjective problem formulation Formiga: Investment Cost (F_{IC}), Weighted Demand Supply Ratio (F_{wDSR}), and System Entropy (F_{SE}) (cf. Chapter 1). The best single objective solution known is indicated by its Investment Cost, to provide a course notion of the quality of the set and for easier comparison with the other approximation sets for Hanoi.

Though nowadays the constrained single objective approach is increasingly viewed to fall short for WDN optimization [95], work nevertheless continues on the SO optimization of the Two Loop, Hanoi, and New York City WDNs [45, 64, 66, 68, 84, 86, 87, 91]. New optimal results appear in literature, although it is sometimes not clear what value was used for the Hazen-Williams numerical conversion constant ω in obtaining these results. By leading to less calculated friction head loss, application of a lower ω value allows for finding cheaper feasible solutions. Generally the new solutions only improve slightly on Investment Cost as compared to the SO solutions listed as being optimal here, which were confirmed feasible [76] (i.e., all nodal heads satisfying the minimum pressure requirement) when evaluated with an ω value greater than or at least equal to what is used in this study (cf. Section 1.2.6).

4.6 Raw Results

The hypervolume scores of the raw experiments are given in Table 4.10, and box plots summarizing these results are displayed in Figure 4.6. Furthermore, attainment surface plots of the median, best, and worst Pareto front approximations per algorithm variant and test problem combination can be found in Appendix A.



(a) Two Loop

(b) Hanoi

(c) New York City

Figure 4.6: **Raw Results Overview**
The box plots summarize the $\mathcal{S}_{raw}$ values (i.e., hypervolume based on raw fitness) of the 30 result sets obtained per method. NS-II is short for NSGA-II, and SMS for SMS-EMOA.

In our search for the best performing NSGA-II and SMS-EMOA variant, NSGA-II-sa-sp-λ (self-adaptive NSGA-II, all offspring from a single parent per generation, and $\lambda = 10$ offspring individuals per generation) and SMS-EMOA-sa (self-adaptive SMS-EMOA) are the clear winners. For all three test problems holds that NSGA-II-sa-sp-λ outperforms the other NSGA-II variants with $> 99\%$ significance, except for NSGA-II-sa-sp on Two Loop, determined by a pairwise two-sample t-test assuming equal variances. SMS-EMOA-sa outperforms outperforms all other methods tested, including the NSGA-II

Table 4.10: **Raw Hypervolume Results**
Mean and standard deviation, and median of the $\mathcal{S}_{raw}$ values (i.e., hypervolume based on raw fitness) of the 30 runs performed per tested NSGA-II and SMS-EMOA variant are listed. Furthermore, based on the mean, the rank per algorithm type (AR) and rank per test problem (TR) are listed. The software[4] by Fonseca et al. [42] was used for calculating the hypervolume.

Test Problem	**Algorithm**	**Mean**	**Std**	**Median**	**AR**	**TR**
Two Loop	NSGA-II	0.748438	0.001922	0.747851	3	5
	NSGA-II-sa	0.708046	0.011563	0.709782	4	8
	NSGA-II-sa-sp	0.760933	0.001324	0.760995	2	4
	NSGA-II-sa-sp-λ	**0.761546**	0.001310	**0.761550**	**1**	3
	SMS-EMOA	0.748337	0.001960	0.748399	3	6
	SMS-EMOA-dp	0.747895	0.001843	0.747742	4	7
	SMS-EMOA-sa	$\boxed{\mathbf{0.765659}}$	0.000341	$\boxed{\mathbf{0.765669}}$	**1**	**1**
	SMS-EMOA-sa-dp	0.765439	0.000401	0.765457	2	2
Hanoi	NSGA-II	0.178945	0.026828	0.180552	3	7
	NSGA-II-sa	0.130188	0.021982	0.125960	4	8
	NSGA-II-sa-sp	0.373131	0.027302	0.376562	2	4
	NSGA-II-sa-sp-λ	**0.397358**	0.010801	**0.398622**	**1**	3
	SMS-EMOA	0.180601	0.023370	0.177832	4	6
	SMS-EMOA-dp	0.188012	0.024251	0.184540	3	5
	SMS-EMOA-sa	$\boxed{\mathbf{0.443710}}$	0.013909	$\boxed{\mathbf{0.447712}}$	**1**	**1**
	SMS-EMOA-sa-dp	0.441277	0.018282	0.444481	2	2
New York City	NSGA-II	0.316234	0.002415	0.315725	4	6
	NSGA-II-sa	0.324274	0.001985	0.324190	3	5
	NSGA-II-sa-sp	0.337591	0.008835	0.338253	2	4
	NSGA-II-sa-sp-λ	**0.353746**	0.006000	**0.353331**	**1**	3
	SMS-EMOA	0.301042	0.009771	0.303275	3	7
	SMS-EMOA-dp	0.299845	0.009292	0.302286	4	8
	SMS-EMOA-sa	$\boxed{\mathbf{0.356271}}$	0.005514	$\boxed{\mathbf{0.356497}}$	**1**	**1**
	SMS-EMOA-sa-dp	0.354156	0.006459	0.352706	2	2

variants, with > 99% significance, except for SMS-EMOA-sa-dp (97%, 44%, 82%) and NSGA-II-sa-sp-λ on New York City (91%). To provide insight into the scale of the differences in hypervolume, the median attainment surfaces of the worst and best performing method per test problem are displayed in Figure 4.7. Comparing the mean and median hypervolume score per test problem and method combination in Table 4.10, we see that these values lie very close to each other for all combinations. Therefore, one could argue that the median Pareto front can be seen as the expected front per combination.

Comparing the NSGA-II and SMS-EMOA variants that use the basic, non-self-adaptive variation operators (i.e., NSGA-II, SMS-EMOA, SMS-EMOA-dp, cf. Section 3.1.1), we

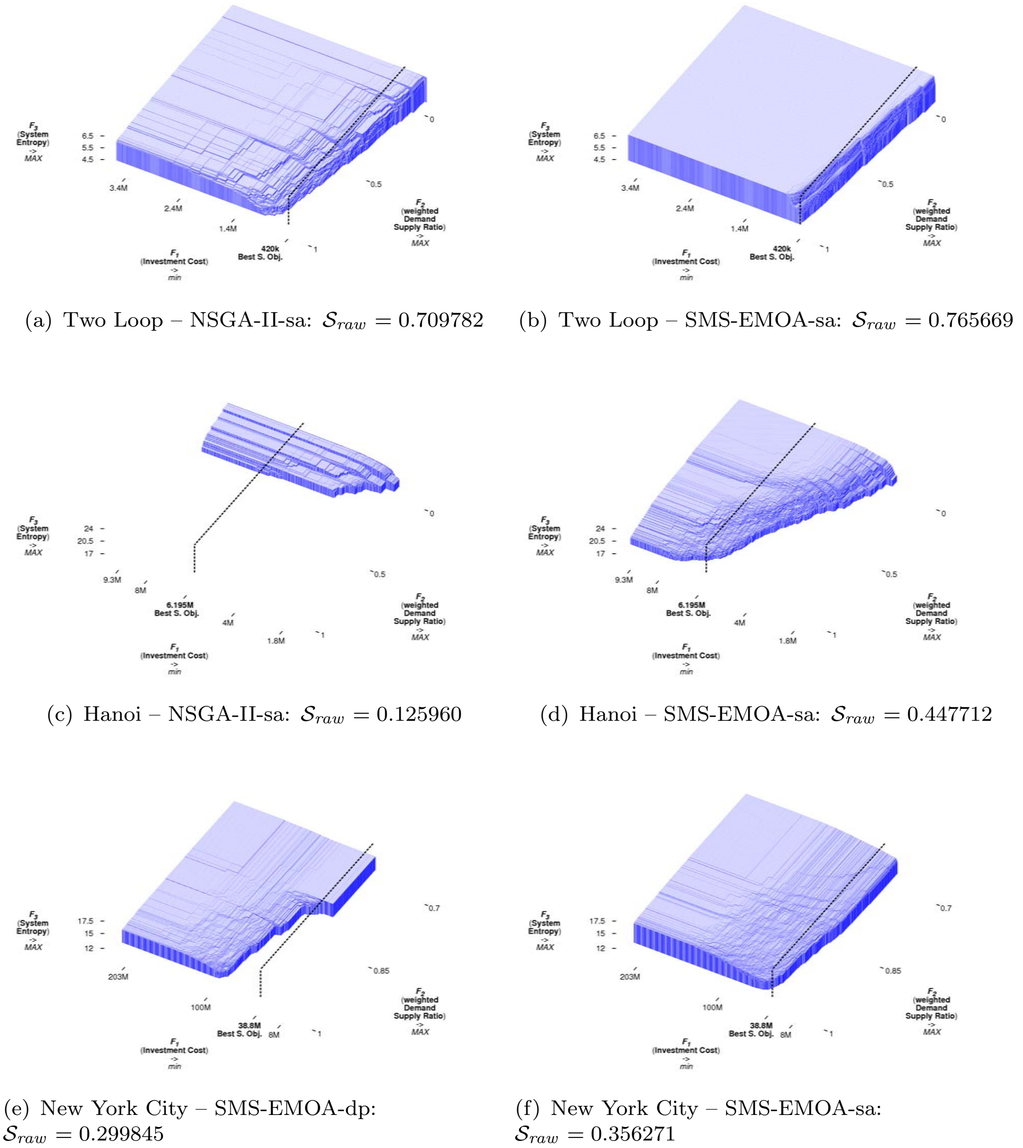
[4] `http://iridia.ulb.ac.be/~manuel/hypervolume`

see equal results for Two Loop and Hanoi, while NSGA-II performs better on New York City. These results are typical, in the sense that SMS-EMOA outperforms NSGA-II on artificial benchmarks [11], and that the SMS-EMOA-sa variants steadily outperform the NSGA-II-sa variants. Hereby note that the best NSGA-II-sa variant, NSGA-II-sa-sp-λ, employs exactly the same scheme as SMS-EMOA-sa: self-adaptation, all offspring per generation are mutated copies of a single parent individual, and λ offspring are created per generation. The only point where the methods differ is in the multiobjective selection they apply: non-dominated sorting and crowding distance ranking, versus non-dominated sorting and hypervolume-based ranking.

Furthermore, results show that using the dominating points (dp) scheme with SMS-EMOA variants generally causes a (slight) decrease in performance. However, an expected runtime improvement was the prime motivation for considering the dp scheme. The average running time per algorithm and test problem combination is given in Table 4.11. Nevertheless, using dp as primary selection measure in SMS-EMOA does not seem to speed up the algorithm either, at least in the context of WDN optimization. Comparing NSGA-II and NSGA-II-sa, it can be seen that the self-adaptive variation scheme slows down the algorithm. For SMS-EMOA and SMS-EMOA-sa holds the opposite, which is explained by the fact that the standard SMS-EMOA uses a steady-state scheme in which only one offspring evaluated per generation, while SMS-EMOA-sa generates 10 offspring per generation (of which 9 are discarded in the pre-screening phase,

Table 4.11: **Running Time of Raw Experiments**
Per test problem and algorithm combination, the mean and standard deviation of the running time of the 30 runs per combination is given in seconds. The experiments were performed on a AMD Athlon64 X2 1.80 GHz machine with 2,00 GB of memory, running Cygwin 1.7 on a 64-bit version of Windows 7, and using the gcc-4.4.2 compiler.

Algorithm	**Two Loop**		**Hanoi**		**New York City**	
	Mean(s)	**Std(s)**	**Mean(s)**	**Std(s)**	**Mean(s)**	**Std(s)**
NSGA-II	6.8	1.2	13.1	1.4	12.7	1.0
NSGA-II-sa	15.0	1.4	20.5	1.1	19.7	1.3
NSGA-II-sa-sp	16.9	3.2	19.8	1.5	19.3	1.2
NSGA-II-sa-sp-λ	91.2	10.2	97.8	9.2	93.9	7.3
SMS-EMOA	399.4	16.9	453.4	13.6	443.4	20.1
SMS-EMOA-dp	404.9	18.0	471.9	25.4	431.3	18.7
SMS-EMOA-sa	76.0	3.4	93.1	3.7	92.4	3.6
SMS-EMOA-sa-dp	76.5	6.2	82.5	6.7	76.6	1.8



(a) Two Loop – NSGA-II-sa: $\mathcal{S}_{raw} = 0.709782$

(b) Two Loop – SMS-EMOA-sa: $\mathcal{S}_{raw} = 0.765669$

(c) Hanoi – NSGA-II-sa: $\mathcal{S}_{raw} = 0.125960$

(d) Hanoi – SMS-EMOA-sa: $\mathcal{S}_{raw} = 0.447712$

(e) New York City – SMS-EMOA-dp: $\mathcal{S}_{raw} = 0.299845$

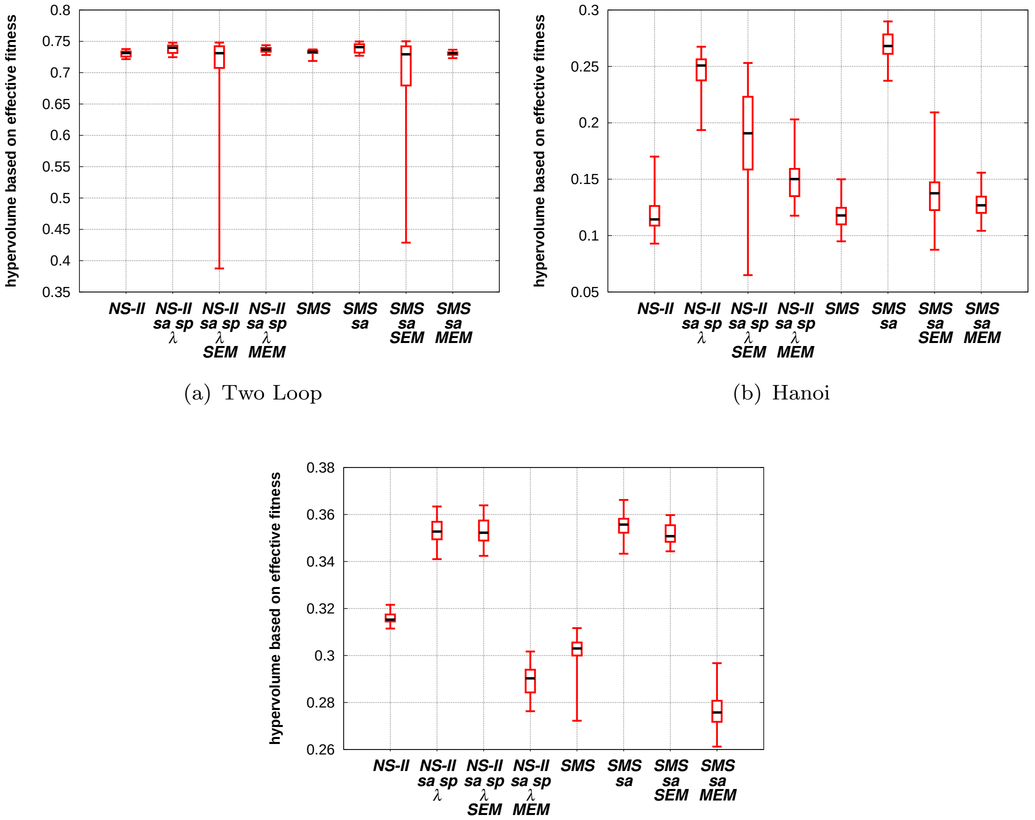
(f) New York City – SMS-EMOA-sa: $\mathcal{S}_{raw} = 0.356271$

FIGURE 4.7: **Median Raw Fronts Compared**
Per test problem, of the worst and best performing method, the attainment surface belonging to the median Pareto front approximation found is displayed. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness.

effectively preserving the steady-state scheme). This leads to a shorter evolution, involving less selection phases in which solutions need to be compared for some to be discarded. This also explains the greater average running time of NSGA-II-sa-sp-λ compared to the other NSGA-II variants: it evaluates 10 offspring individuals per generation, contrary to 100 for the other variants.

4.7 Robust Optimization Results

The hypervolume scores based on effective fitness of the robust optimization experiments are given in Table 4.12, and box plots summarizing these results are displayed in Figure 4.8. Furthermore, attainment surface plots of the median, best, and worst Pareto front approximations per algorithm variant and test problem combination can be found in Appendix B. Like with the raw results, the mean and median effective hypervolume scores per combination are very close, supporting the assumption that the median Pareto front is regarded as the expected front.



(a) Two Loop

(b) Hanoi

(c) New York City

Figure 4.8: **Robust Results Overview**
The box plots summarize the $\mathcal{S}_{\mathit{eff}}$ values of the 30 result sets obtained per method (i.e., hypervolume based on effective fitness of the solutions per result set). The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. NS-II is short for NSGA-II, and SMS for SMS-EMOA.

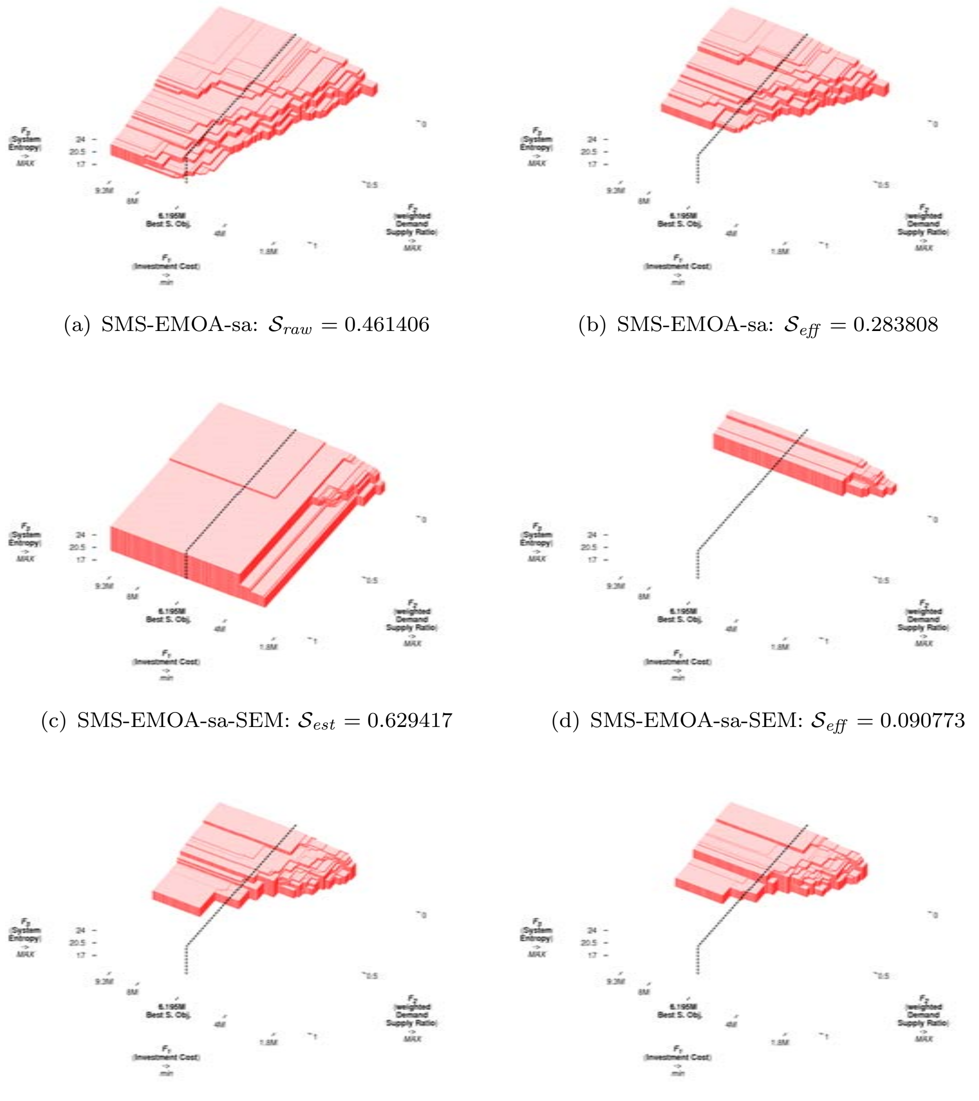
TABLE 4.12: **Robust Hypervolume Results**
Mean and standard deviation, and median of the $\mathcal{S}_{\mathit{eff}}$ values (i.e., hypervolume based on the effective fitness of the solutions in a result set) of the 30 runs performed per tested NSGA-II and SMS-EMOA variant (of which some include the robust evaluation extensions SEM and MEM) are listed. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. Furthermore, based on the mean, the rank per algorithm type (AR) and rank per test problem (TR) are listed. The software[5] by Fonseca et al. [42] was used for calculating the hypervolume.

Test Problem	Algorithm	Mean	Std	Median	AR	TR
Two Loop	NSGA-II	0.730225	0.004681	0.731287	3	6
	NSGA-II-sa-sp-λ	**0.738226**	0.007041	**0.739726**	**1**	2
	NSGA-II-sa-sp-λ-SEM	0.697023	0.089143	0.731093	4	7
	NSGA-II-sa-sp-λ-MEM	0.736560	0.003886	0.736773	2	3
	SMS-EMOA	0.732935	0.003505	0.733188	2	4
	SMS-EMOA-sa	$\boxed{\textbf{0.739309}}$	0.007167	$\boxed{\textbf{0.740761}}$	**1**	**1**
	SMS-EMOA-sa-SEM	0.689947	0.081256	0.729341	4	8
	SMS-EMOA-sa-MEM	0.730682	0.003300	0.730963	3	5
Hanoi	NSGA-II	0.118880	0.015350	0.114375	4	7
	NSGA-II-sa-sp-λ	**0.247061**	0.015868	**0.250803**	**1**	2
	NSGA-II-sa-sp-λ-SEM	0.185937	0.044410	0.190709	2	3
	NSGA-II-sa-sp-λ-MEM	0.149747	0.018552	0.150106	3	4
	SMS-EMOA	0.118010	0.012854	0.117908	4	8
	SMS-EMOA-sa	$\boxed{\textbf{0.267140}}$	0.014514	$\boxed{\textbf{0.268071}}$	**1**	**1**
	SMS-EMOA-sa-SEM	0.138262	0.031694	0.137554	2	5
	SMS-EMOA-sa-MEM	0.127311	0.010882	0.126873	3	6
New York City	NSGA-II	0.316082	0.002419	0.315273	3	5
	NSGA-II-sa-sp-λ	**0.353171**	0.005990	**0.352729**	**1**	2
	NSGA-II-sa-sp-λ-SEM	0.353027	0.005494	0.352249	2	3
	NSGA-II-sa-sp-λ-MEM	0.289590	0.006901	0.290292	4	7
	SMS-EMOA	0.300786	0.009754	0.302985	3	6
	SMS-EMOA-sa	$\boxed{\textbf{0.355450}}$	0.005530	$\boxed{\textbf{0.355717}}$	**1**	**1**
	SMS-EMOA-sa-SEM	0.352061	0.004451	0.350776	2	4
	SMS-EMOA-sa-MEM	0.277086	0.007321	0.275752	4	8

Contrarily to what would be expected, the raw methods NSGA-II-sa-sp-λ and SMS-EMOA-sa outperform their variants that explicitly include extensions for safeguarding robustness of results. For all three test problems raw SMS-EMOA-sa outperforms almost all other methods listed with $> 99\%$ significance concerning $\mathcal{S}_{\mathit{eff}}$, except for NSGA-II-sa-sp-λ on Two Loop (44%) and New York City (86%), NSGA-II-sa-sp-λ-SEM on Two Loop (98%) and New York City (90%), and NSGA-II-sa-sp-λ-MEM on Two Loop (92%).

[5] http://iridia.ulb.ac.be/~manuel/hypervolume

In an attempt to determine the cause behind these results, a comparison is made between what the optimizer regards as the fitness of its solutions, and what is the actual effective fitness of those solutions. Concerning problem Hanoi, the result set of the best raw run with respect to raw fitness, the result set of the best SEM run with respect to estimated fitness, and the result set of the best MEM run with respect to estimated fitness are analyzed in Figure 4.9. SEM seems to suffer from the strong effect of positive outliers in



(a) SMS-EMOA-sa: $\mathcal{S}_{raw} = 0.461406$

(b) SMS-EMOA-sa: $\mathcal{S}_{eff} = 0.283808$

(c) SMS-EMOA-sa-SEM: $\mathcal{S}_{est} = 0.629417$

(d) SMS-EMOA-sa-SEM: $\mathcal{S}_{eff} = 0.090773$

(e) NSGA-II-sa-sp-λ-MEM: $\mathcal{S}_{est} = 0.231888$

(f) NSGA-II-sa-sp-λ-MEM: $\mathcal{S}_{eff} = 0.203025$

Figure 4.9: **Supposedly Best Fronts Analyzed**
To visualize the difference in the fitness seen internally by the optimizer (raw fitness for raw methods, estimated fitness for SEM/MEM extended methods) and the actual effective fitness, the best runs for test problem Hanoi with respect to "internal" fitness are displayed. Hypervolume based on raw fitness is denoted by $\mathcal{S}_{raw}$, hypervolume based on estimated fitness by $\mathcal{S}_{est}$, and hypervolume based on the effective fitness by $\mathcal{S}_{eff}$. Investment Cost does not depend on the varying environmental parameters, unlike Weighted Demand Supply Ratio and System Entropy.

the estimated fitness, caused by biased generations in which all individuals get coupled with a beneficial sample of environmental parameters. These individuals are preserved in the population throughout the optimization because of the elitist MOEAs used, i.e., all but some individuals continue to a new generation. MEM on the other hand seems to fall behind because of the high evaluation cost per individual: each candidate solution is evaluated over 10 samples of environmental parameters, compared to a single fixed set of environmental parameters for the raw method. Although its implementation of estimated fitness delivers values close to the actual effective fitness, MEM requires more evaluations in order to converge than the raw method.

Note that for the more difficult test problems Hanoi and New York City, the SEM extended methods do outperform basic NSGA-II and SMS-EMOA concerning $\mathcal{S}_{\mathit{eff}}$. Initial results of experiments based on more evaluations show that the ordering between SMS-EMOA-sa and the methods including extensions for robust evaluation still holds, at least when using 100,000 evaluations. In literature however (much) larger numbers of evaluations with WDN optimization are not uncommon for problems Hanoi and New York City, using two objective optimization models (e.g., 250k [52], 1M [54], 2M [71]).

The average running time per algorithm and test problem combination is given in Table 4.13. Apparently generating a random sample of environmental parameters each generation slows the optimizer down in case of SEM. This should hold to a greater extent for MEM as well, but is nullified by the shorter evolution, caused by the factor 10 more evaluations used per generation.

Table 4.13: **Running Time of Robust Optimization Experiments**
Per test problem and algorithm combination, the mean and standard deviation of the running time of the 30 runs per combination is given in seconds. The experiments were performed on a AMD Athlon64 X2 1.80 GHz machine with 2,00 GB of memory, running Cygwin 1.7 on a 64-bit version of Windows 7, and using the gcc-4.4.2 compiler.

Algorithm	**Two Loop**		**Hanoi**		**New York City**	
	Mean(s)	**Std(s)**	**Mean(s)**	**Std(s)**	**Mean(s)**	**Std(s)**
NSGA-II	6.8	1.2	13.1	1.4	12.7	1.0
NSGA-II-sa-sp-λ	91.2	10.2	97.8	9.2	93.9	7.3
NSGA-II-sa-sp-λ-SEM	97.6	1.2	136.0	4.3	130.7	1.2
NSGA-II-sa-sp-λ-MEM	12.5	1.3	26.2	0.5	20.7	0.6
SMS-EMOA	399.4	16.9	453.4	13.6	443.4	20.1
SMS-EMOA-sa	76.0	3.4	93.1	3.7	92.4	3.6
SMS-EMOA-sa-SEM	104.8	3.5	134.6	7.9	151.5	7.9
SMS-EMOA-sa-MEM	12.4	0.7	24.0	0.6	19.8	0.4

Chapter 5

Conclusion

Optimization of water distribution network (WDN) design forms a field of study attracting a lot of attention by scientists in different fields. Originally being defined as a single objective (SO) least cost problem subject to a constraint on the minimum nodal pressure, in the past decade the aim has been towards alternative formulations of the optimization model. This is motivated by the fact that the difficulties of WDN optimization are not reflected well in the SO approach. The main issues of WDN optimization are that it is hard to define the appropriate objective functions and constraints, not a single set of demands should be used, the WDN is not built all at once, and that optimization tends to eliminate redundancy in the system.

The unconstrained multiobjective model by Formiga et al. [43] tackles some of these issues by appreciating redundancy in the network and discarding the constraint on nodal pressure. Further extending this model to account for varying environmental parameters (e.g., different sets of demands) is possible through applying robust optimization. Robust optimization usually requires more evaluations in order to converge to a set of good solutions, hence the aim of this study was to compare the basic multiobjective model (raw) with the multiobjective robust model (robust) on attained robustness of solutions, using a fixed budget of evaluations by the hydraulic simulator. For comparing the different result sets, which represent Pareto front approximations, state-of-the-art methods for performance assessment were used: the $\mathcal{S}$ metric expressing the space covered in the objective space by an approximation set, and attainment surface plots for visually comparing the fronts obtained.

Initially raw experiments were performed in order to determine the best variant of the tested Multiobjective Evolutionary Algorithms (MOEAs) NSGA-II and SMS-EMOA in the context of WDN optimization. In this work NSGA-II and SMS-EMOA employ variation that operates on individuals consisting of decision variables representing the diameter code selected per pipe. It was chosen to view these numerical diameter codes, which point to size/price/roughness triples in the table of commercial diameters, as nominal values. In principle the ordering determined by the size component of these diameter triples could be exploited during the optimization, but because of the limited amount of diameters defined per test problem (i.e., 6–15) it is assumed that the nominal approach is faster. The basic nominal variation has a self-adaptive counterpart, which is governed by a single mutation probability included in each individual and that is prone to variation itself. The self-adaptive variation gives rise to self-adaptive variants of NSGA-II and SMS-EMOA, of which NSGA-II-sa-sp-λ and SMS-EMOA-sa were shown to deliver the best results. NSGA-II-sa-sp-λ outperformed all other NSGA-II variants, while SMS-EMOA-sa even outperformed all other raw methods when tested on the three examined WDN problems: Two Loop, Hanoi, and New York City. Furthermore, in the context of WDN optimization, the SMS-EMOA-dp and SMS-EMOA-sa-dp variants (i.e., utilizing dominating points selection when the population contains dominated individuals) were shown not to improve on SMS-EMOA and SMS-EMOA-sa, either with respect to quality of results or elapsed runtime.

Applying the straightforward extensions for robust evaluation SEM and MEM, the robust model was seemingly shown to have no added value whatsoever over the raw model: the raw self-adaptive variant of SMS-EMOA (SMS-EMOA-sa) delivered the best results in terms of robustness. The optimal raw solutions found by SMS-EMOA-sa appear to also be the robust optimal solutions. This assumption is backed by the claim of Giustolisi et al. [47] of evidence of the fact that optimal raw solutions belong to, or are at least close to, the Pareto front of robust solutions. The course of action for obtaining robust solutions would be applying the raw SMS-EMOA-sa method, and then re-evaluating the results to determine the robust objective values.

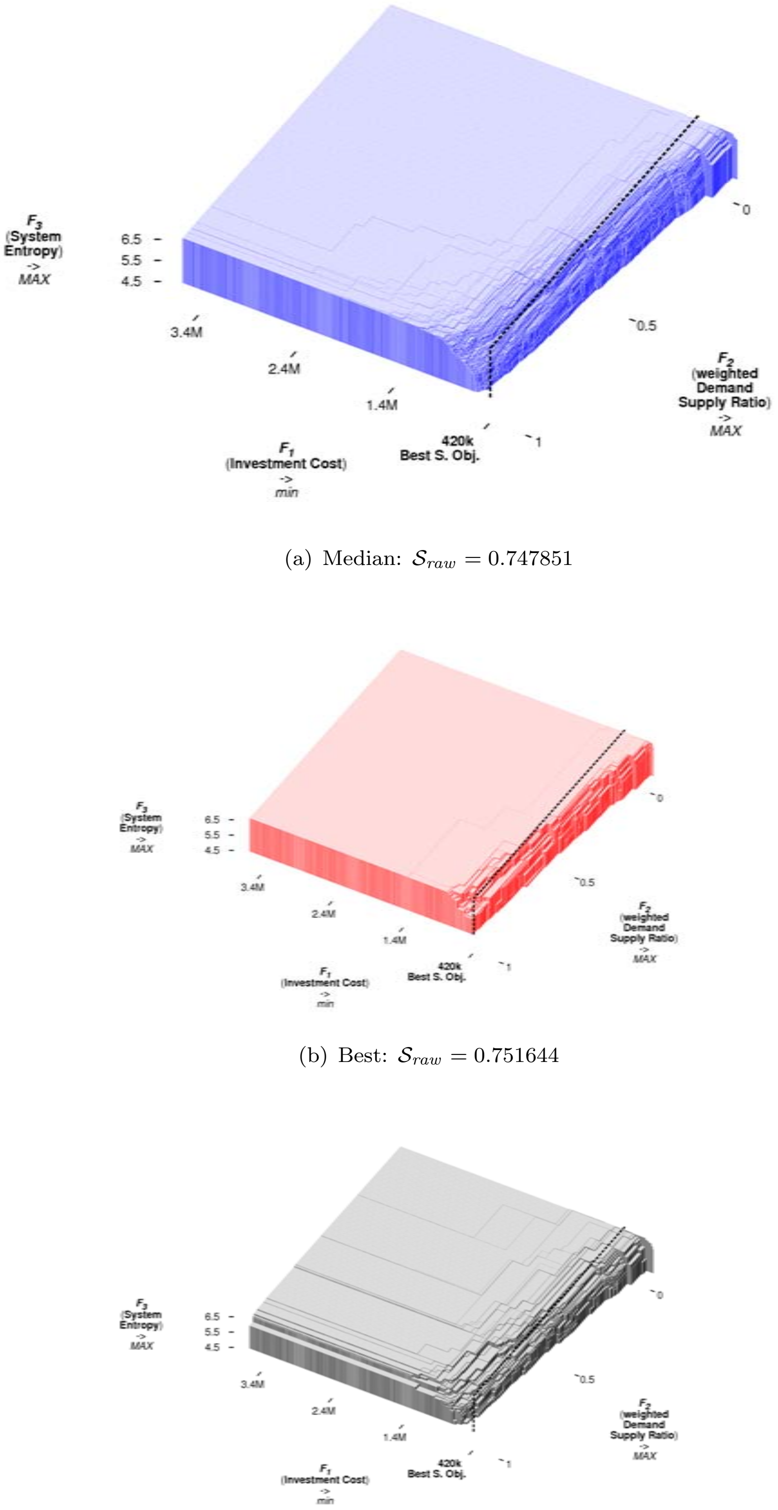
The outcome of the model comparison shows that SMS-EMOA-sa is a promising tool for WDN optimization. However, usage of the simple SEM and MEM schemes restricts us from drawing sharp conclusions with respect to the added value of the robust model.

SEM suffers from generations in which all offspring solutions are paired with a single beneficial sample of environmental parameters, as these biased solutions will get preserved in the population because of the coupling with the elitist Multiobjective Evolutionary Algorithms. MEM on the other hand is hindered because it requires a lot of evaluations, since the quality of each solution is averaged over multiple samples of environmental parameters. However, very recently the archive-based MEM derivative ABRSS became available [61], which aims at more efficient use of evaluations. ABRSS could be applied to WDN optimization, possibly coupled with re-evaluation of all solutions contained in the population at a certain generation and averaging over the earlier obtained fitness values, as done in the MEM-based robust NSGA-II derivative RNSGAII, used for robust WDN optimization in [54].

Future work should involve deviation from the three objective WDN optimization model by Formiga et al. [43], in order to allow for a comparison with results from related studies, for instance by applying the bi-objective Investment Cost and robustness model as used with RNSGAII in [54] (similar to the model used with OPTIMOGA in [47], which also applies MEM-like robust evaluation based on sampling). Convergence behavior of MEM extended methods is to be assessed by allowing the optimization to continue until stagnation occurs, instead of using a fixed evaluation budget, providing insight into the number of evaluations required by MEM to stabilize. Lastly, different probability distributions [47, 54] for varying the samples of environmental parameters in robust WDN optimization are to be tested, to determine whether under these conditions the ordering between the performance of the different methods (i.e., raw and robust) holds.

Appendix A

Raw Attainment Surfaces

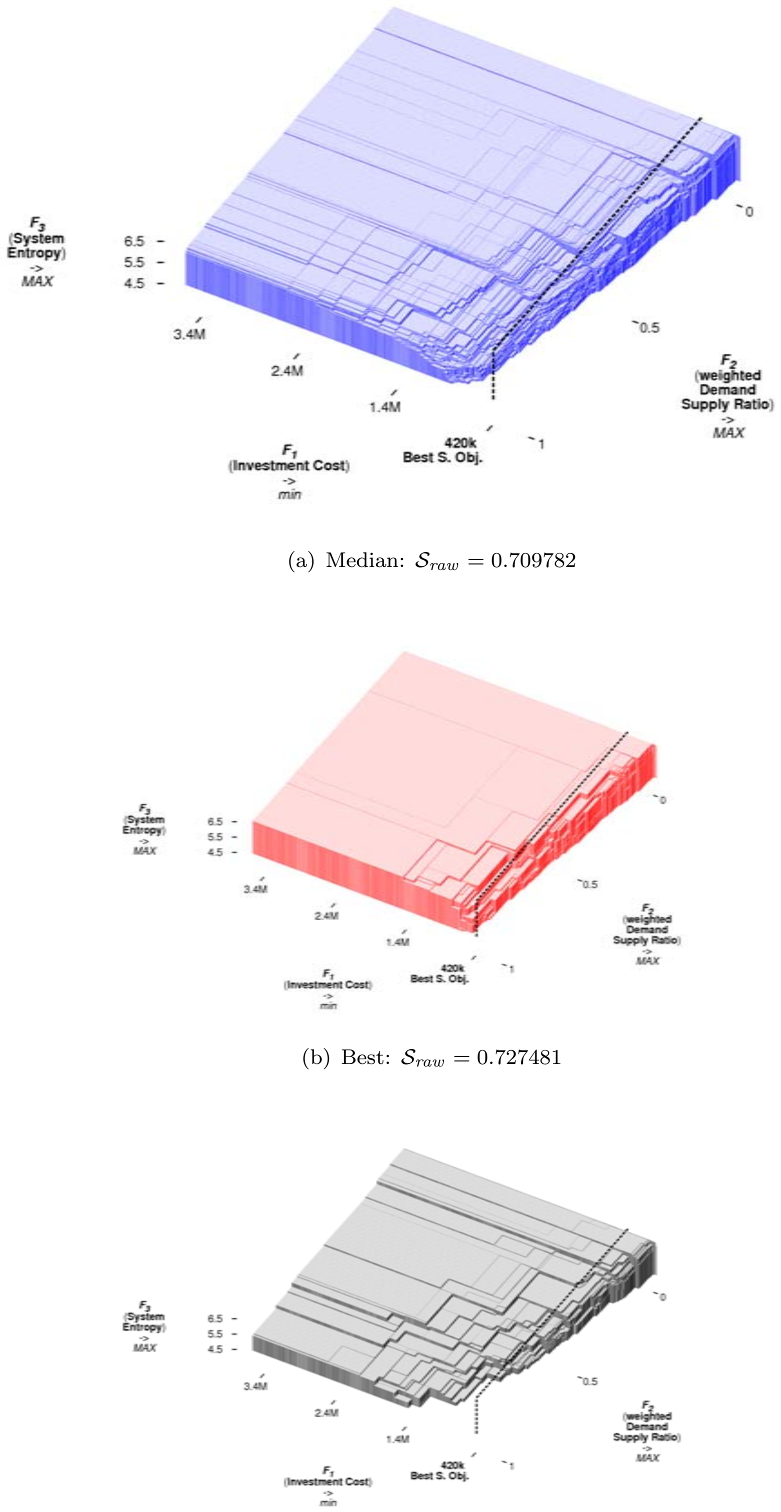


(a) Median: $\mathcal{S}_{raw} = 0.747851$

(b) Best: $\mathcal{S}_{raw} = 0.751644$

(c) Worst: $\mathcal{S}_{raw} = 0.745416$

Figure A.1: **Raw – Two Loop – NSGA-II**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

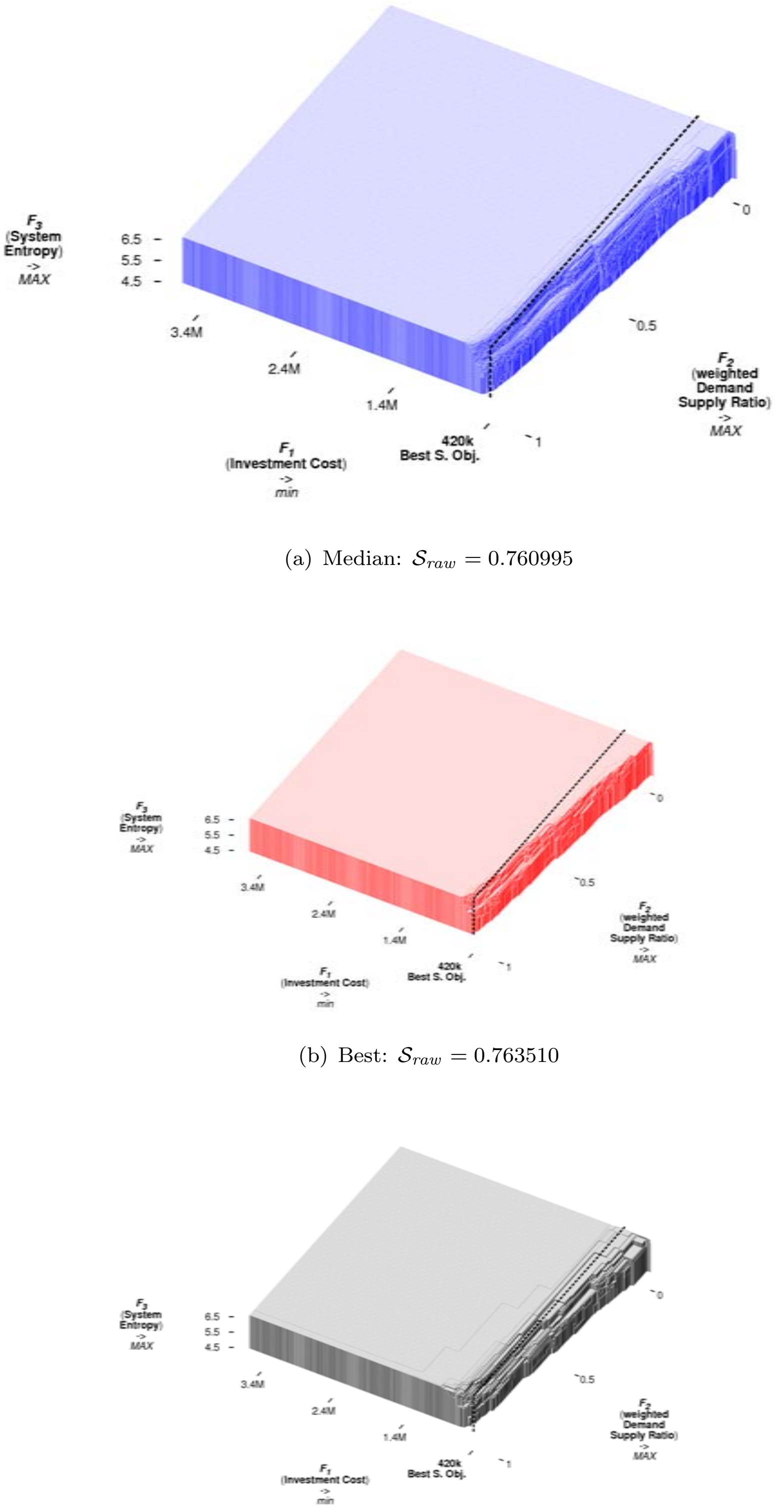


(a) Median: $\mathcal{S}_{raw} = 0.709782$

(b) Best: $\mathcal{S}_{raw} = 0.727481$

(c) Worst: $\mathcal{S}_{raw} = 0.678980$

Figure A.2: **Raw – Two Loop – NSGA-II-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

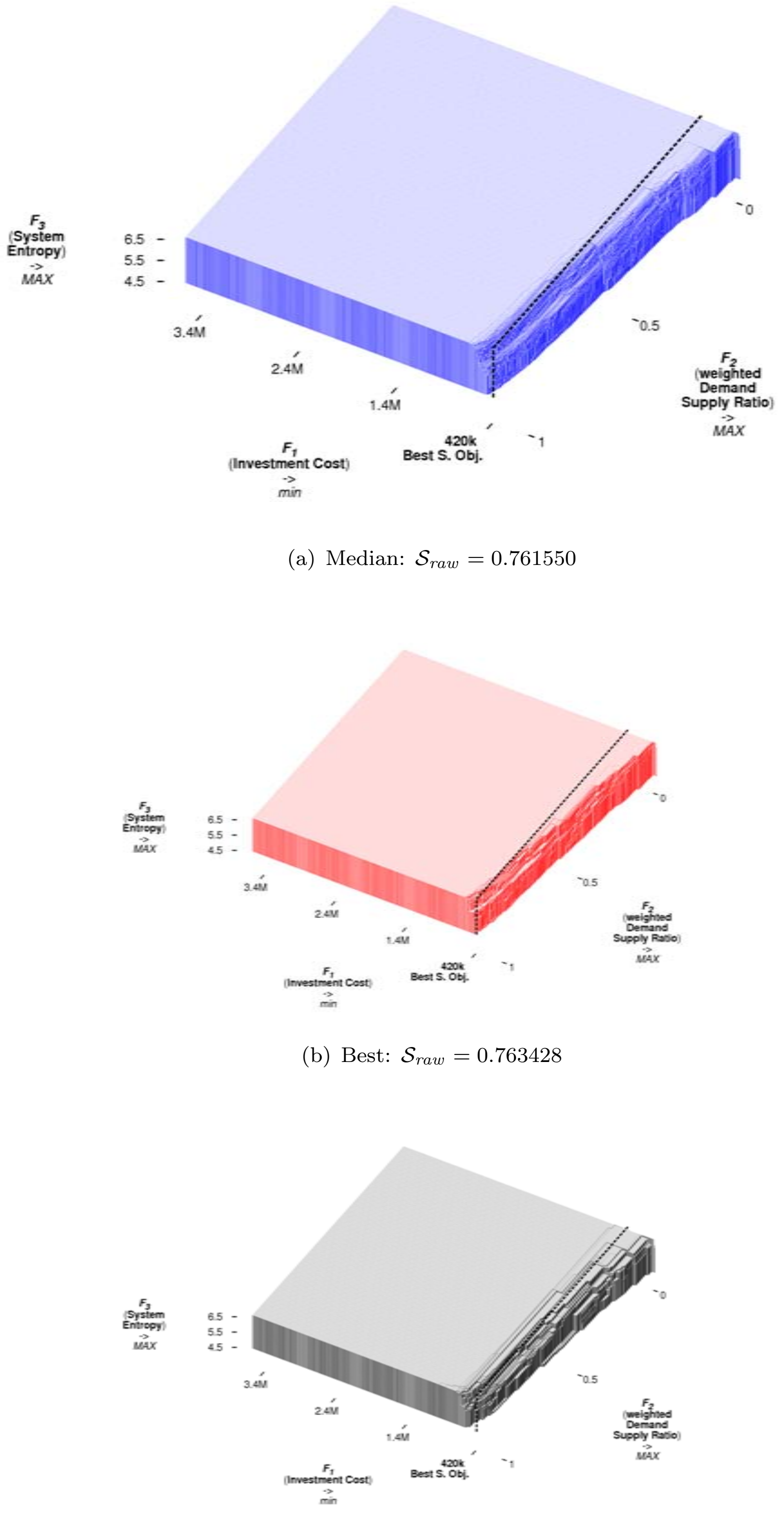


(a) Median: $\mathcal{S}_{raw} = 0.760995$

(b) Best: $\mathcal{S}_{raw} = 0.763510$

(c) Worst: $\mathcal{S}_{raw} = 0.757428$

Figure A.3: **Raw – Two Loop – NSGA-II-sa-sp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

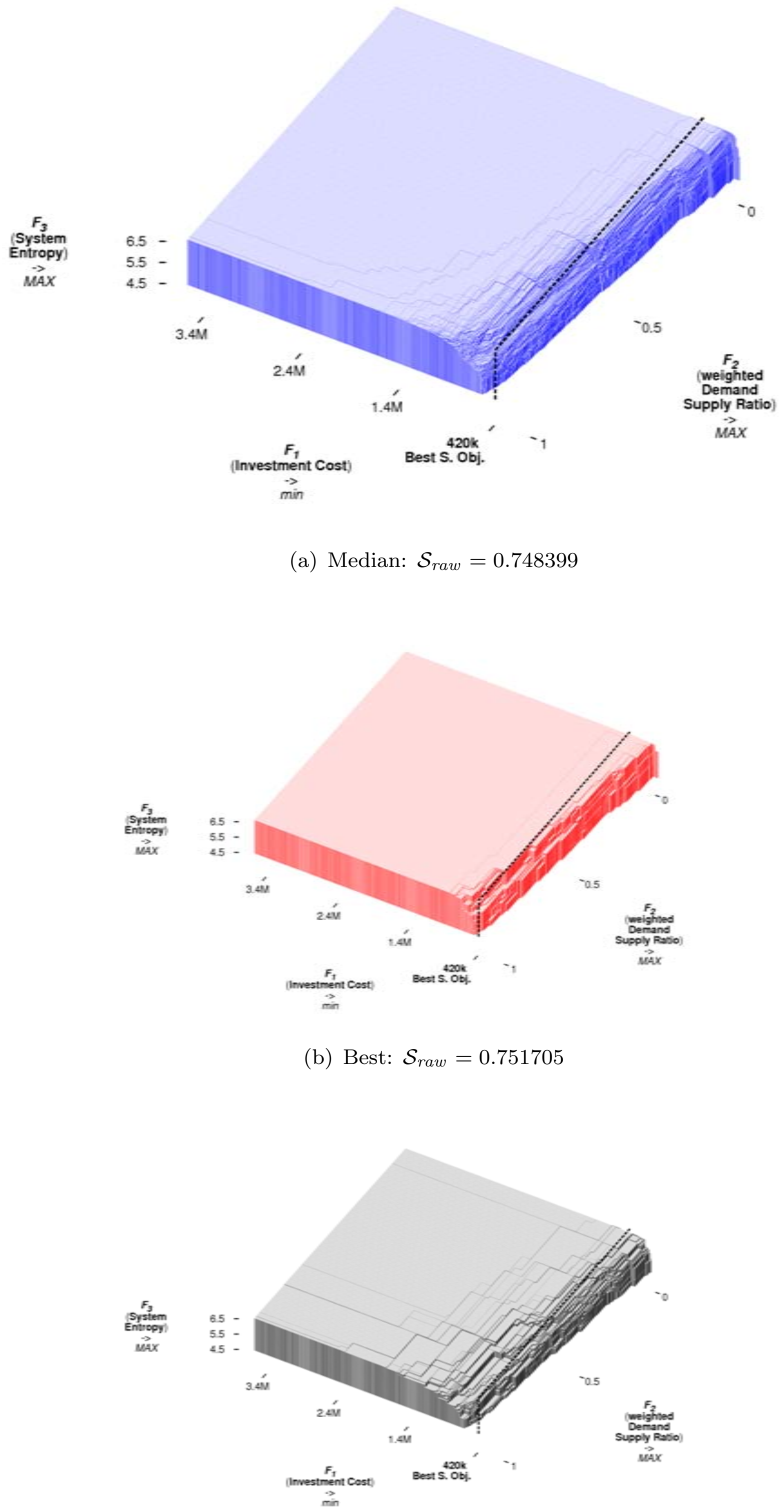


(a) Median: $\mathcal{S}_{raw} = 0.761550$

(b) Best: $\mathcal{S}_{raw} = 0.763428$

(c) Worst: $\mathcal{S}_{raw} = 0.756467$

FIGURE A.4: **Raw – Two Loop – NSGA-II-sa-sp-λ**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

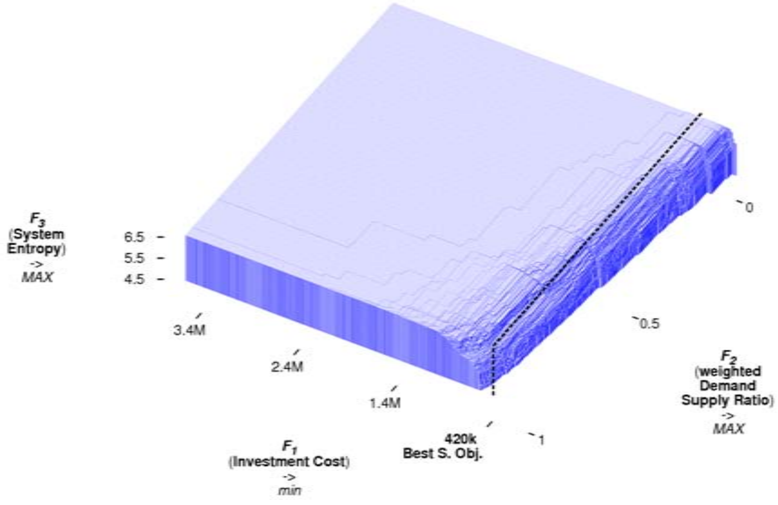


(a) Median: $\mathcal{S}_{raw} = 0.748399$

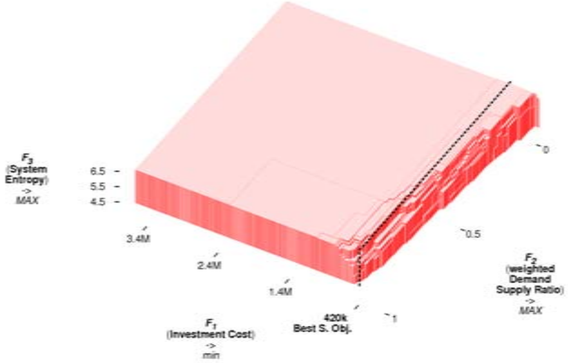
(b) Best: $\mathcal{S}_{raw} = 0.751705$

(c) Worst: $\mathcal{S}_{raw} = 0.743534$

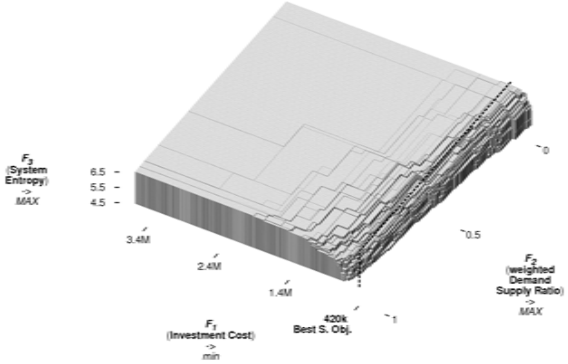
Figure A.5: **Raw – Two Loop – SMS-EMOA**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.



(a) Median: $\mathcal{S}_{raw} = 0.747742$

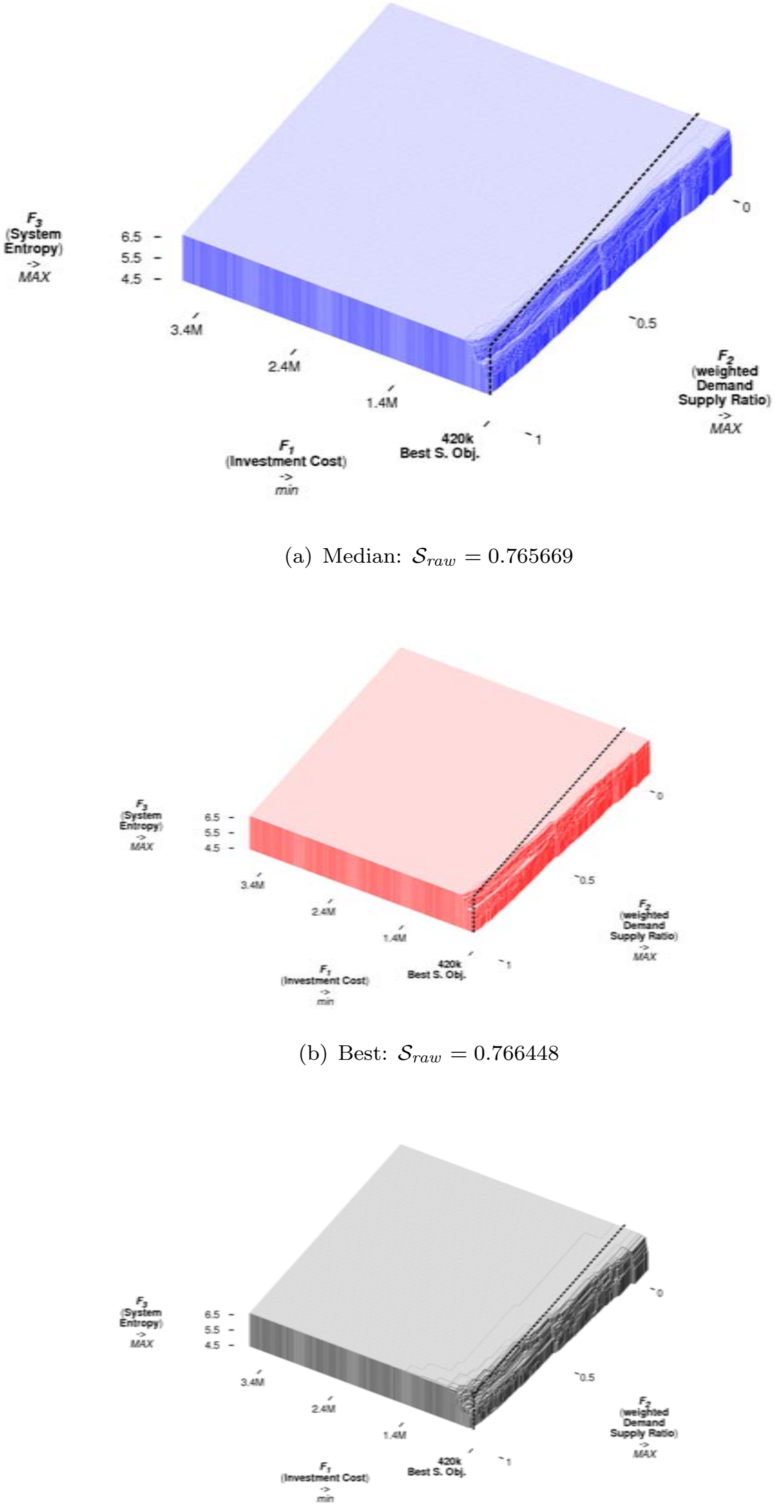


(b) Best: $\mathcal{S}_{raw} = 0.752085$



(c) Worst: $\mathcal{S}_{raw} = 0.744476$

FIGURE A.6: **Raw – Two Loop – SMS-EMOA-dp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-dp runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

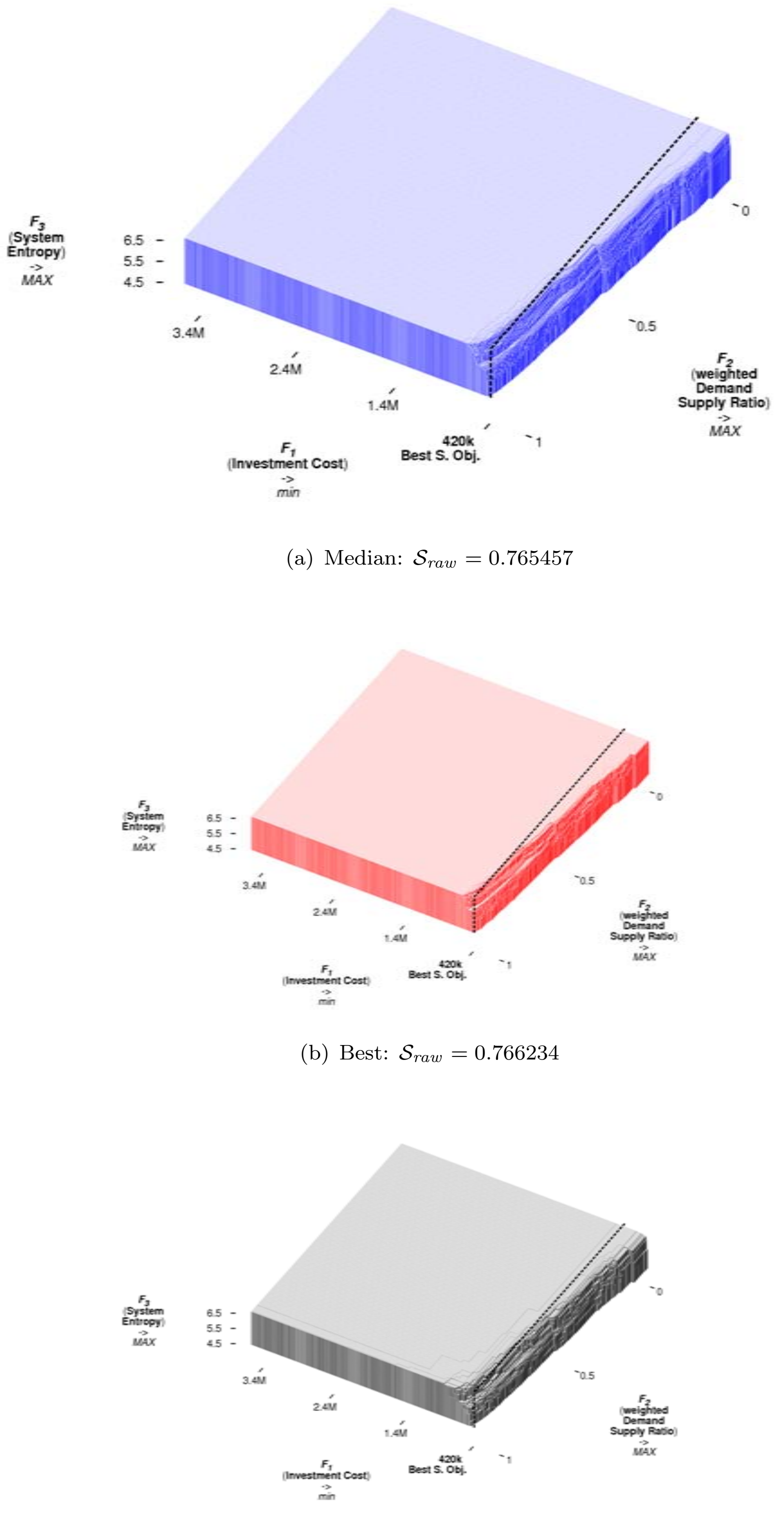


(a) Median: $\mathcal{S}_{raw} = 0.765669$

(b) Best: $\mathcal{S}_{raw} = 0.766448$

(c) Worst: $\mathcal{S}_{raw} = 0.765079$

Figure A.7: **Raw – Two Loop – SMS-EMOA-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

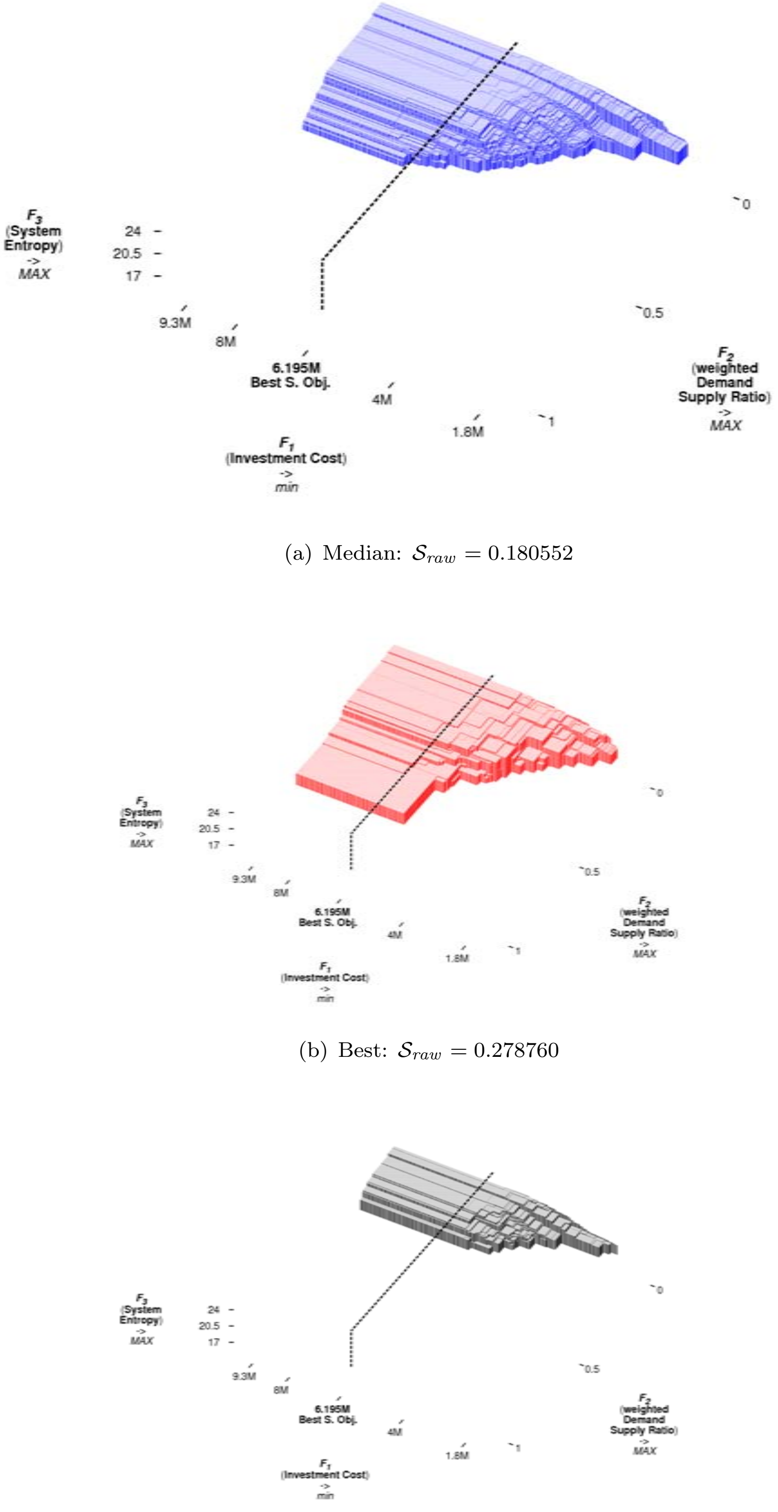


(a) Median: $\mathcal{S}_{raw} = 0.765457$

(b) Best: $\mathcal{S}_{raw} = 0.766234$

(c) Worst: $\mathcal{S}_{raw} = 0.764530$

FIGURE A.8: **Raw – Two Loop – SMS-EMOA-sa-dp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-dp runs for test problem Two Loop. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

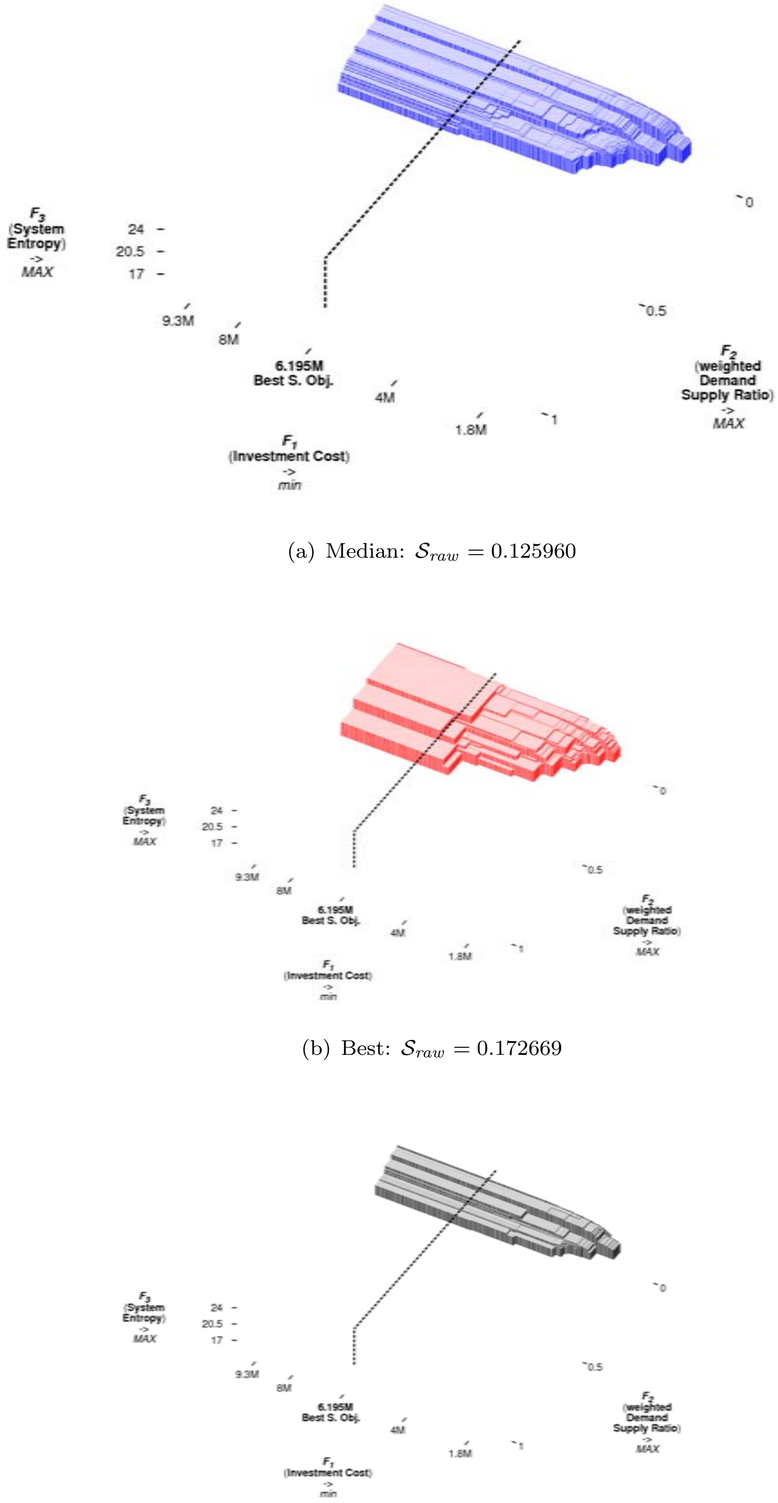


(a) Median: $\mathcal{S}_{raw} = 0.180552$

(b) Best: $\mathcal{S}_{raw} = 0.278760$

(c) Worst: $\mathcal{S}_{raw} = 0.119362$

Figure A.9: **Raw – Hanoi – NSGA-II**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

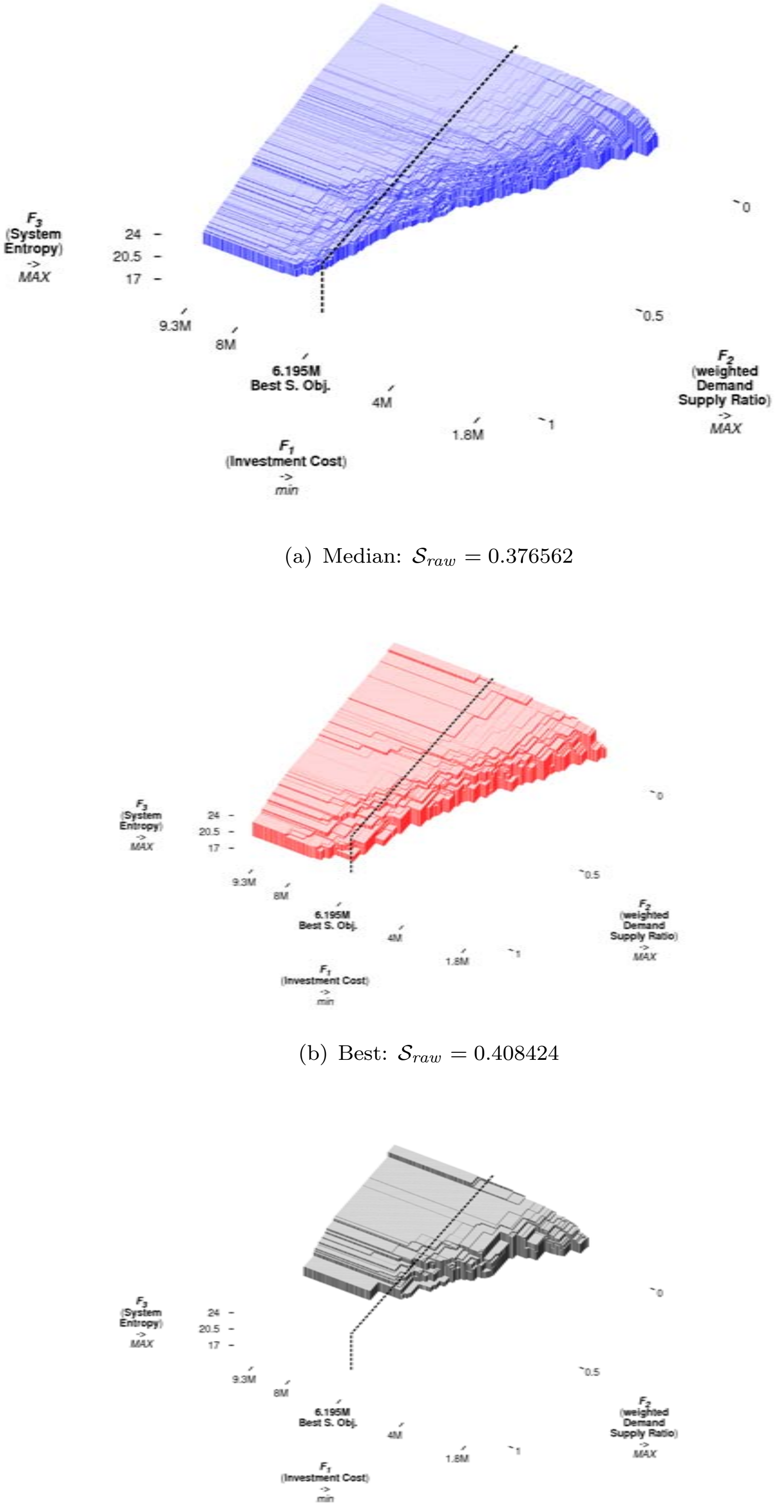


(a) Median: $\mathcal{S}_{raw} = 0.125960$

(b) Best: $\mathcal{S}_{raw} = 0.172669$

(c) Worst: $\mathcal{S}_{raw} = 0.088576$

Figure A.10: **Raw – Hanoi – NSGA-II-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

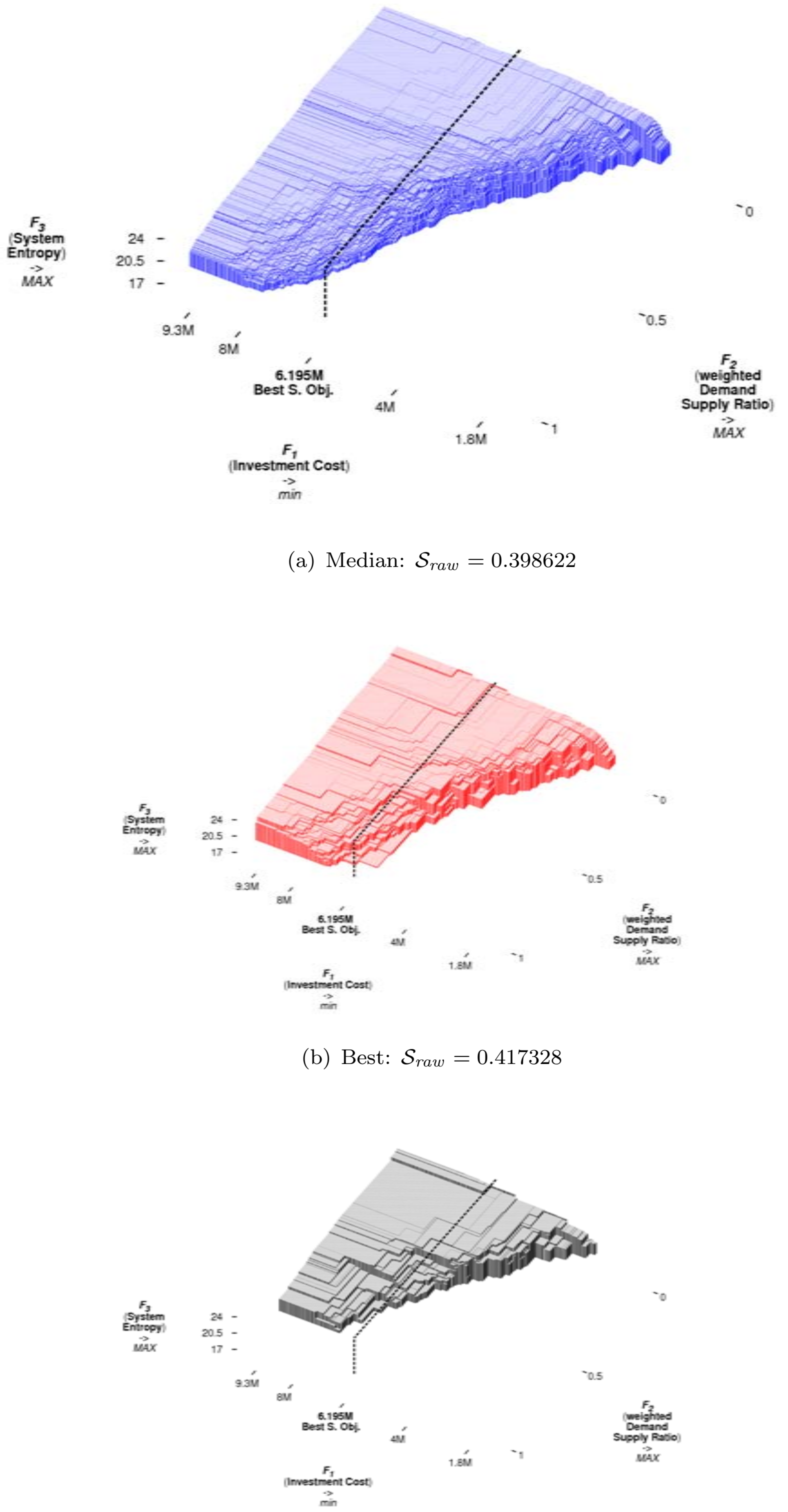


(a) Median: $\mathcal{S}_{raw} = 0.376562$

(b) Best: $\mathcal{S}_{raw} = 0.408424$

(c) Worst: $\mathcal{S}_{raw} = 0.293202$

Figure A.11: **Raw – Hanoi – NSGA-II-sa-sp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

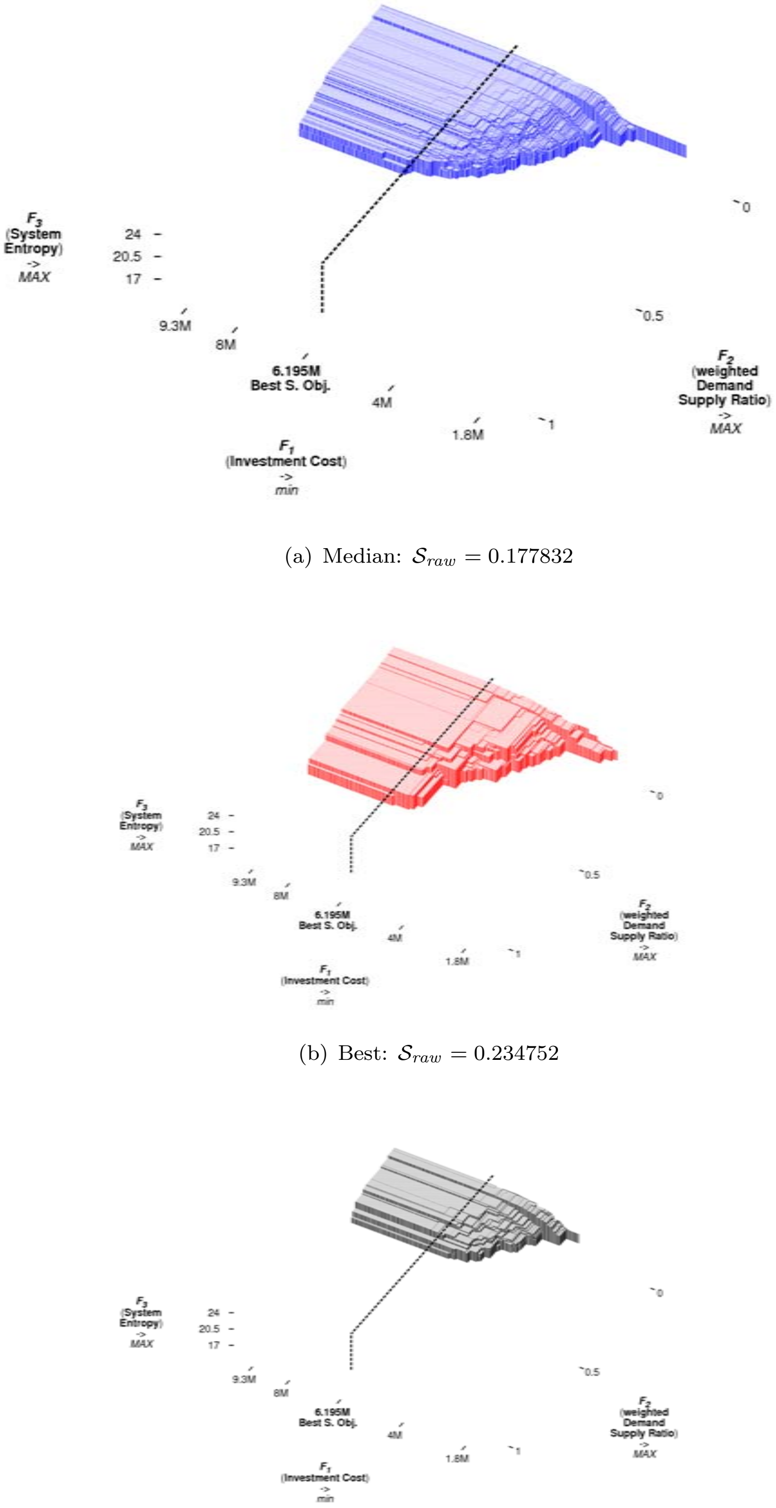


(a) Median: $\mathcal{S}_{raw} = 0.398622$

(b) Best: $\mathcal{S}_{raw} = 0.417328$

(c) Worst: $\mathcal{S}_{raw} = 0.360623$

FIGURE A.12: **Raw – Hanoi – NSGA-II-sa-sp-$\boldsymbol{\lambda}$**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

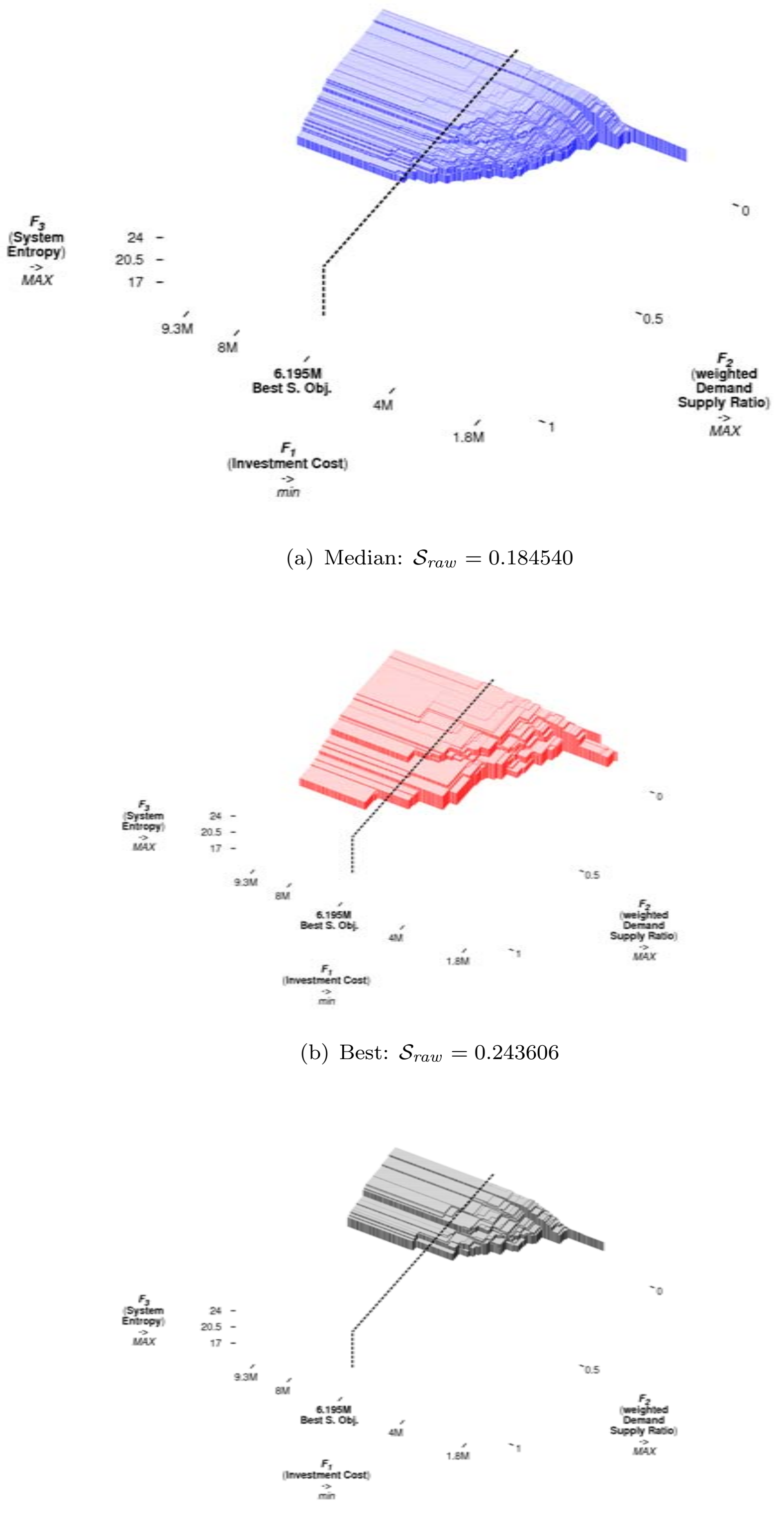


(a) Median: $\mathcal{S}_{raw} = 0.177832$

(b) Best: $\mathcal{S}_{raw} = 0.234752$

(c) Worst: $\mathcal{S}_{raw} = 0.141623$

Figure A.13: **Raw – Hanoi – SMS-EMOA**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

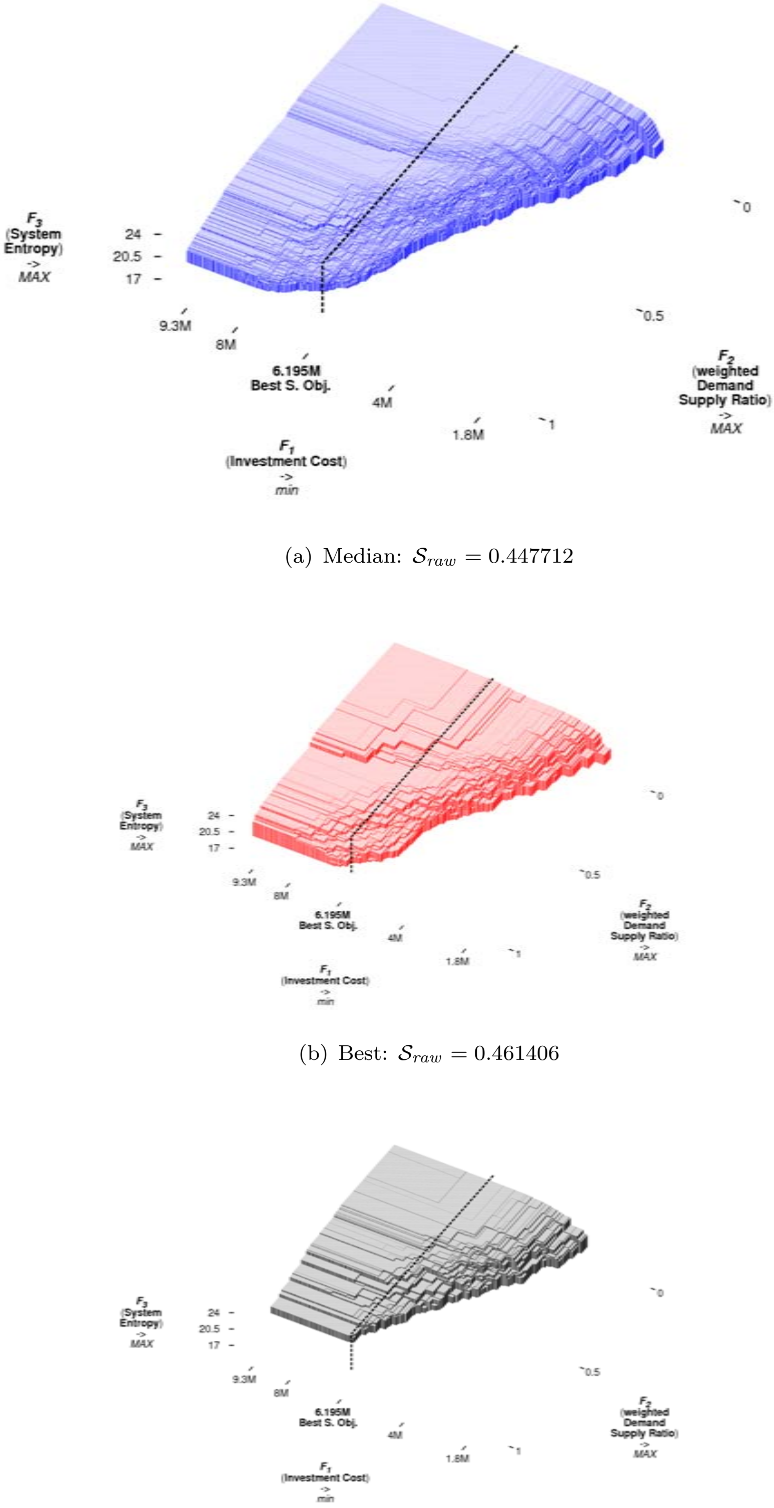


(a) Median: $\mathcal{S}_{raw} = 0.184540$

(b) Best: $\mathcal{S}_{raw} = 0.243606$

(c) Worst: $\mathcal{S}_{raw} = 0.156396$

FIGURE A.14: **Raw – Hanoi – SMS-EMOA-dp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-dp runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

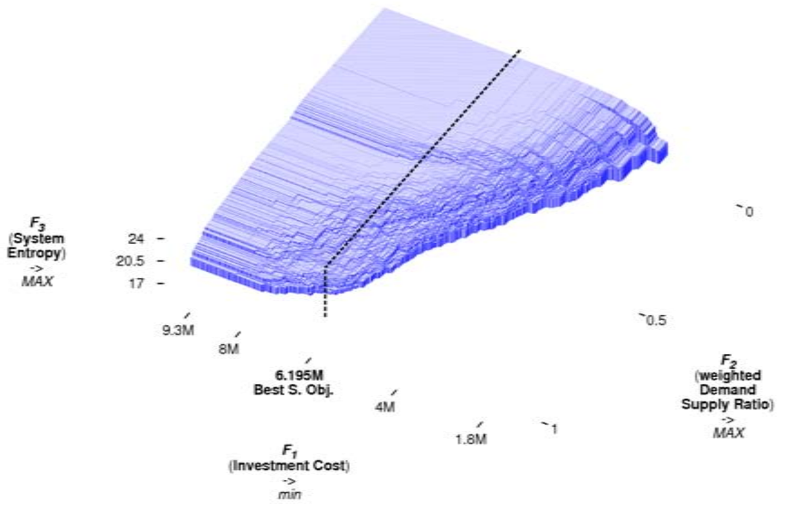


(a) Median: $\mathcal{S}_{raw} = 0.447712$

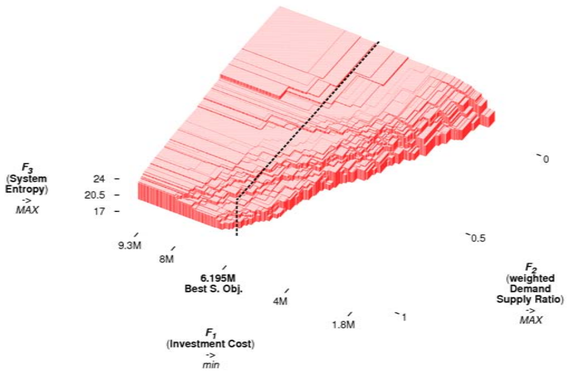
(b) Best: $\mathcal{S}_{raw} = 0.461406$

(c) Worst: $\mathcal{S}_{raw} = 0.410160$

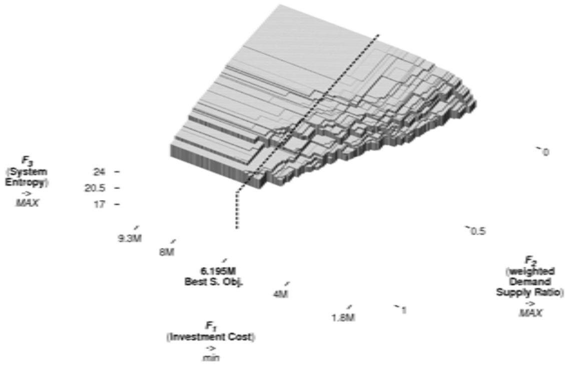
Figure A.15: **Raw – Hanoi – SMS-EMOA-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.



(a) Median: $\mathcal{S}_{raw} = 0.444481$

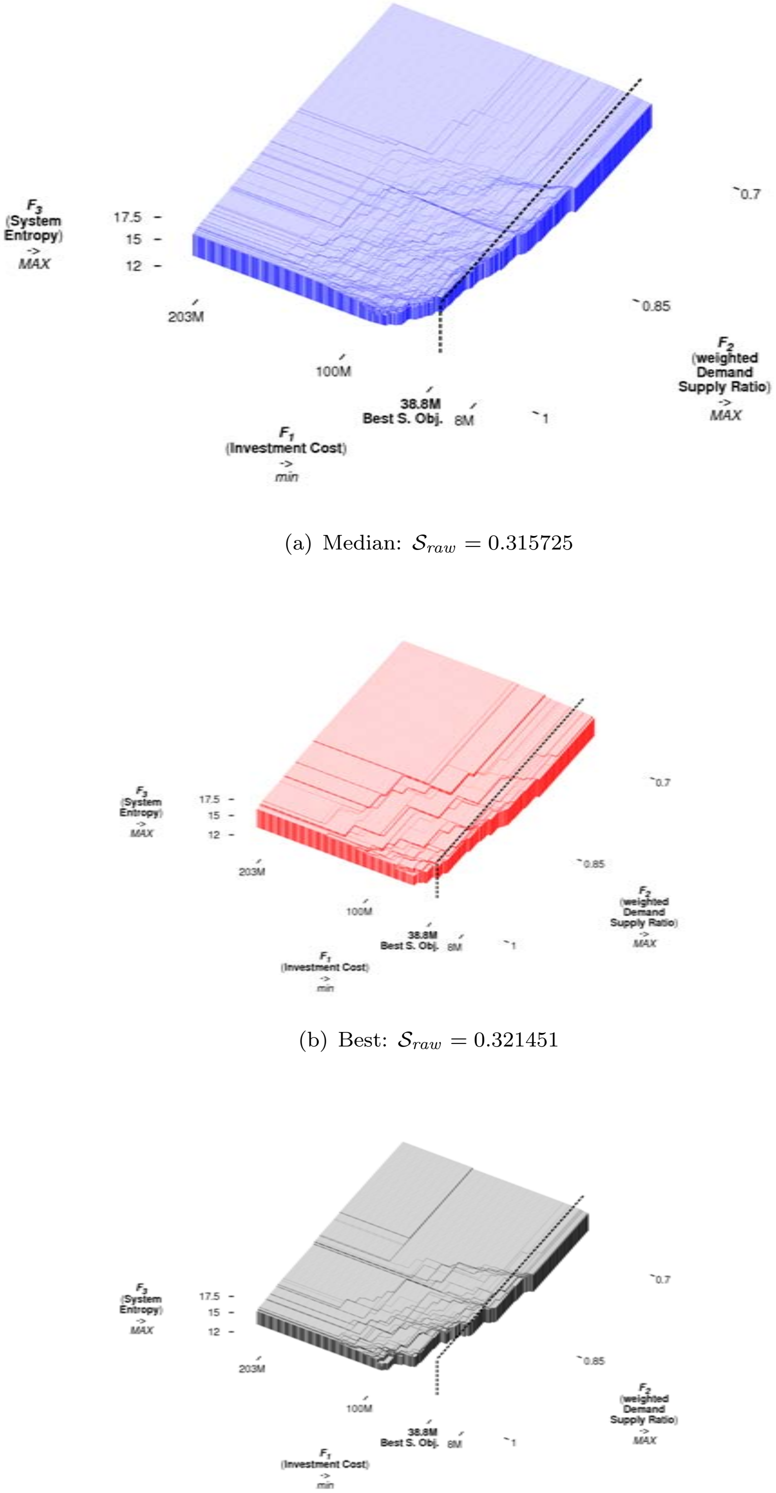


(b) Best: $\mathcal{S}_{raw} = 0.464501$



(c) Worst: $\mathcal{S}_{raw} = 0.369576$

FIGURE A.16: **Raw – Hanoi – SMS-EMOA-sa-dp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-dp runs for test problem Hanoi. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

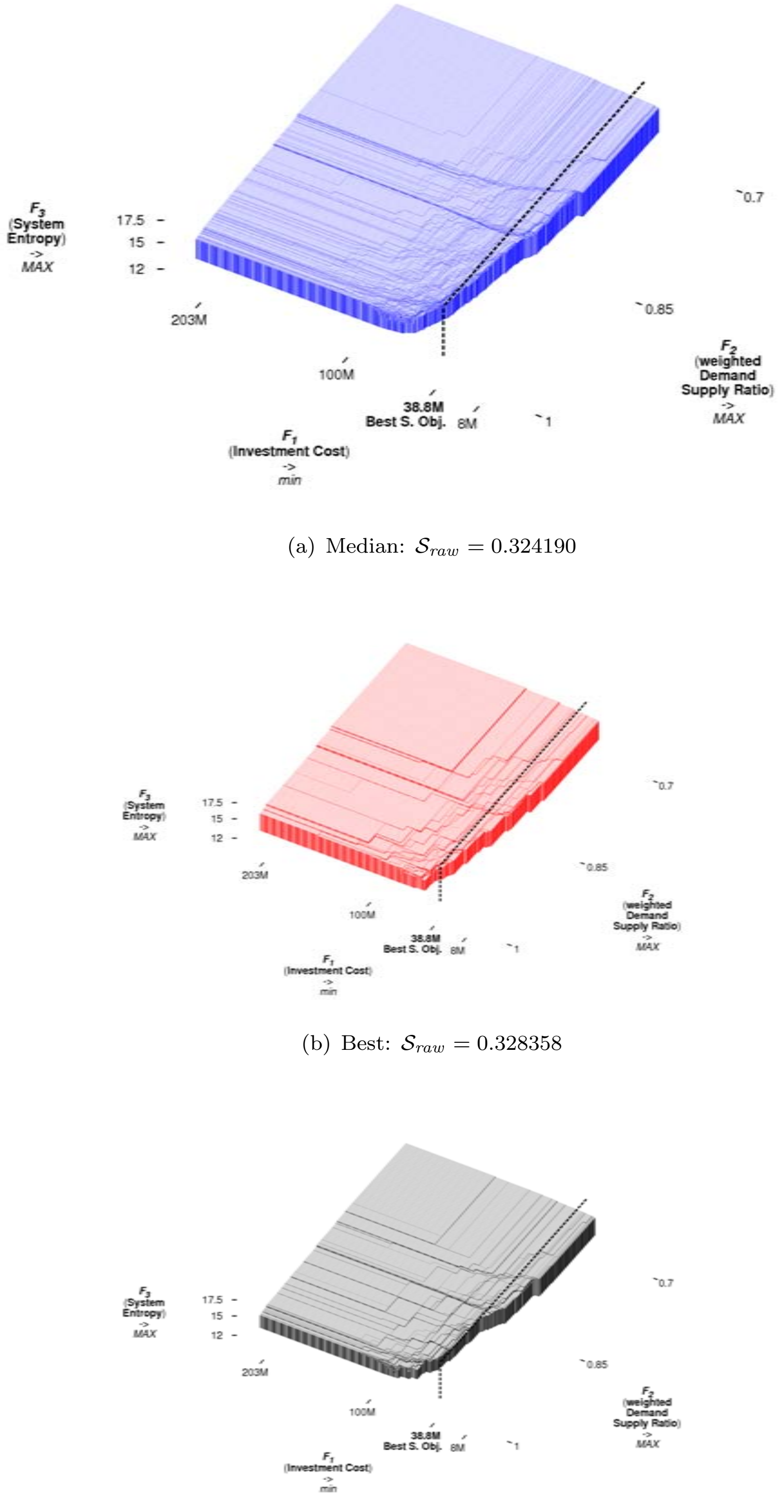


(a) Median: $\mathcal{S}_{raw} = 0.315725$

(b) Best: $\mathcal{S}_{raw} = 0.321451$

(c) Worst: $\mathcal{S}_{raw} = 0.311407$

FIGURE A.17: **Raw – New York City – NSGA-II**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

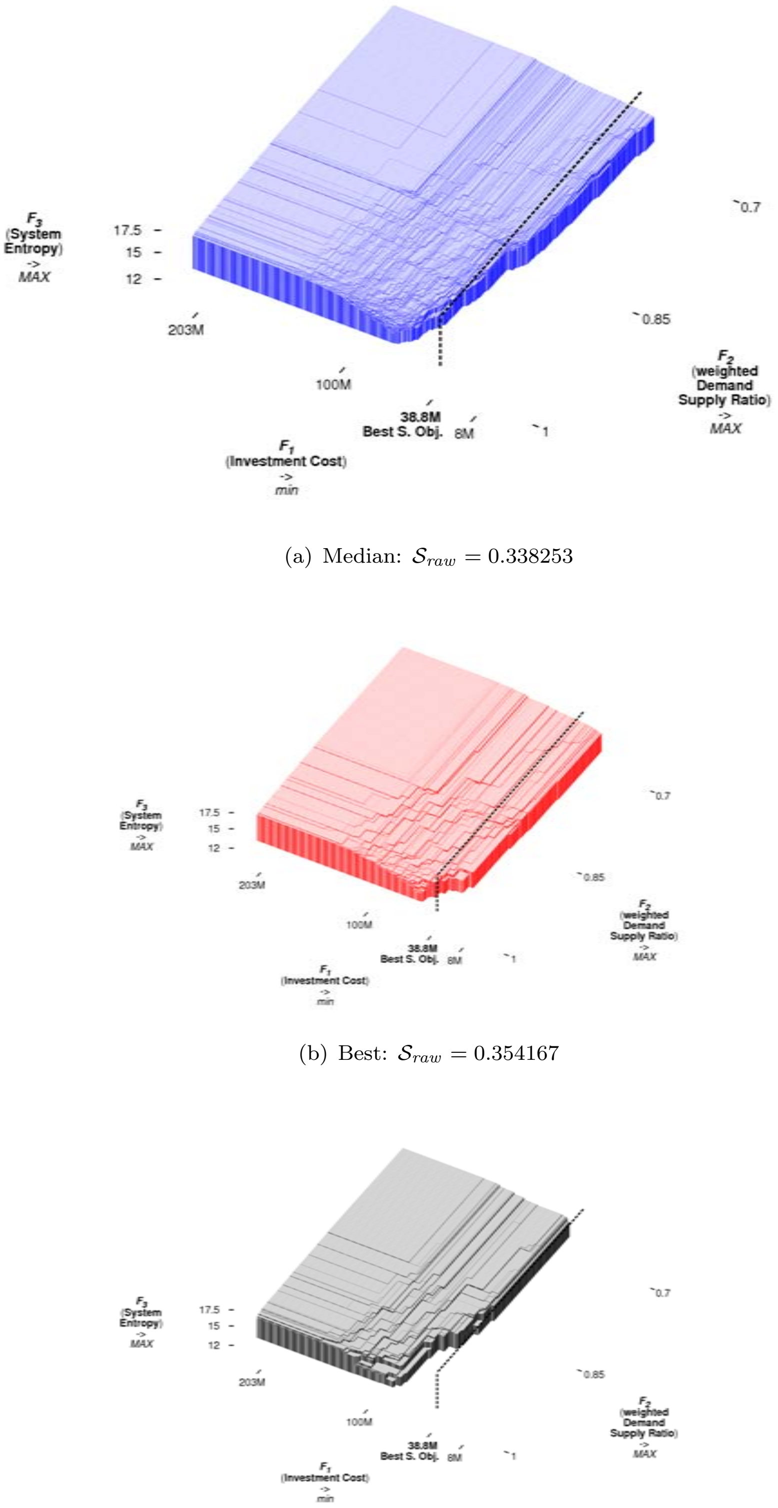


(a) Median: $\mathcal{S}_{raw} = 0.324190$

(b) Best: $\mathcal{S}_{raw} = 0.328358$

(c) Worst: $\mathcal{S}_{raw} = 0.321091$

Figure A.18: **Raw – New York City – NSGA-II-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

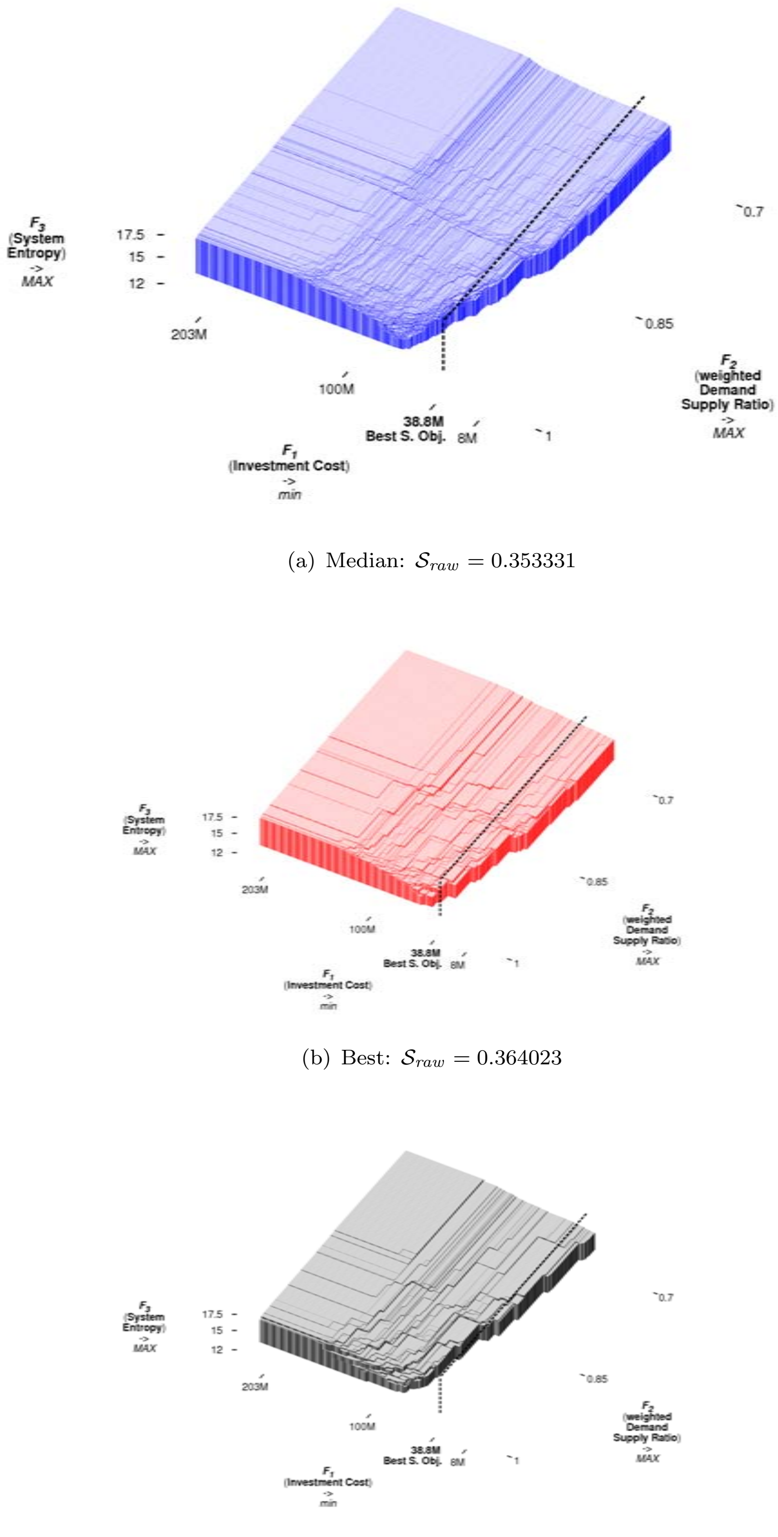


(a) Median: $\mathcal{S}_{raw} = 0.338253$

(b) Best: $\mathcal{S}_{raw} = 0.354167$

(c) Worst: $\mathcal{S}_{raw} = 0.313773$

FIGURE A.19: **Raw – New York City – NSGA-II-sa-sp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

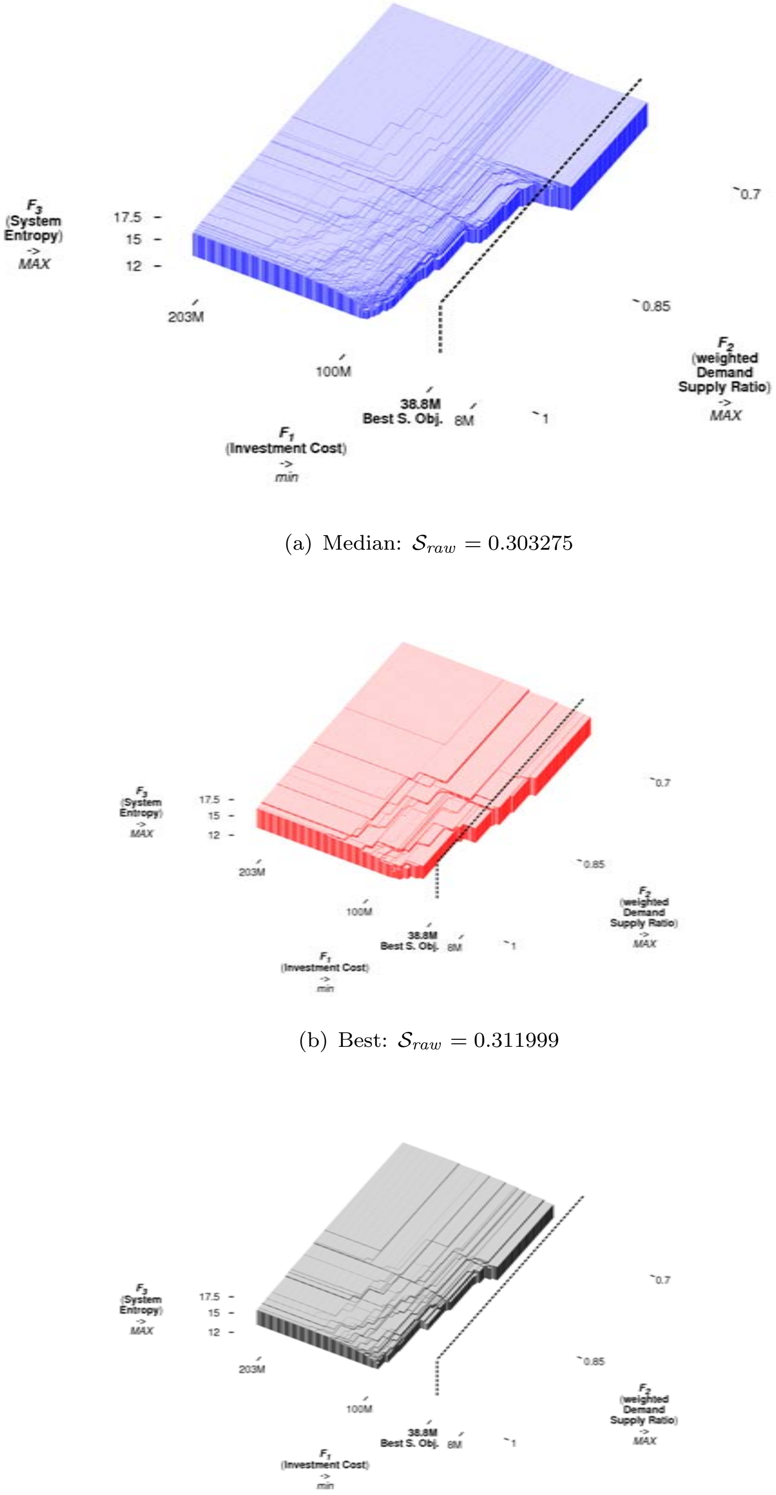


(a) Median: $\mathcal{S}_{raw} = 0.353331$

(b) Best: $\mathcal{S}_{raw} = 0.364023$

(c) Worst: $\mathcal{S}_{raw} = 0.341552$

FIGURE A.20: **Raw – New York City – NSGA-II-sa-sp-λ**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

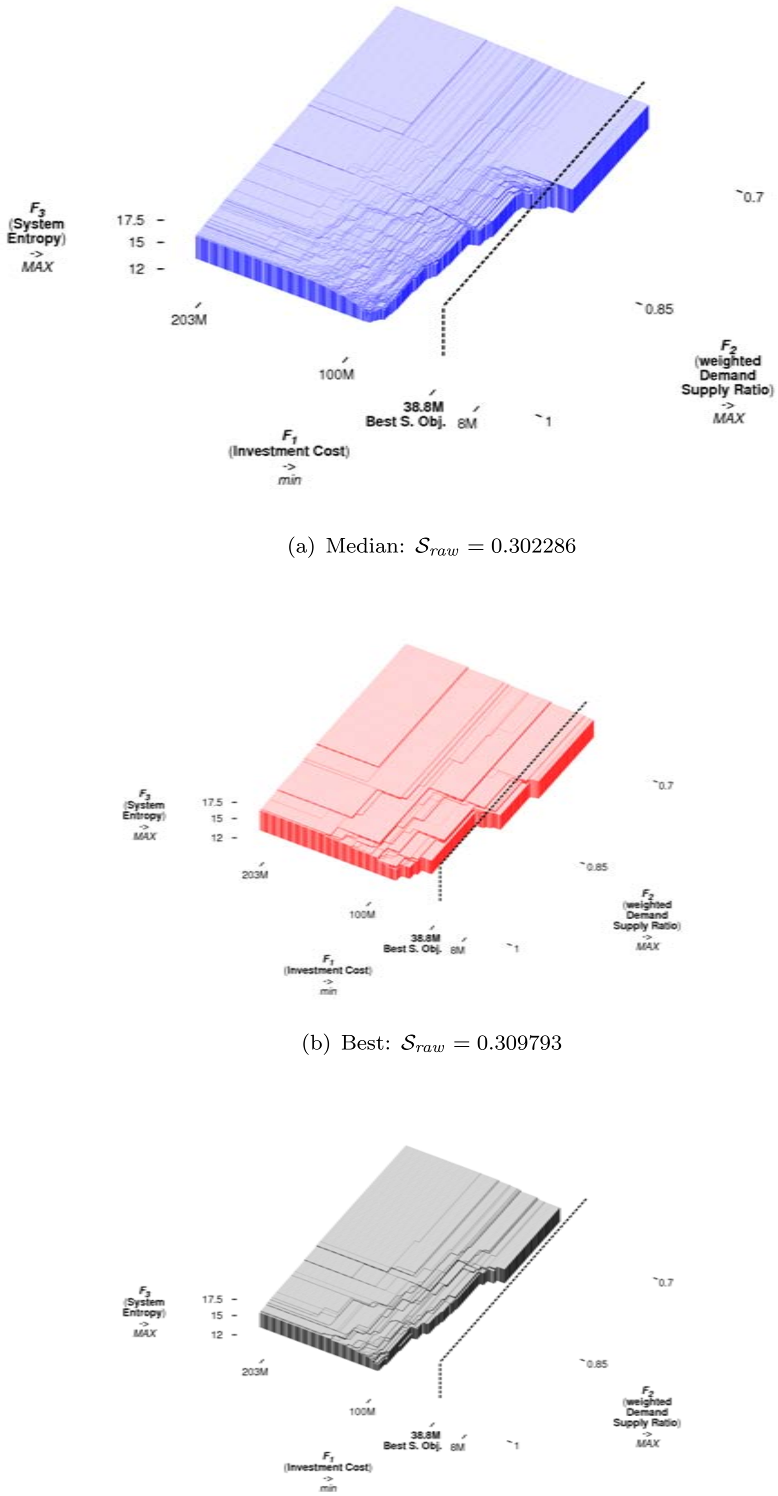


(a) Median: $\mathcal{S}_{raw} = 0.303275$

(b) Best: $\mathcal{S}_{raw} = 0.311999$

(c) Worst: $\mathcal{S}_{raw} = 0.272478$

FIGURE A.21: **Raw – New York City – SMS-EMOA**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

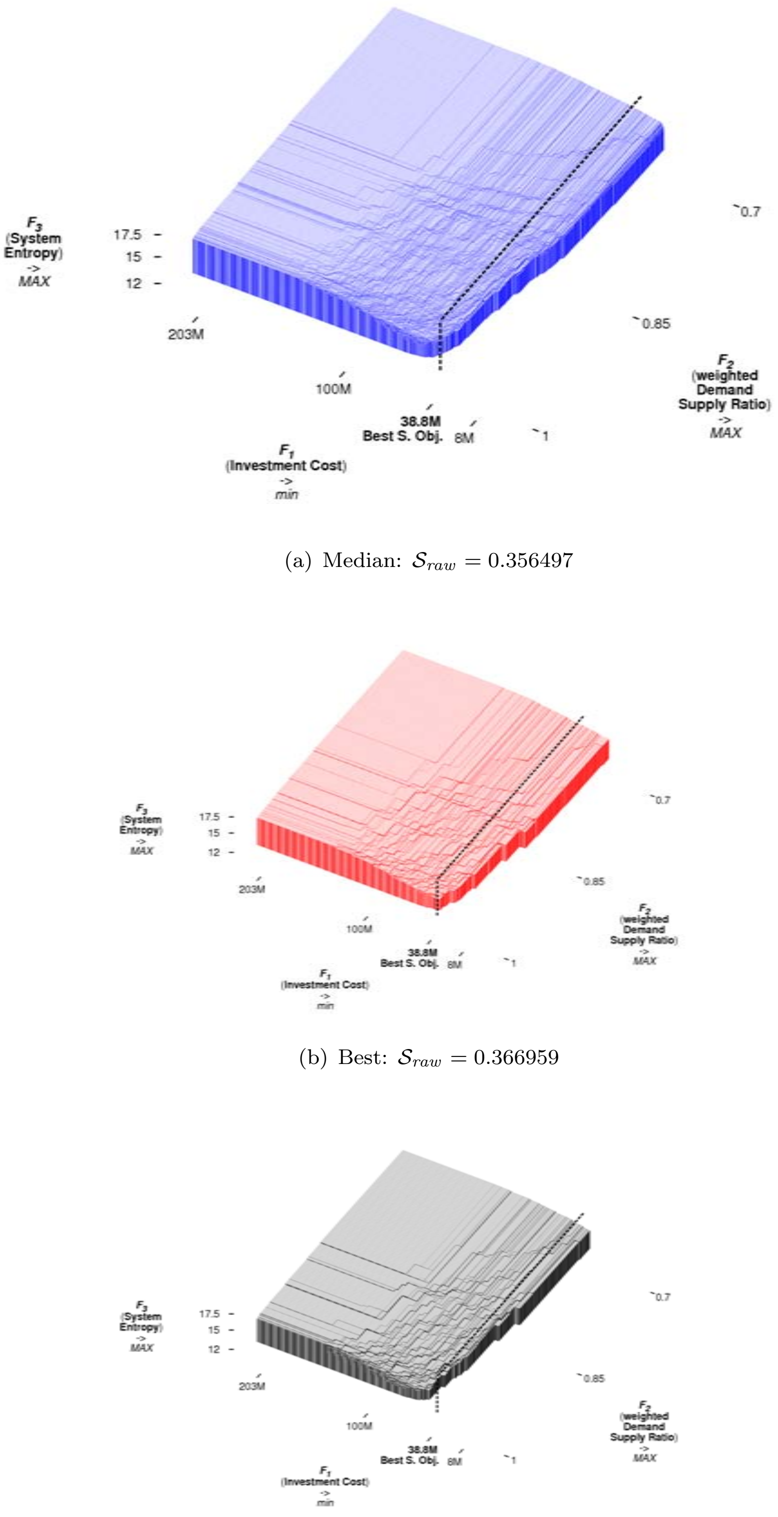


(a) Median: $\mathcal{S}_{raw} = 0.302286$

(b) Best: $\mathcal{S}_{raw} = 0.309793$

(c) Worst: $\mathcal{S}_{raw} = 0.271830$

FIGURE A.22: **Raw – New York City – SMS-EMOA-dp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-dp runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

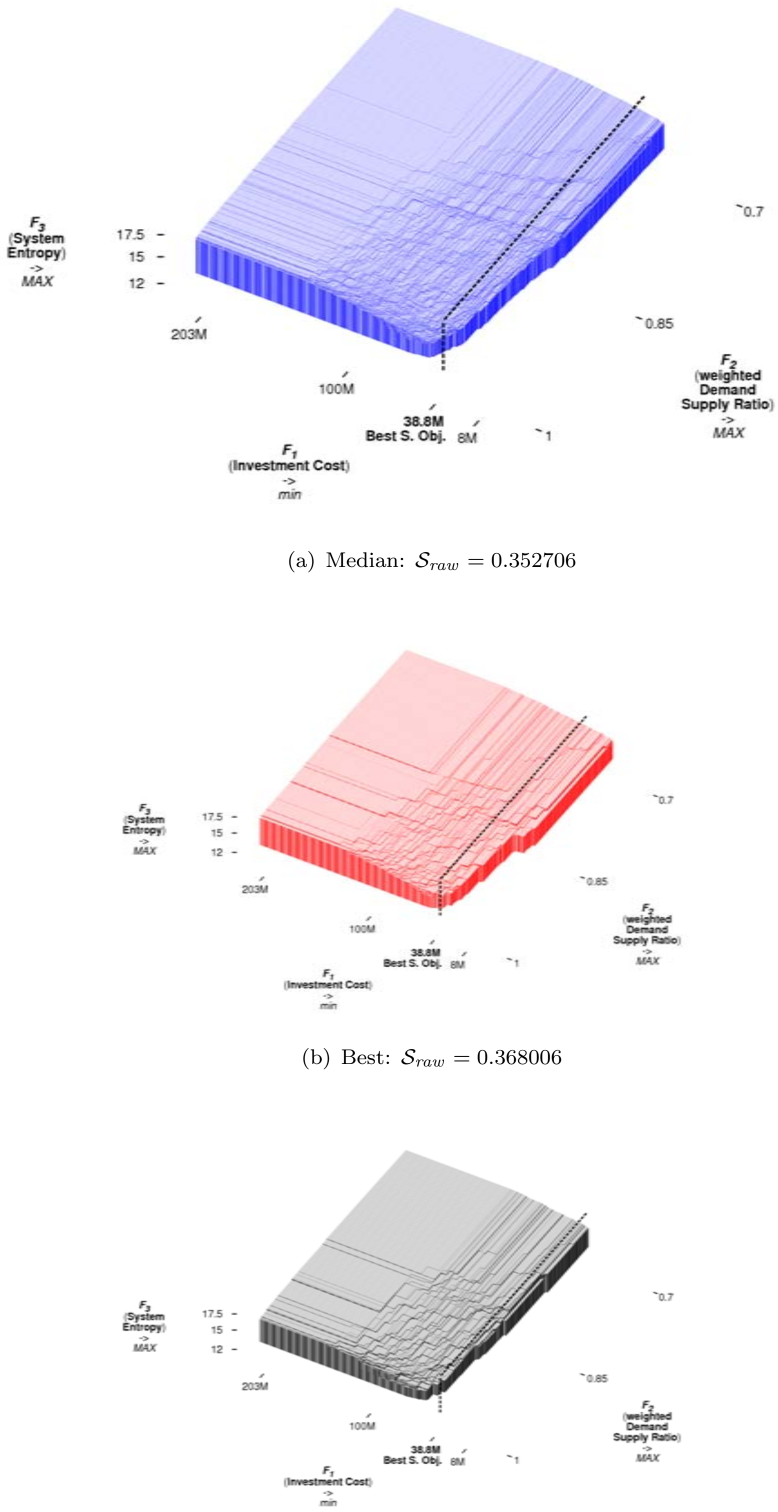


(a) Median: $\mathcal{S}_{raw} = 0.356497$

(b) Best: $\mathcal{S}_{raw} = 0.366959$

(c) Worst: $\mathcal{S}_{raw} = 0.343821$

Figure A.23: **Raw – New York City – SMS-EMOA-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.



(a) Median: $\mathcal{S}_{raw} = 0.352706$

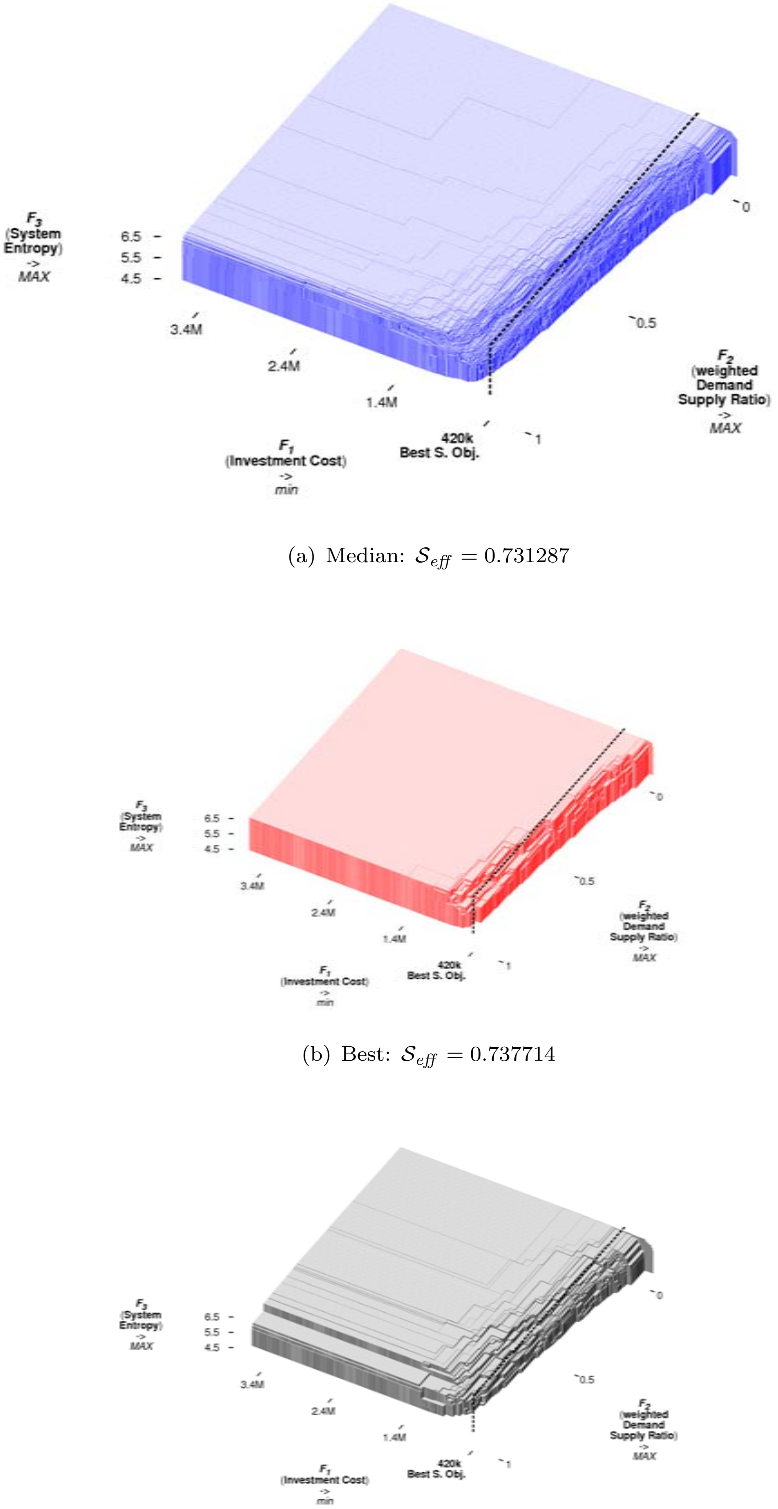
(b) Best: $\mathcal{S}_{raw} = 0.368006$

(c) Worst: $\mathcal{S}_{raw} = 0.341140$

FIGURE A.24: **Raw – New York City – SMS-EMOA-sa-dp**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-dp runs for test problem New York City. The Pareto front approximations are determined by the raw fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on raw fitness. The best single objective solution known is indicated by its Investment Cost.

Appendix B

Robust Attainment Surfaces

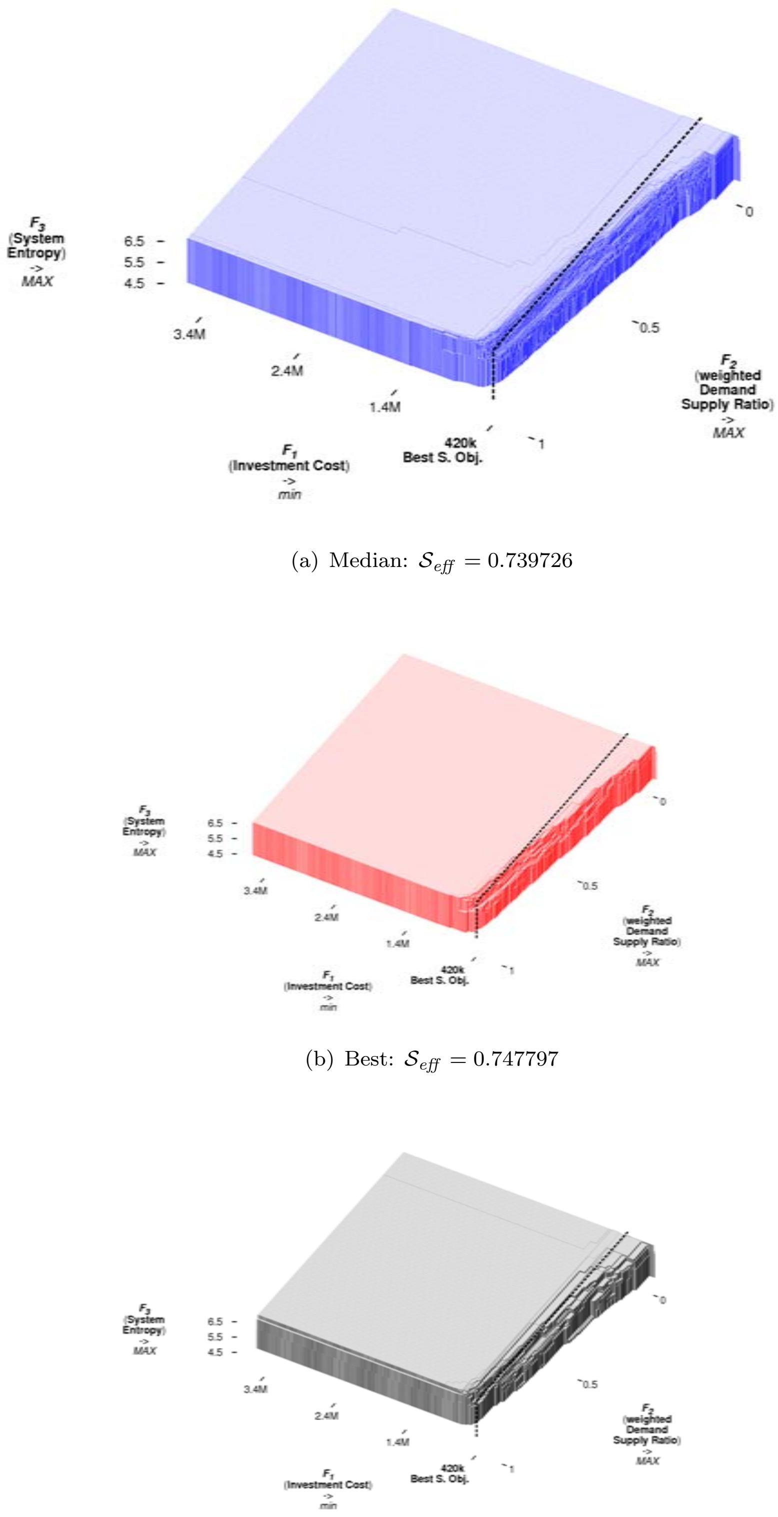


(a) Median: $\mathcal{S}_{\mathit{eff}} = 0.731287$

(b) Best: $\mathcal{S}_{\mathit{eff}} = 0.737714$

(c) Worst: $\mathcal{S}_{\mathit{eff}} = 0.721586$

Figure B.1: **Robust – Two Loop – NSGA-II**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

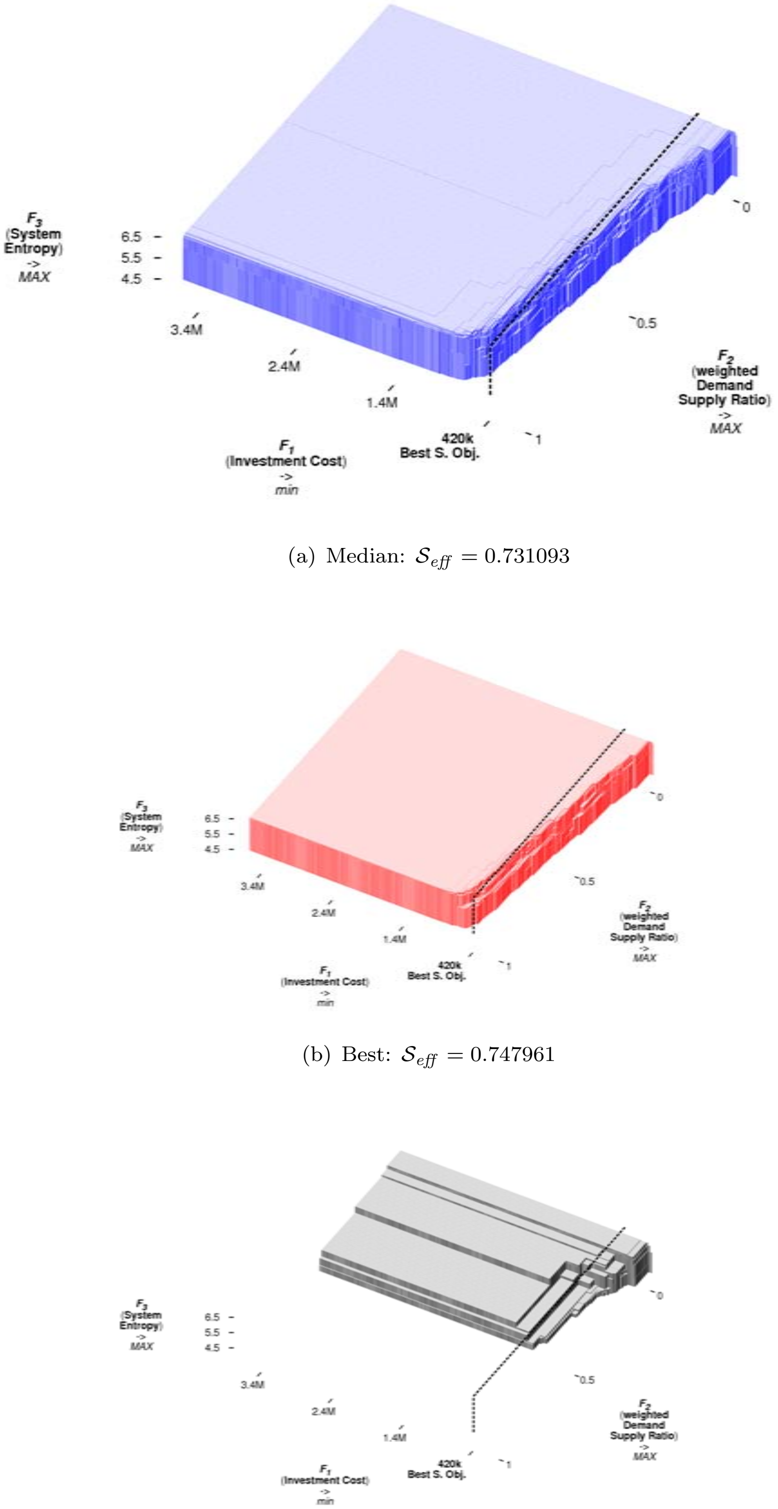


(a) Median: $\mathcal{S}_{eff} = 0.739726$

(b) Best: $\mathcal{S}_{eff} = 0.747797$

(c) Worst: $\mathcal{S}_{eff} = 0.724728$

FIGURE B.2: **Robust – Two Loop – NSGA-II-sa-sp-lambda**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-lambda runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

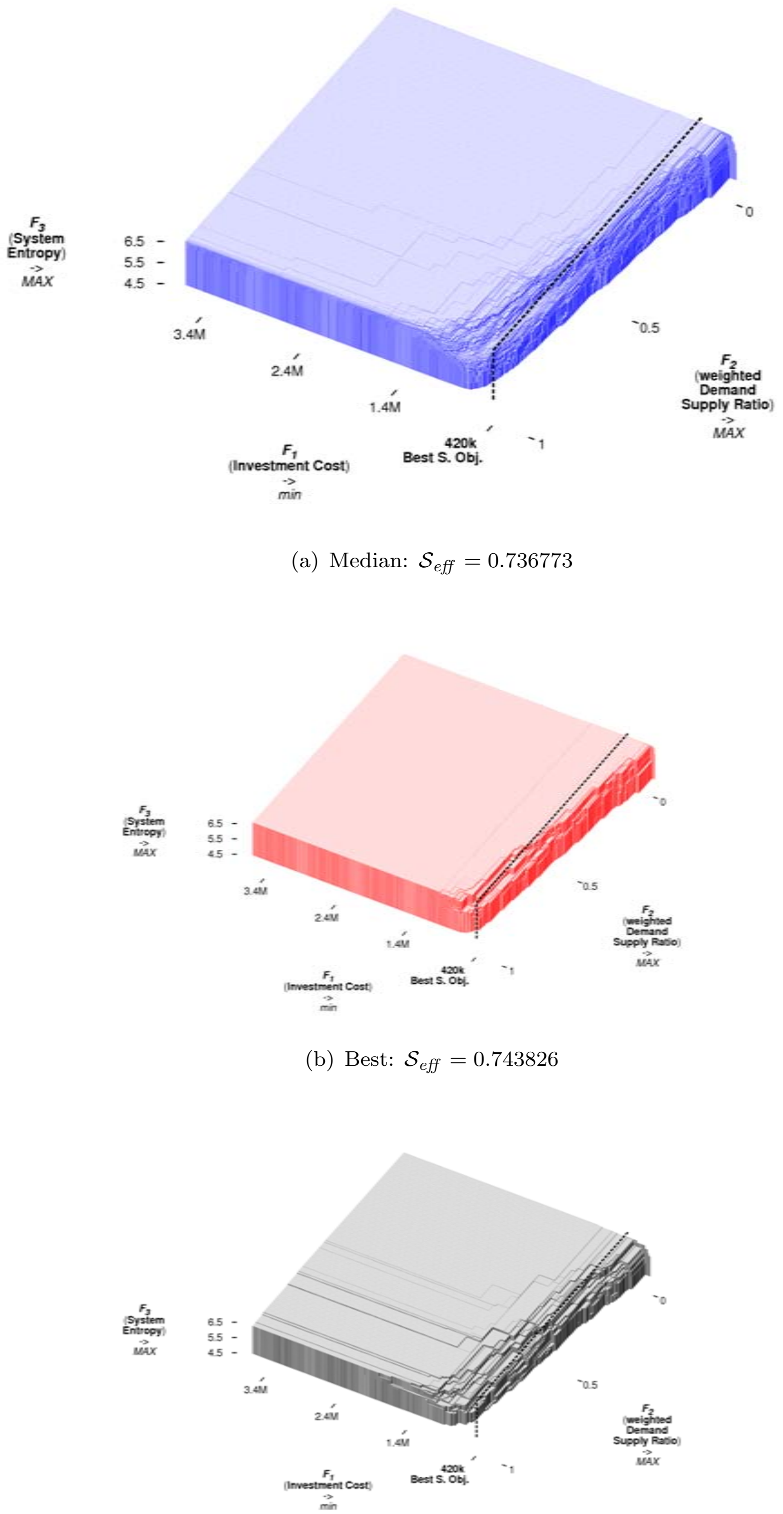


(a) Median: $\mathcal{S}_{eff} = 0.731093$

(b) Best: $\mathcal{S}_{eff} = 0.747961$

(c) Worst: $\mathcal{S}_{eff} = 0.387580$

FIGURE B.3: **Robust – Two Loop – NSGA-II-sa-sp-λ-SEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ-SEM runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

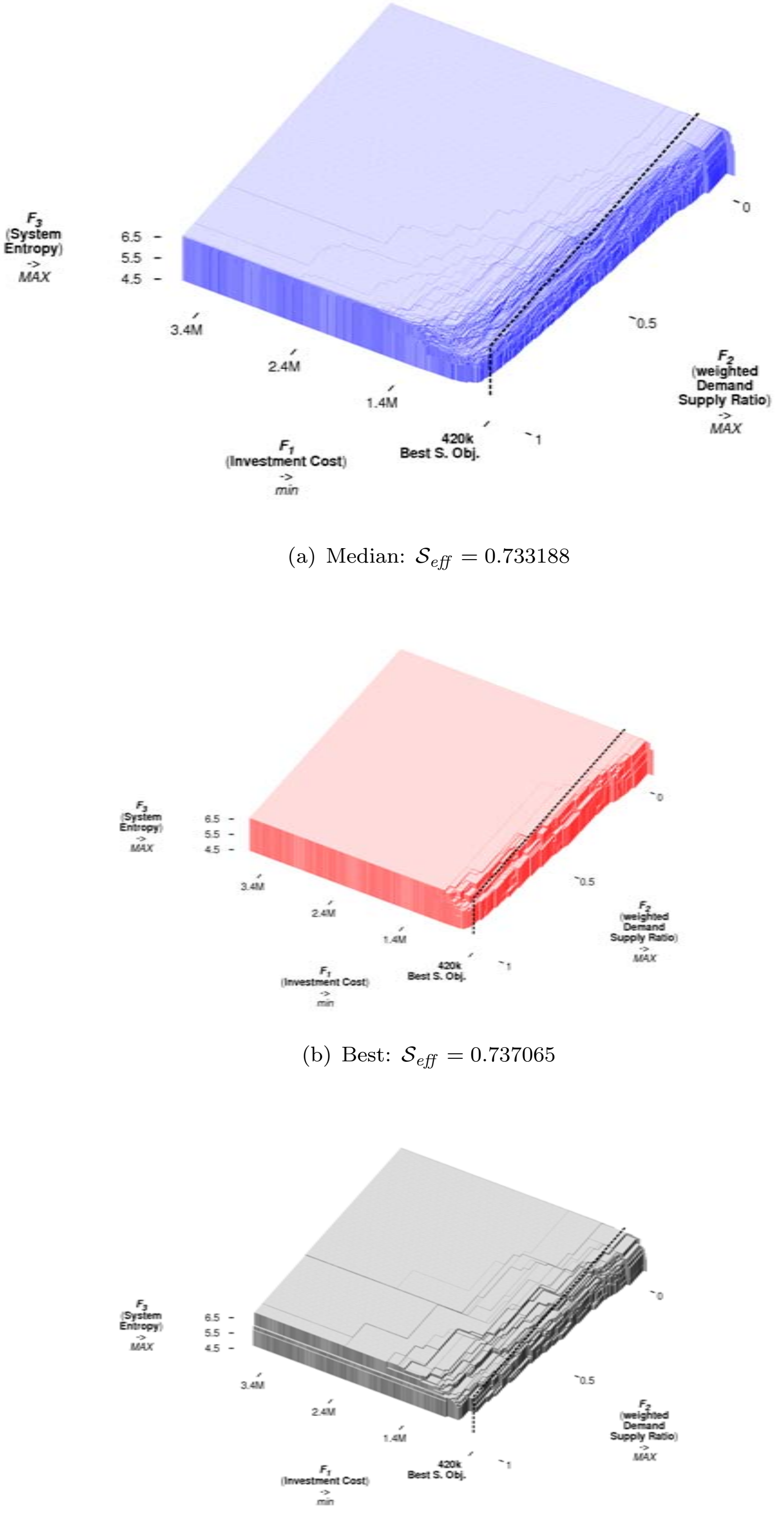


(a) Median: $\mathcal{S}_{eff} = 0.736773$

(b) Best: $\mathcal{S}_{eff} = 0.743826$

(c) Worst: $\mathcal{S}_{eff} = 0.728170$

FIGURE B.4: **Robust – Two Loop – NSGA-II-sa-sp-λ-MEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ-MEM runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

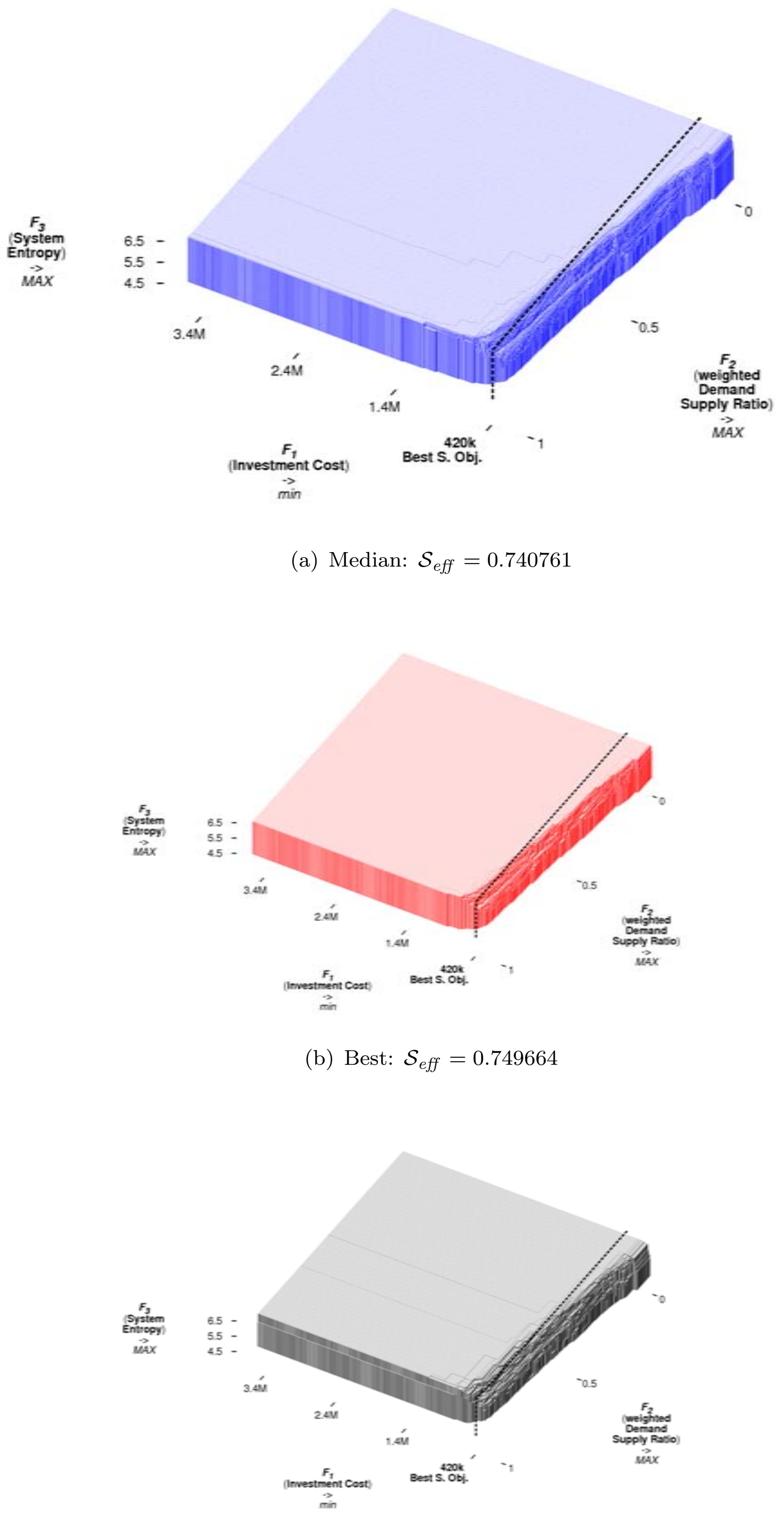


(a) Median: $\mathcal{S}_{\mathit{eff}} = 0.733188$

(b) Best: $\mathcal{S}_{\mathit{eff}} = 0.737065$

(c) Worst: $\mathcal{S}_{\mathit{eff}} = 0.718683$

Figure B.5: **Robust – Two Loop – SMS-EMOA**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

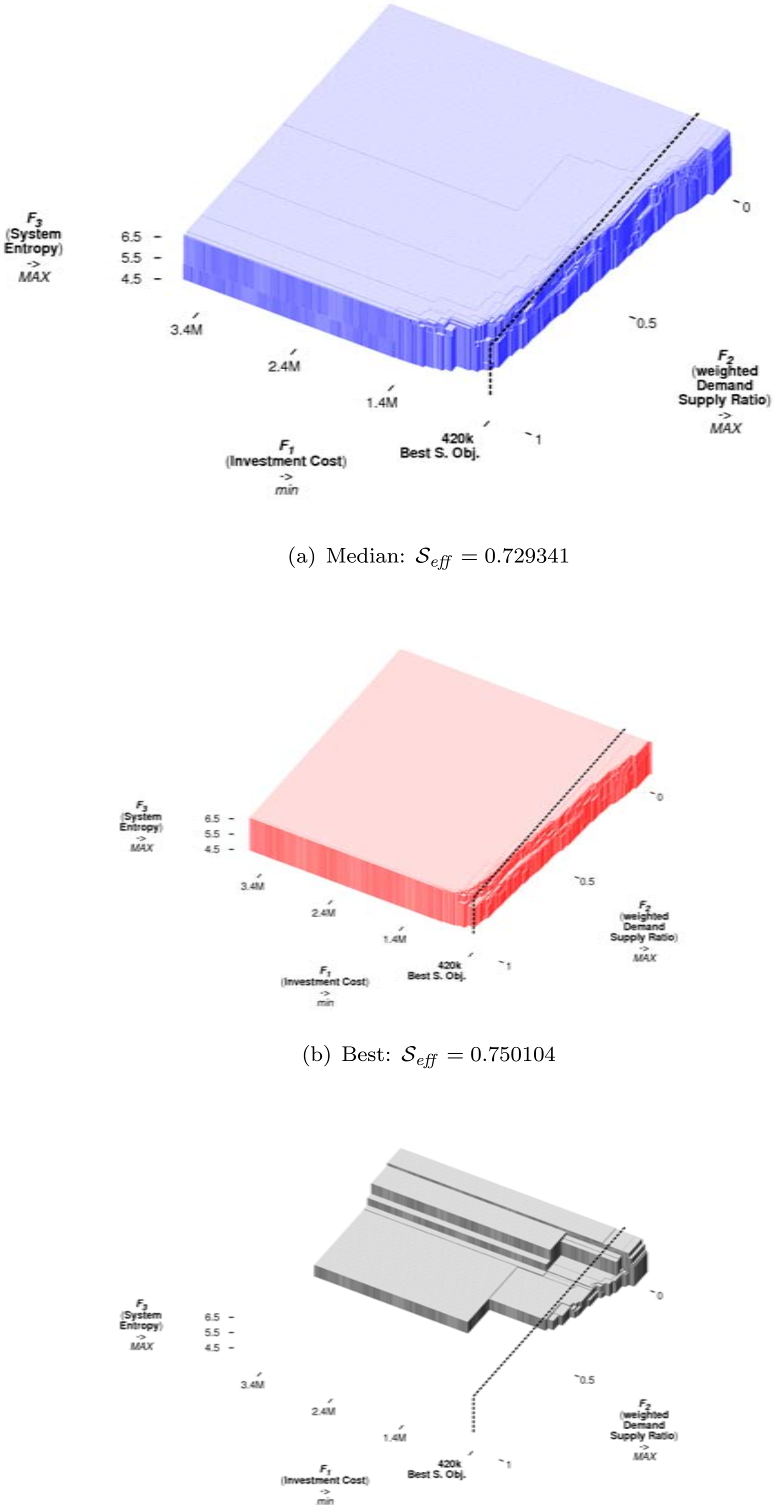


(a) Median: $\mathcal{S}_{eff} = 0.740761$

(b) Best: $\mathcal{S}_{eff} = 0.749664$

(c) Worst: $\mathcal{S}_{eff} = 0.727030$

FIGURE B.6: **Robust – Two Loop – SMS-EMOA-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

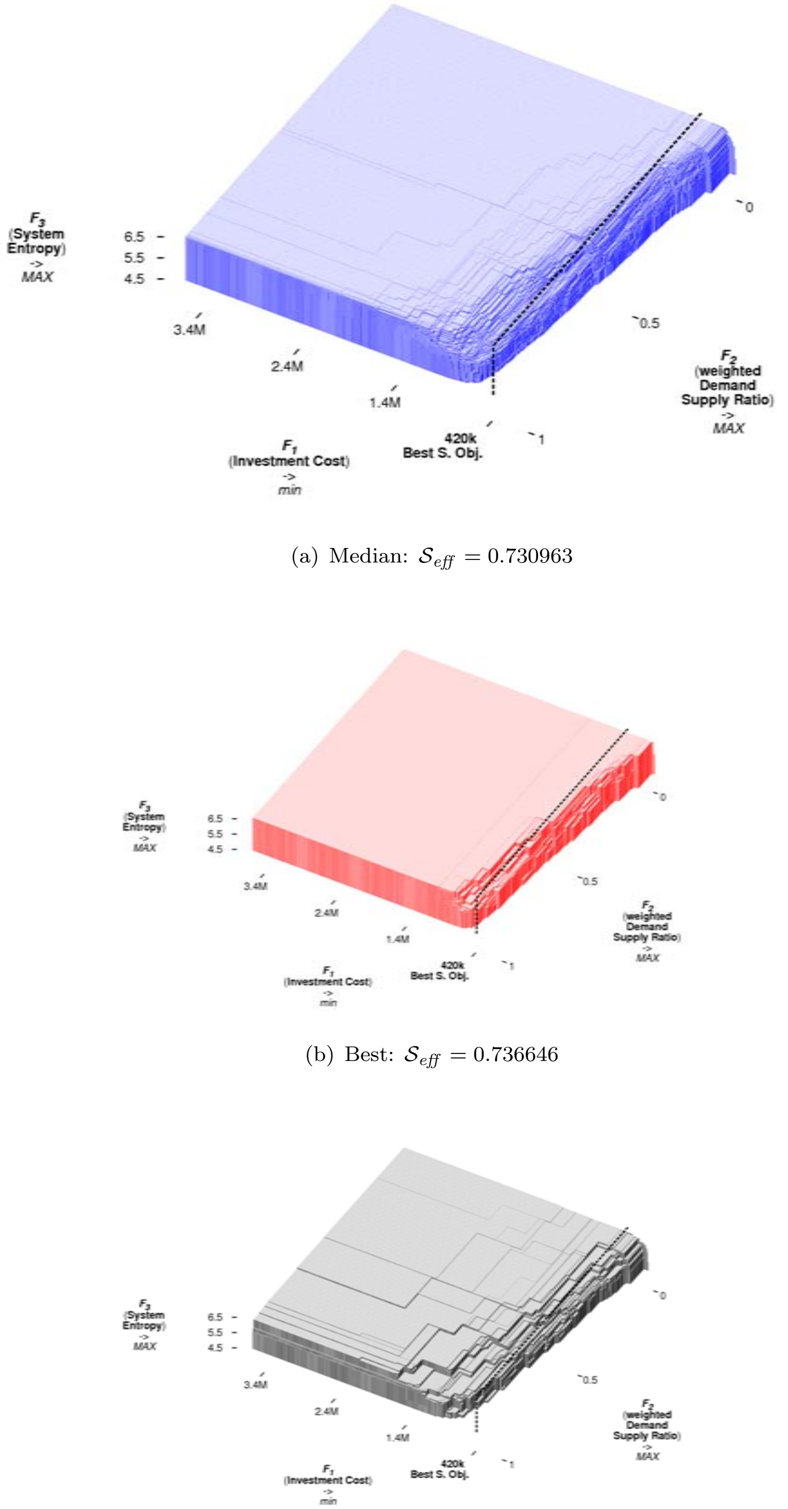


(a) Median: $\mathcal{S}_{\mathit{eff}} = 0.729341$

(b) Best: $\mathcal{S}_{\mathit{eff}} = 0.750104$

(c) Worst: $\mathcal{S}_{\mathit{eff}} = 0.428744$

Figure B.7: **Robust – Two Loop – SMS-EMOA-sa-SEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-SEM runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

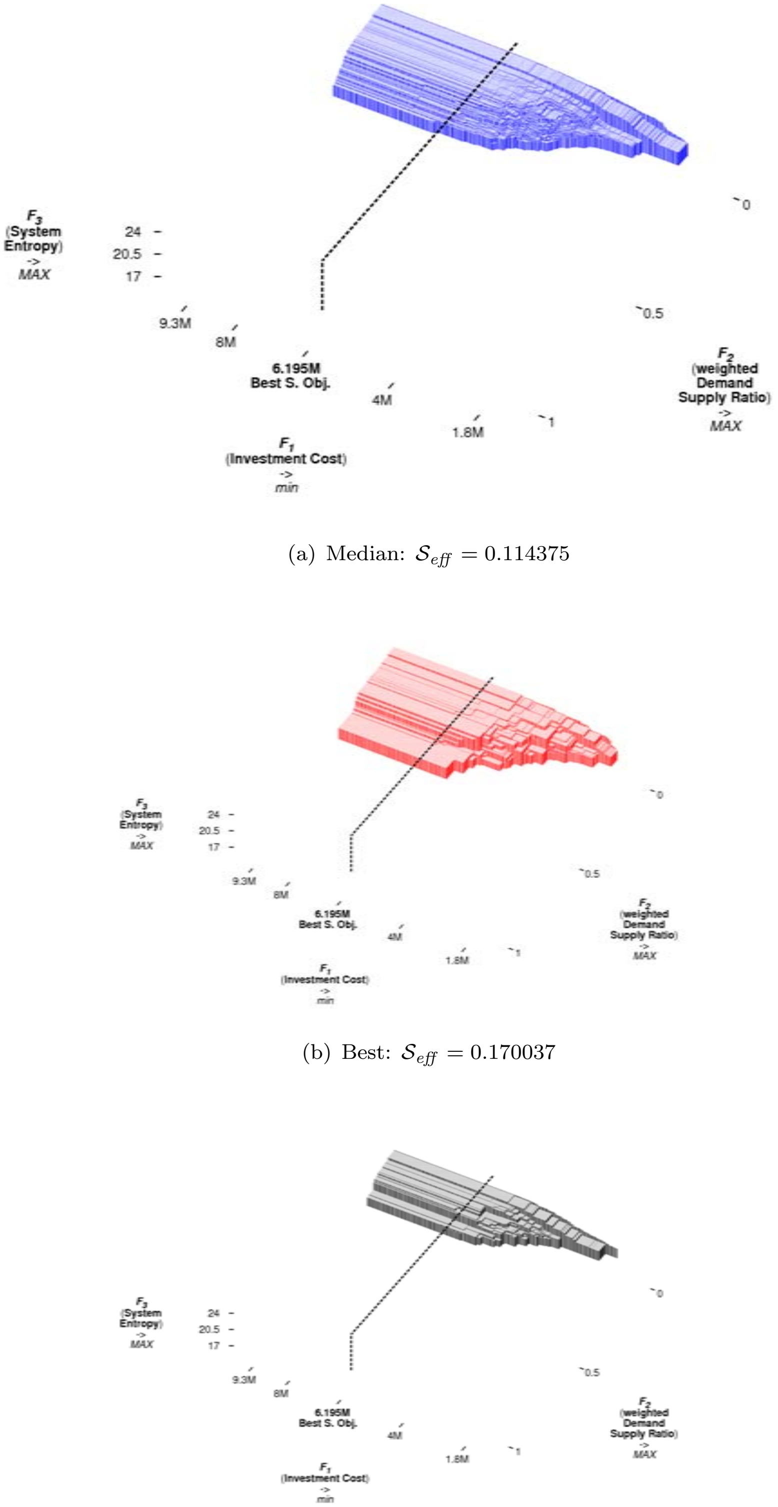


(a) Median: $\mathcal{S}_{eff} = 0.730963$

(b) Best: $\mathcal{S}_{eff} = 0.736646$

(c) Worst: $\mathcal{S}_{eff} = 0.723160$

Figure B.8: **Robust – Two Loop – SMS-EMOA-sa-MEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-MEM runs for test problem Two Loop. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

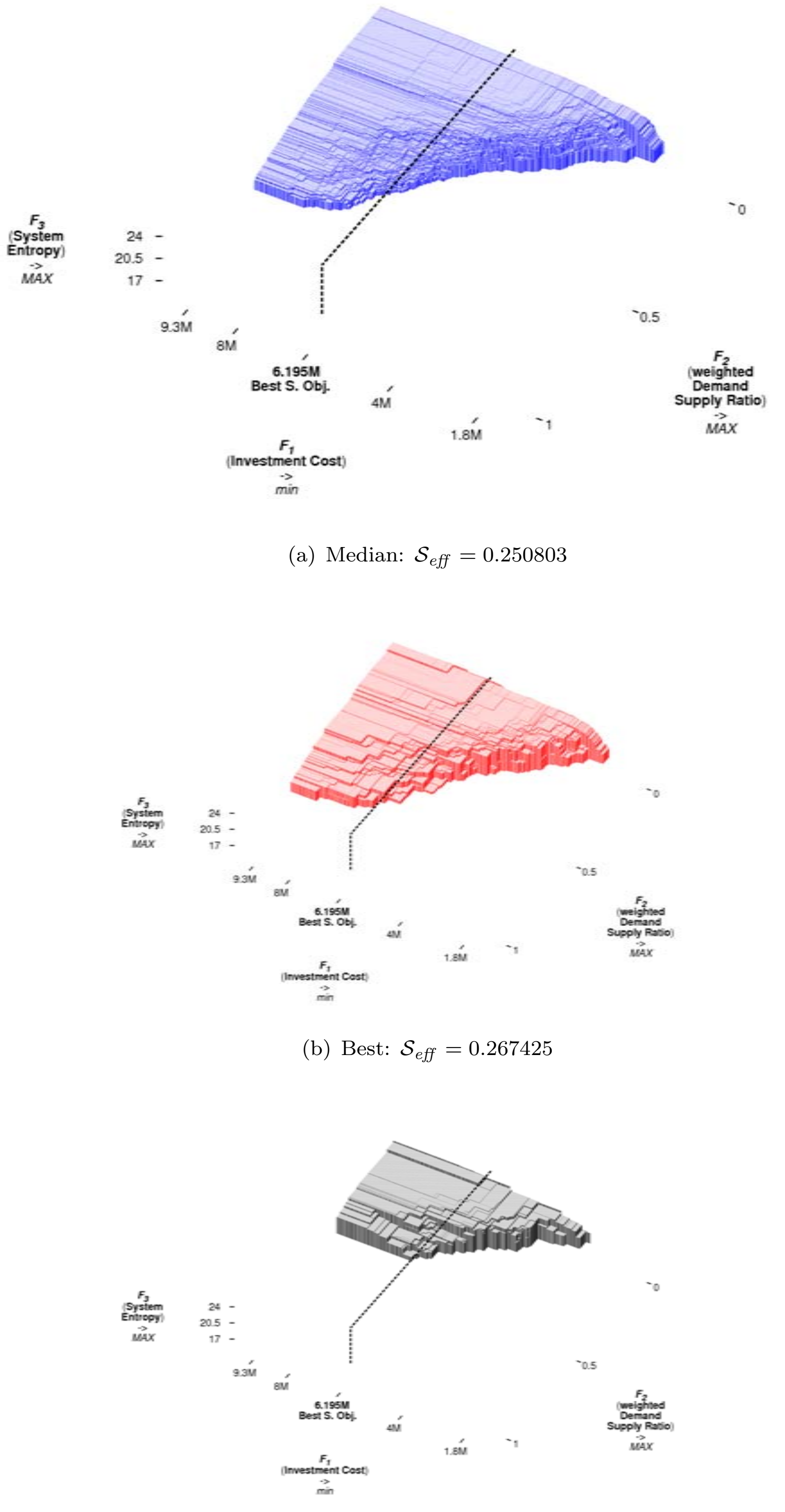


(a) Median: $\mathcal{S}_{\mathit{eff}} = 0.114375$

(b) Best: $\mathcal{S}_{\mathit{eff}} = 0.170037$

(c) Worst: $\mathcal{S}_{\mathit{eff}} = 0.092906$

FIGURE B.9: **Robust – Hanoi – NSGA-II**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

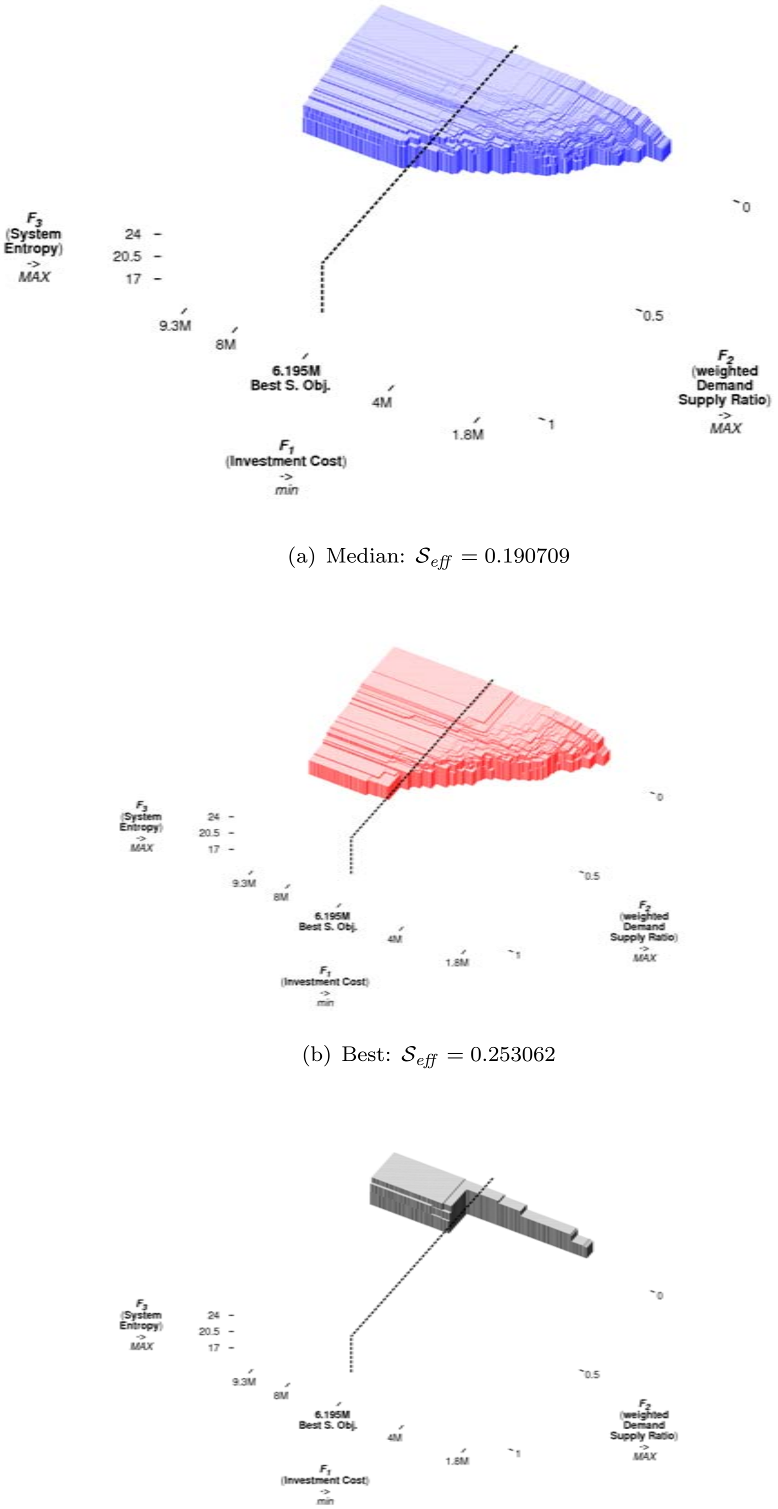


(a) Median: $\mathcal{S}_{eff} = 0.250803$

(b) Best: $\mathcal{S}_{eff} = 0.267425$

(c) Worst: $\mathcal{S}_{eff} = 0.193548$

Figure B.10: **Robust − Hanoi − NSGA-II-sa-sp-lambda**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-lambda runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

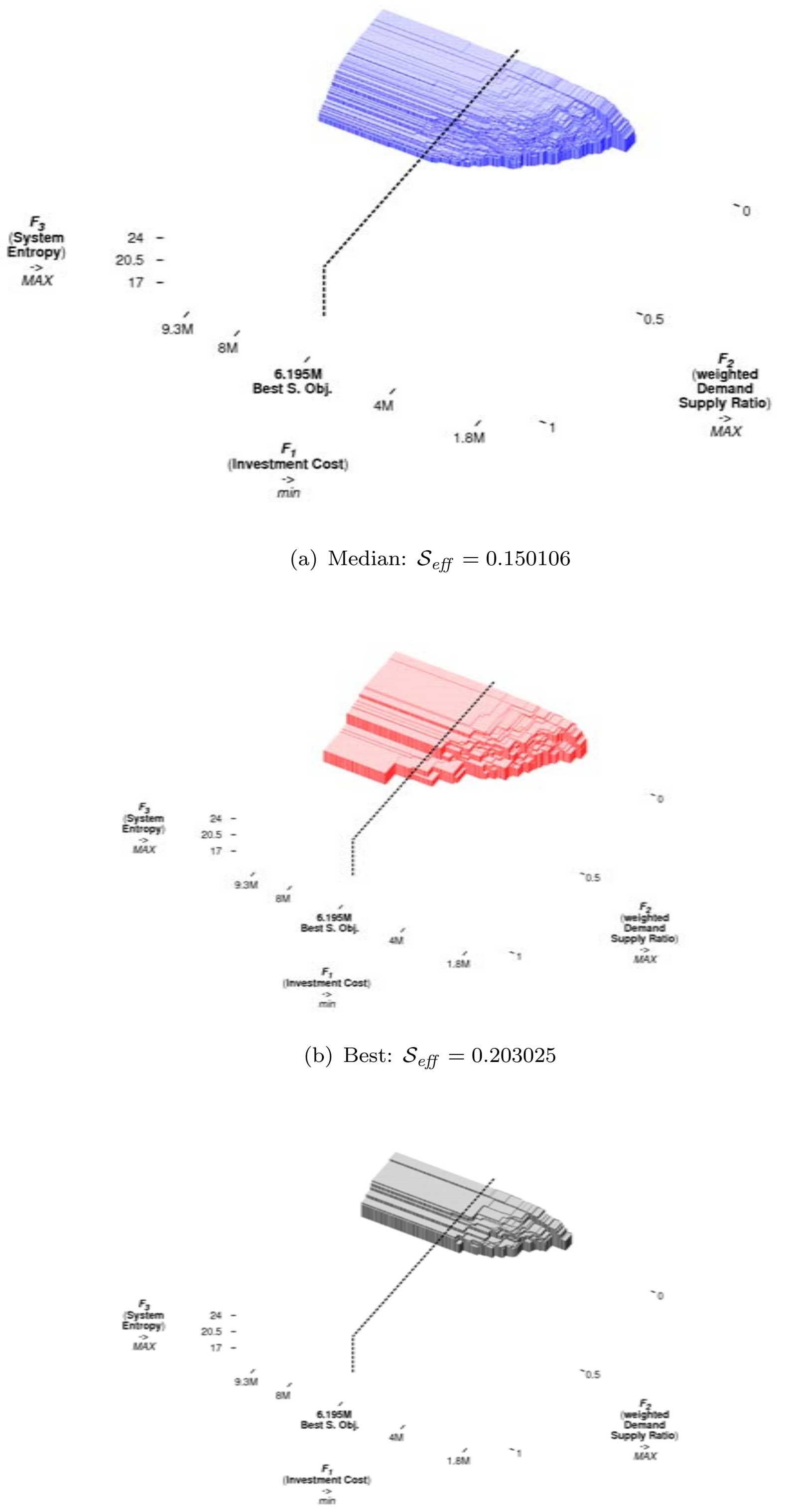


(a) Median: $\mathcal{S}_{eff} = 0.190709$

(b) Best: $\mathcal{S}_{eff} = 0.253062$

(c) Worst: $\mathcal{S}_{eff} = 0.064947$

Figure B.11: **Robust – Hanoi – NSGA-II-sa-sp-λ-SEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ-SEM runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

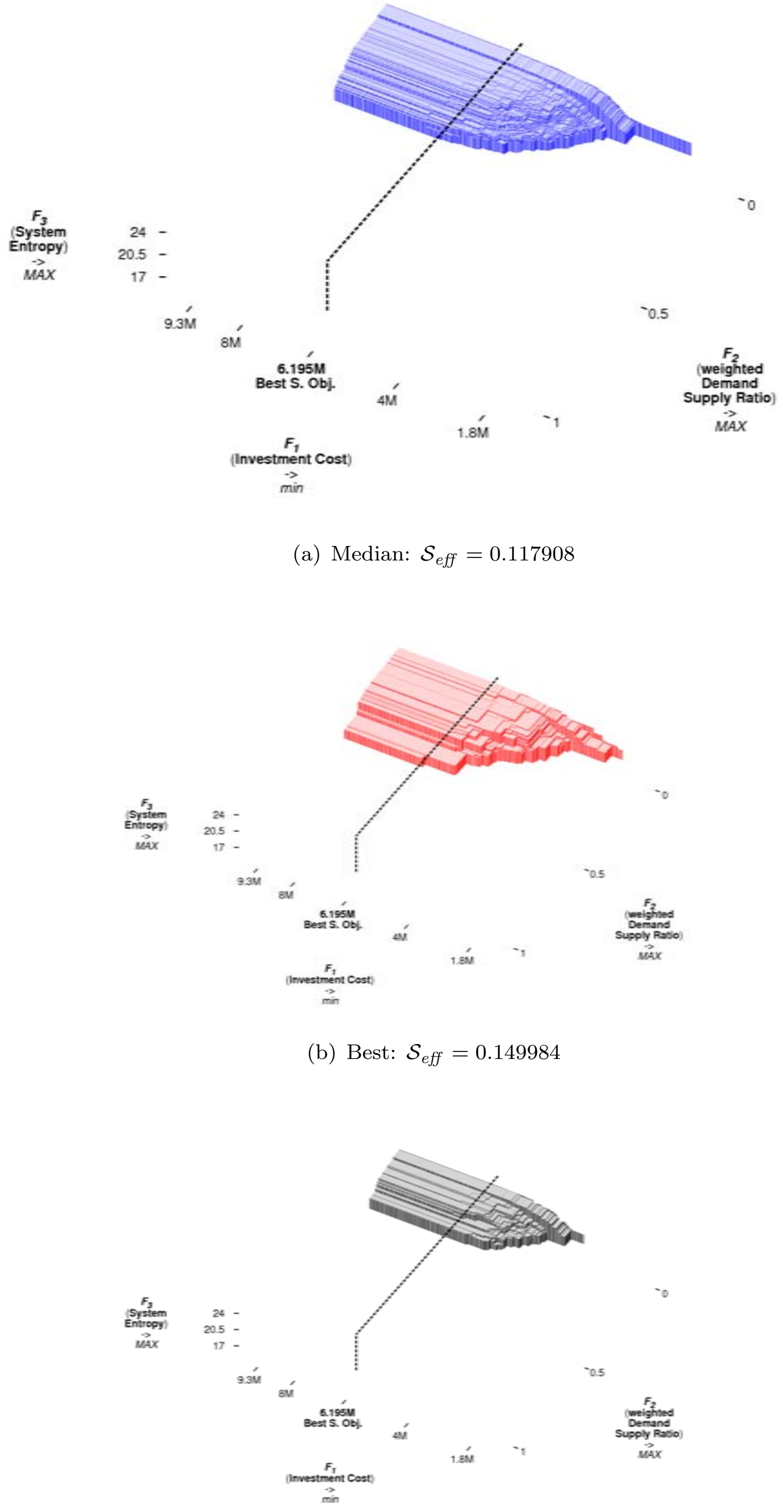


(a) Median: $\mathcal{S}_{eff} = 0.150106$

(b) Best: $\mathcal{S}_{eff} = 0.203025$

(c) Worst: $\mathcal{S}_{eff} = 0.117687$

FIGURE B.12: **Robust – Hanoi – NSGA-II-sa-sp-λ-MEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ-MEM runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

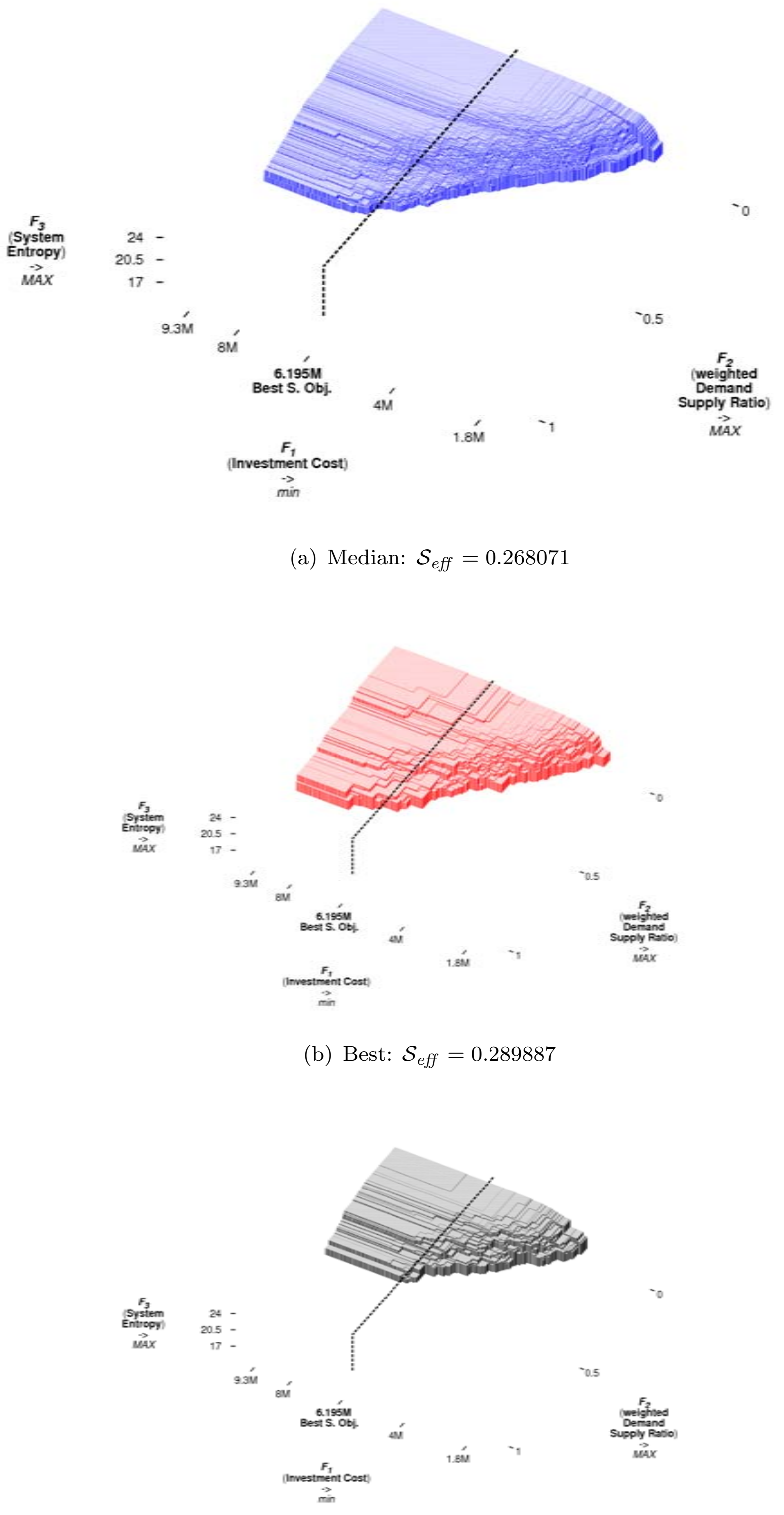


(a) Median: $\mathcal{S}_{eff} = 0.117908$

(b) Best: $\mathcal{S}_{eff} = 0.149984$

(c) Worst: $\mathcal{S}_{eff} = 0.094929$

Figure B.13: **Robust – Hanoi – SMS-EMOA**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

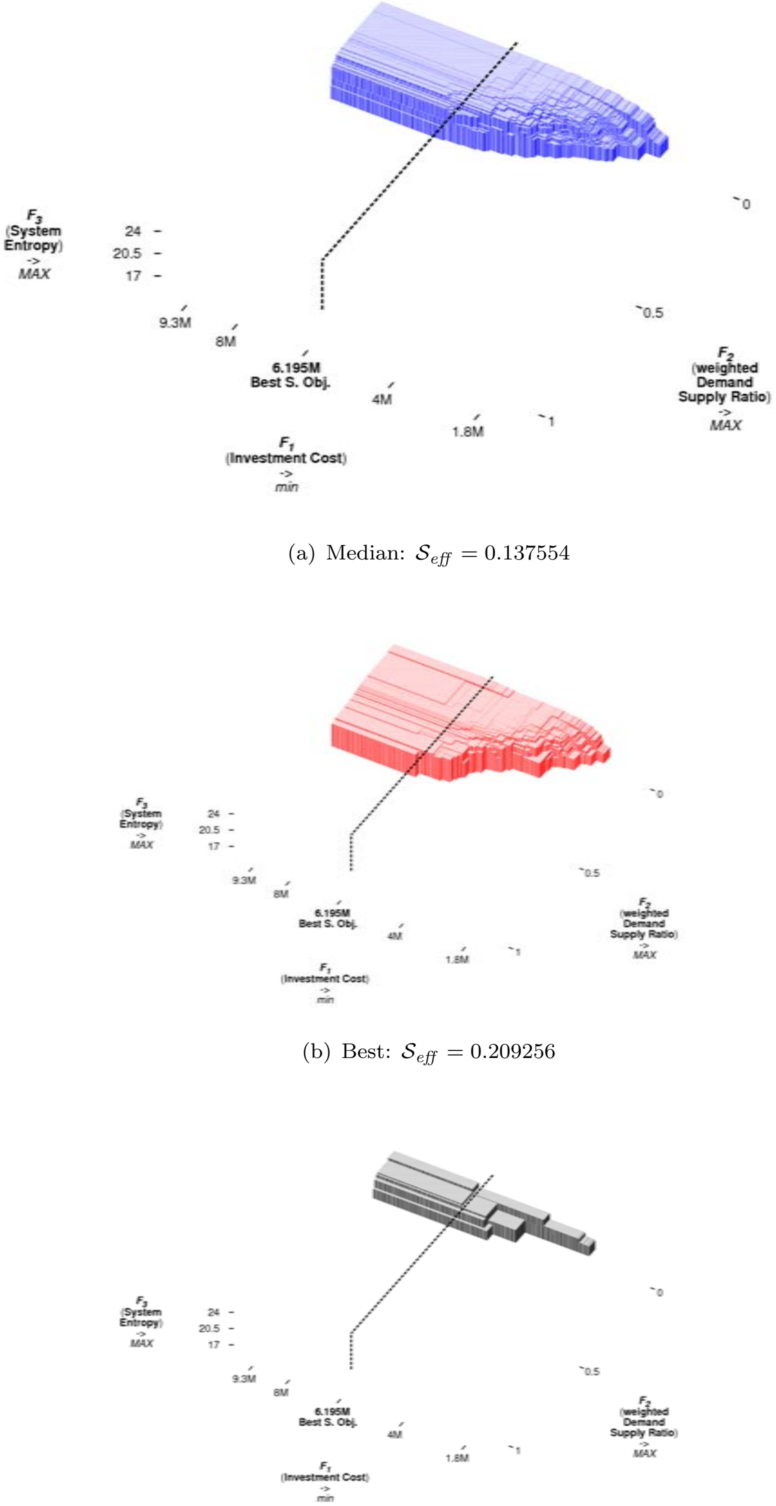


(a) Median: $\mathcal{S}_{\mathit{eff}} = 0.268071$

(b) Best: $\mathcal{S}_{\mathit{eff}} = 0.289887$

(c) Worst: $\mathcal{S}_{\mathit{eff}} = 0.237268$

FIGURE B.14: **Robust – Hanoi – SMS-EMOA-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

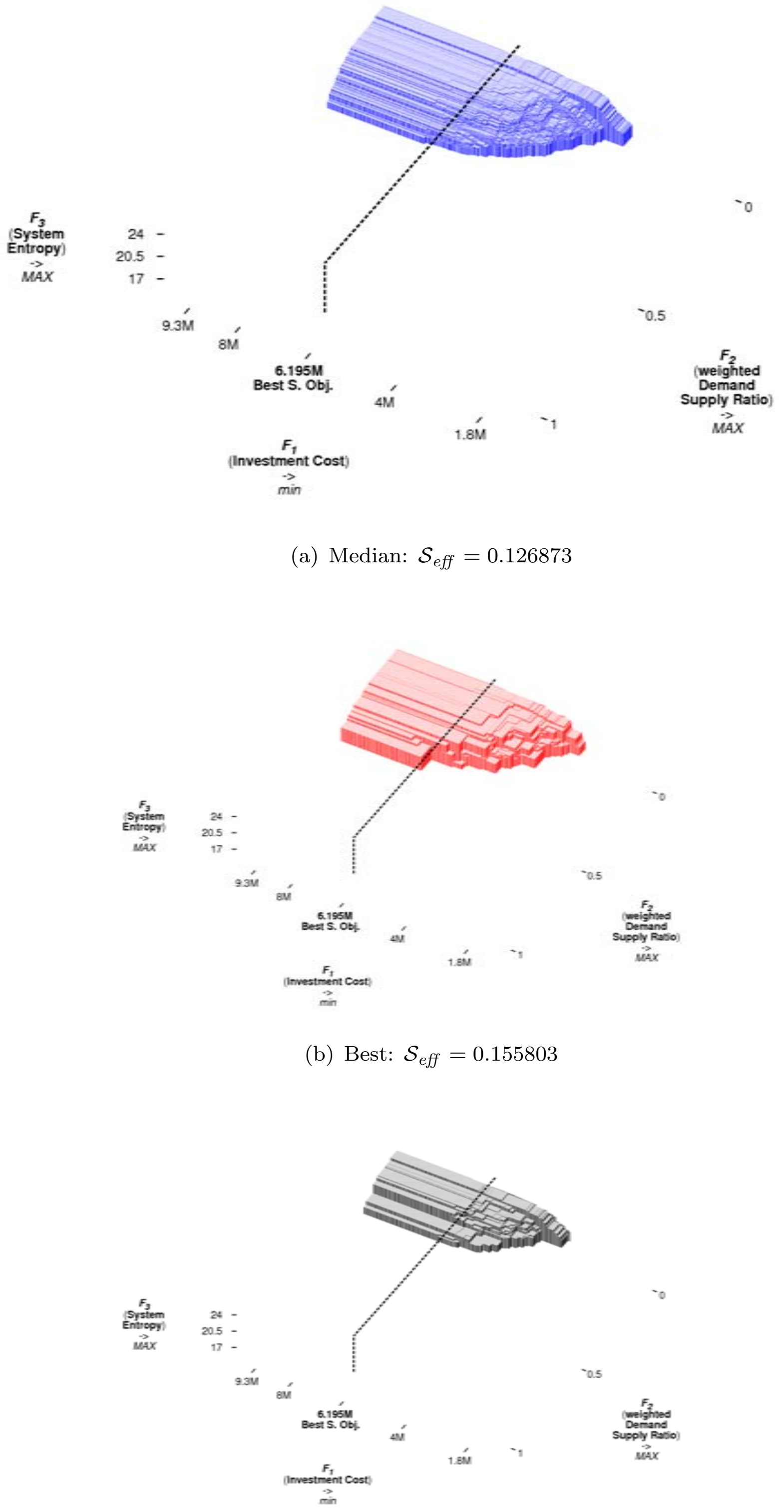


(a) Median: $\mathcal{S}_{eff} = 0.137554$

(b) Best: $\mathcal{S}_{eff} = 0.209256$

(c) Worst: $\mathcal{S}_{eff} = 0.087448$

Figure B.15: **Robust – Hanoi – SMS-EMOA-sa-SEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-SEM runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

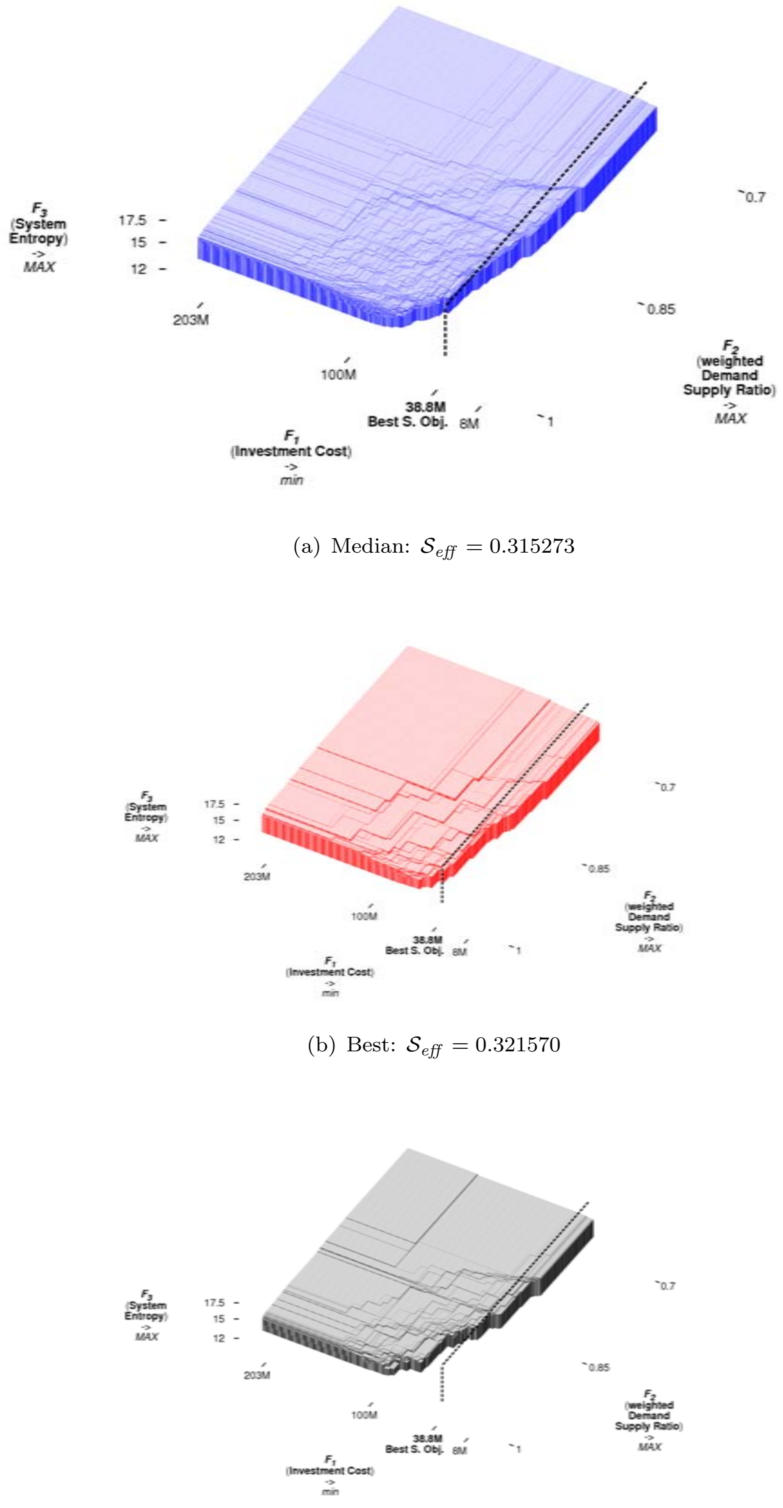


(a) Median: $\mathcal{S}_{eff} = 0.126873$

(b) Best: $\mathcal{S}_{eff} = 0.155803$

(c) Worst: $\mathcal{S}_{eff} = 0.104283$

FIGURE B.16: **Robust – Hanoi – SMS-EMOA-sa-MEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-MEM runs for test problem Hanoi. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

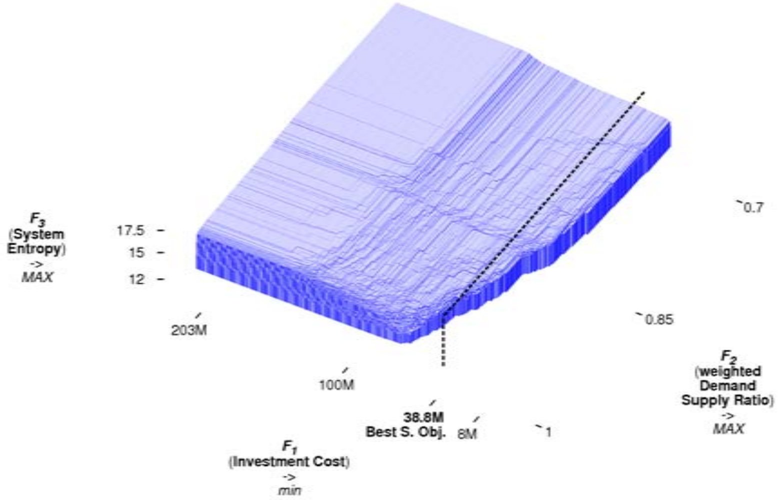


(a) Median: $\mathcal{S}_{eff} = 0.315273$

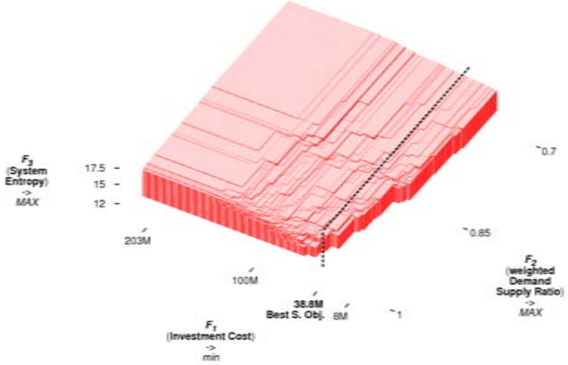
(b) Best: $\mathcal{S}_{eff} = 0.321570$

(c) Worst: $\mathcal{S}_{eff} = 0.311470$

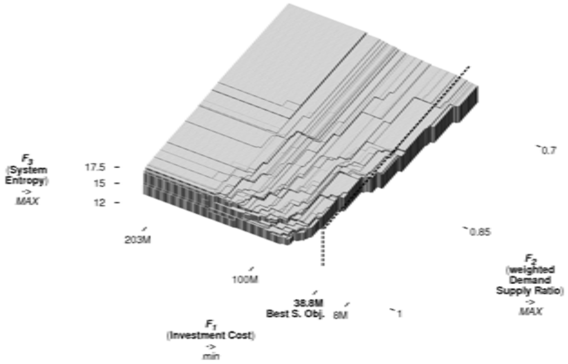
Figure B.17: **Robust – New York City – NSGA-II**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.



(a) Median: $\mathcal{S}_{eff} = 0.352729$

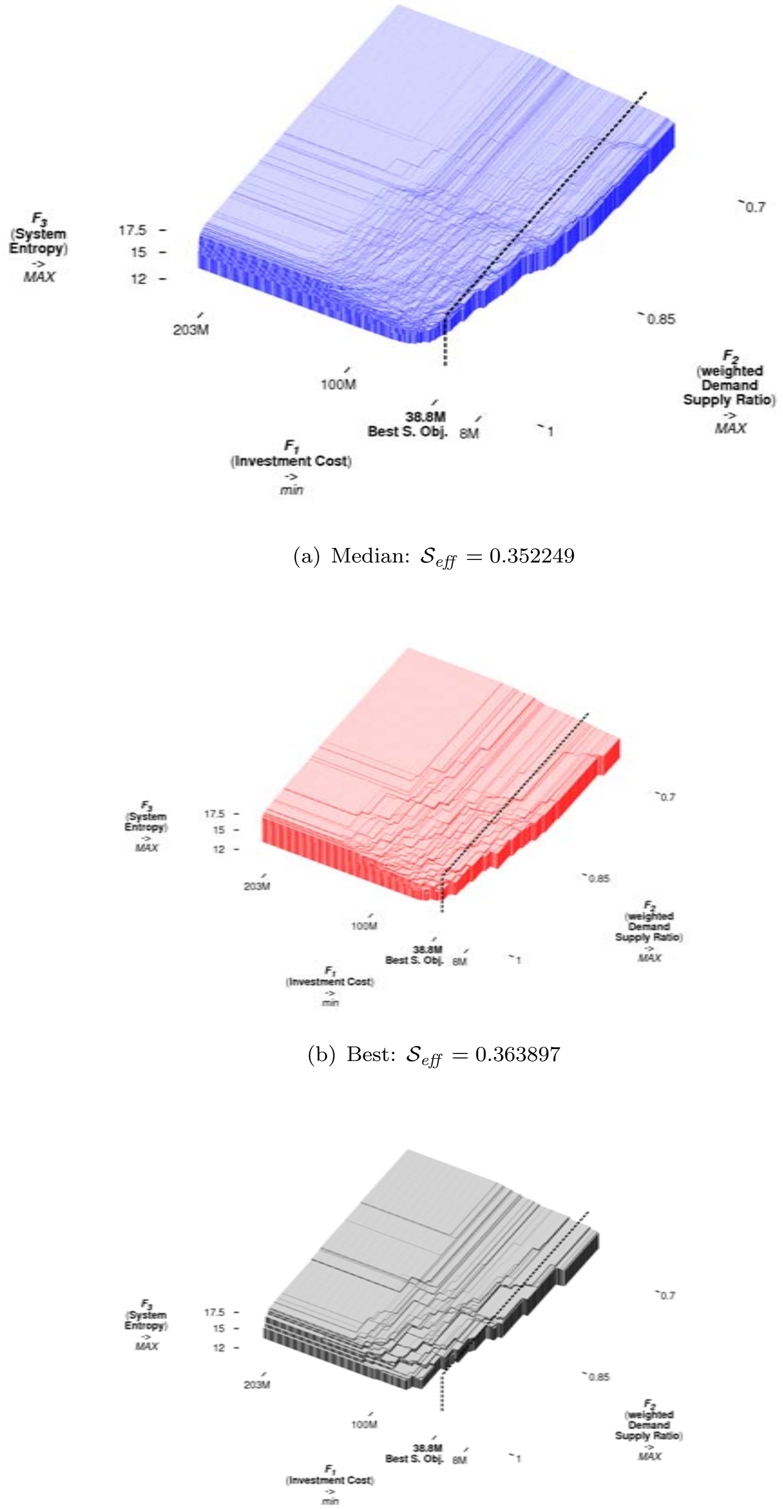


(b) Best: $\mathcal{S}_{eff} = 0.363401$



(c) Worst: $\mathcal{S}_{eff} = 0.341007$

FIGURE B.18: **Robust – New York City – NSGA-II-sa-sp-lambda**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-lambda runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

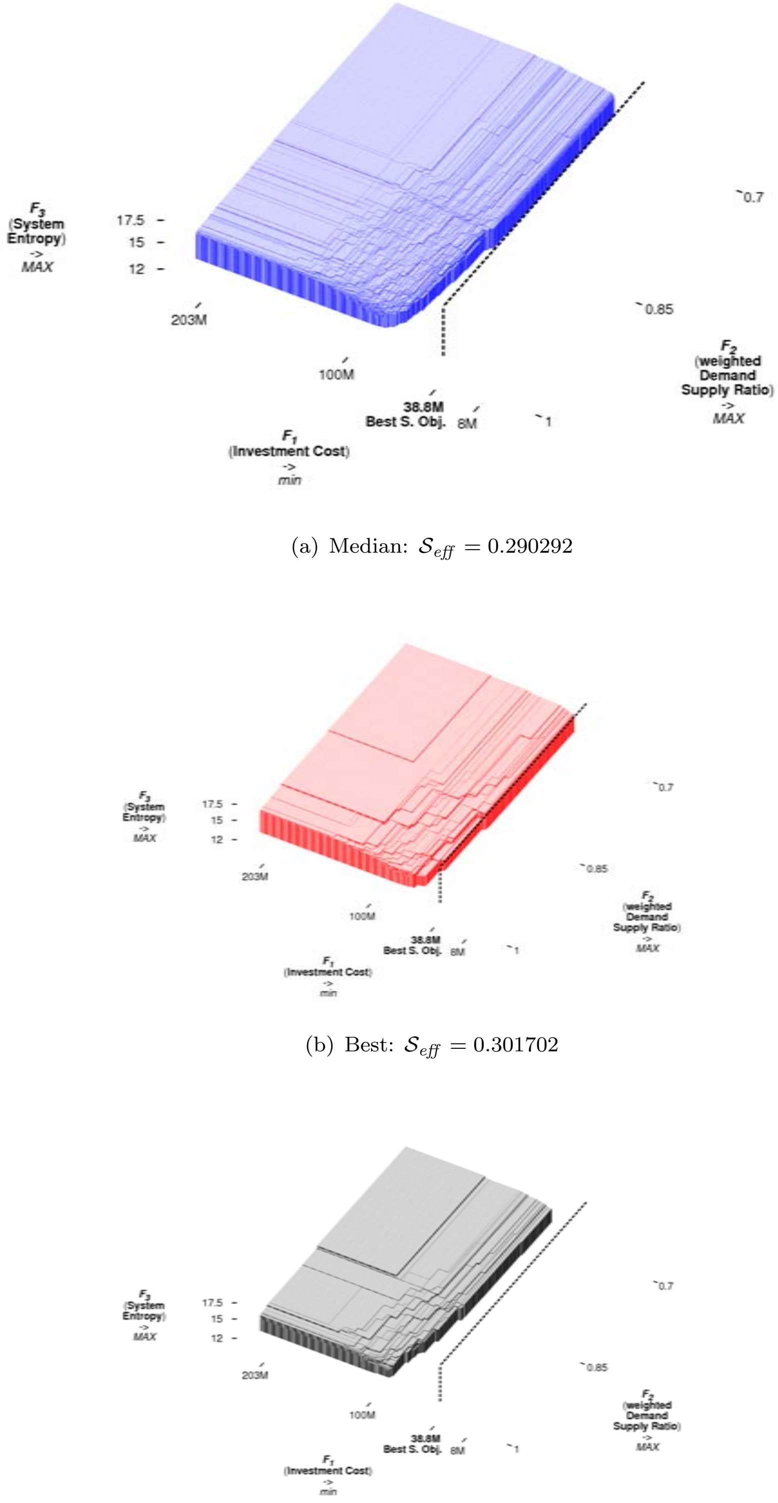


(a) Median: $\mathcal{S}_{eff} = 0.352249$

(b) Best: $\mathcal{S}_{eff} = 0.363897$

(c) Worst: $\mathcal{S}_{eff} = 0.342408$

FIGURE B.19: **Robust – New York City – NSGA-II-sa-sp-λ-SEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ-SEM runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

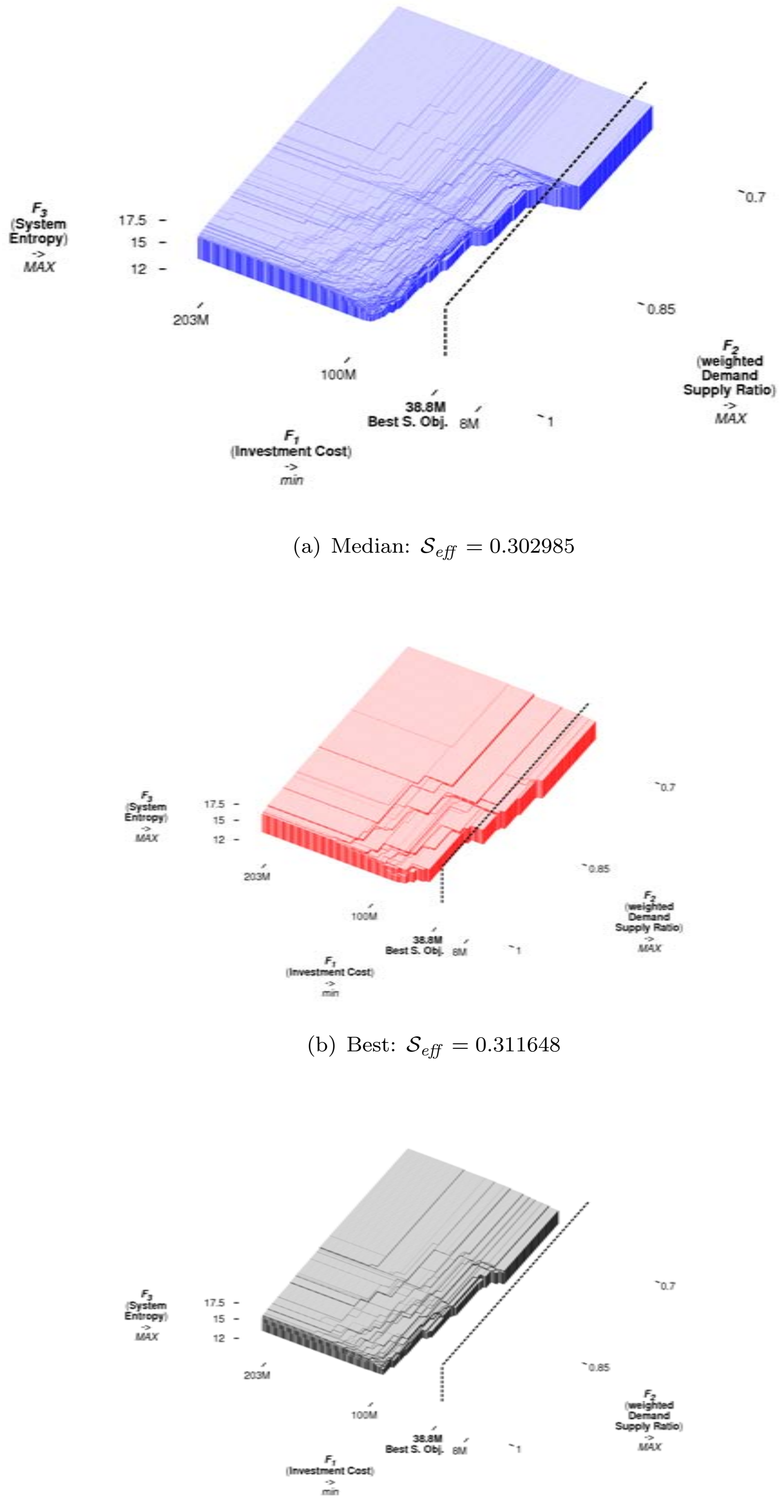


(a) Median: $\mathcal{S}_{eff} = 0.290292$

(b) Best: $\mathcal{S}_{eff} = 0.301702$

(c) Worst: $\mathcal{S}_{eff} = 0.276309$

Figure B.20: **Robust – New York City – NSGA-II-sa-sp-λ-MEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the NSGA-II-sa-sp-λ-MEM runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

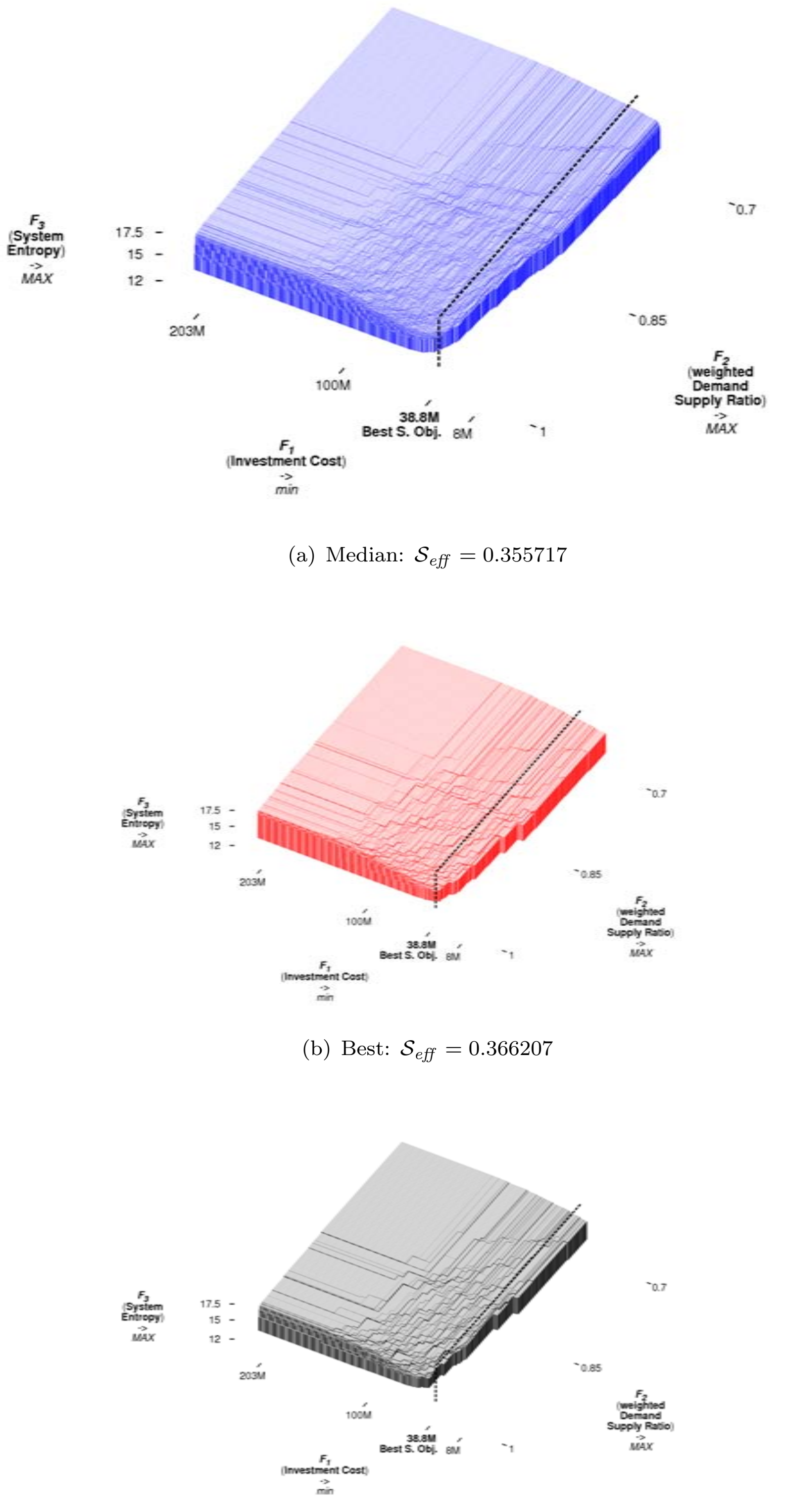


(a) Median: $\mathcal{S}_{eff} = 0.302985$

(b) Best: $\mathcal{S}_{eff} = 0.311648$

(c) Worst: $\mathcal{S}_{eff} = 0.272257$

Figure B.21: **Robust – New York City – SMS-EMOA**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

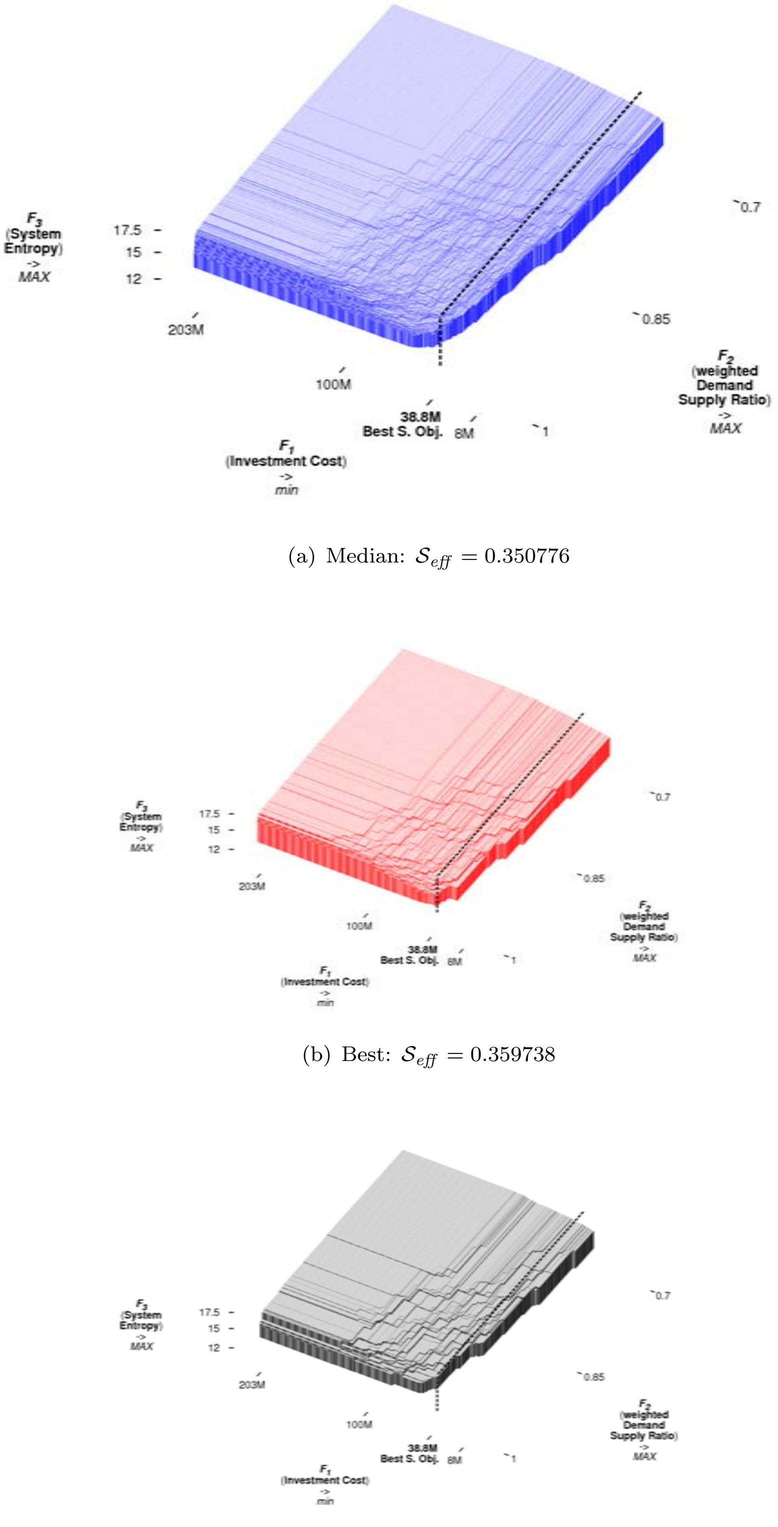


(a) Median: $\mathcal{S}_{\mathit{eff}} = 0.355717$

(b) Best: $\mathcal{S}_{\mathit{eff}} = 0.366207$

(c) Worst: $\mathcal{S}_{\mathit{eff}} = 0.343320$

FIGURE B.22: **Robust – New York City – SMS-EMOA-sa**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

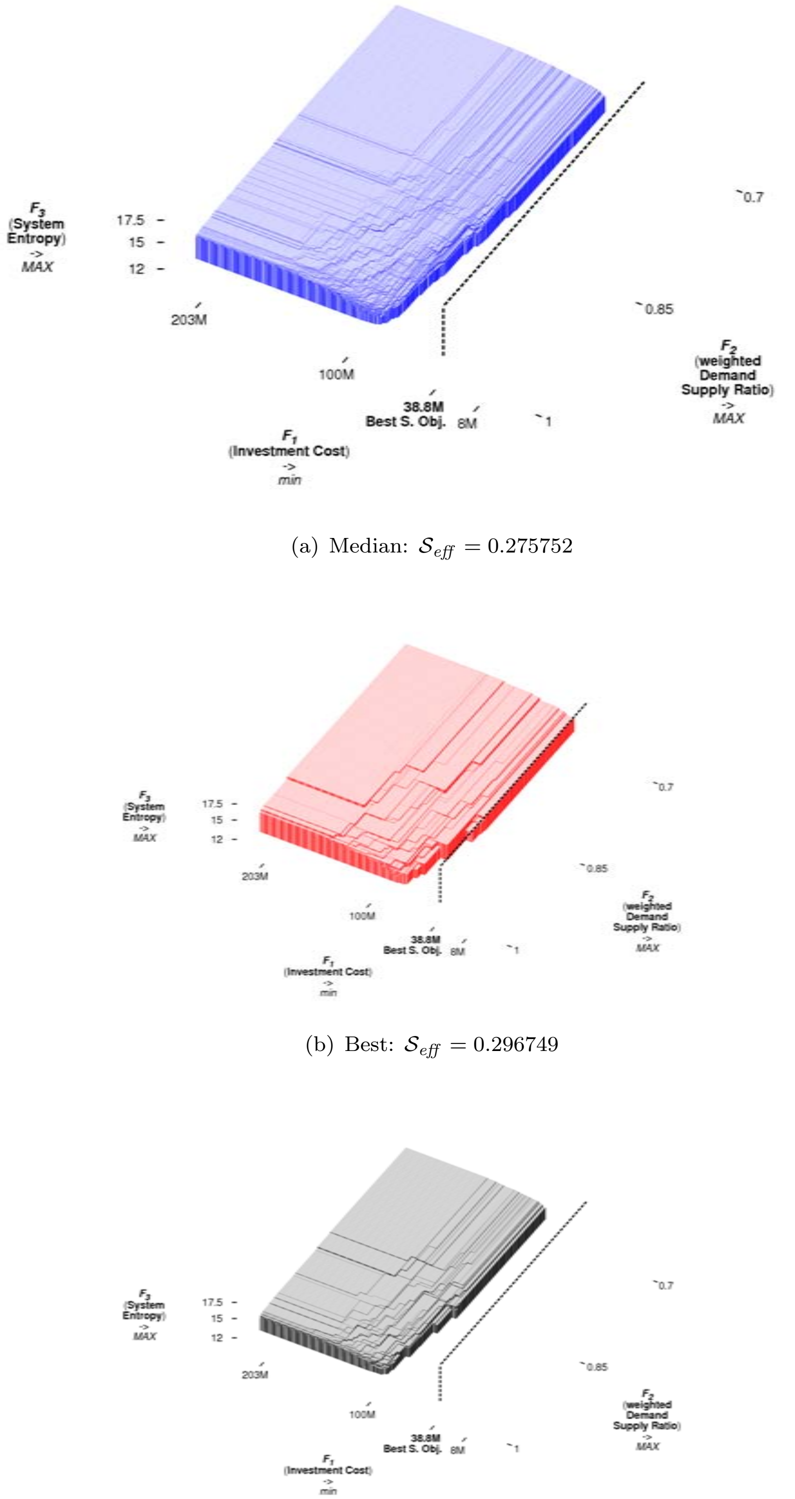


(a) Median: $\mathcal{S}_{\mathit{eff}} = 0.350776$

(b) Best: $\mathcal{S}_{\mathit{eff}} = 0.359738$

(c) Worst: $\mathcal{S}_{\mathit{eff}} = 0.344337$

Figure B.23: **Robust – New York City – SMS-EMOA-sa-SEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-SEM runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.



(a) Median: $\mathcal{S}_{eff} = 0.275752$

(b) Best: $\mathcal{S}_{eff} = 0.296749$

(c) Worst: $\mathcal{S}_{eff} = 0.261244$

FIGURE B.24: **Robust – New York City – SMS-EMOA-sa-MEM**
Median, best, and worst attainment surfaces are displayed of the Pareto front approximations found by the SMS-EMOA-sa-MEM runs for test problem New York City. The Pareto front approximations are determined by the effective fitness of the solutions per result set, and labeled with the $\mathcal{S}$ metric score based on effective fitness. The robust objective function values (i.e., the effective fitness) have been approximated using Monte Carlo integration with a sample set size of $N^{sample} = 2000$. The best single objective solution known is indicated by its Investment Cost.

Bibliography

[1] A. Abebe and D. Solomatine. Application of Global Optimization to the Design of Pipe Networks. In *Proceedings of the 3rd International Conference on Hydroinformatics*, pages 989–996. Balkema, 1998.

[2] E. Alperovits and U. Shamir. Design of Optimal Water Distribution Systems. *Water Resources Research*, 13(6):885–900, 1977.

[3] C. Arsene, A. Bargiela, and D. Al-Dabass. Modelling and Simulation of Water Systems Based on Loop Equations. *International Journal of Simulation*, 5(1-2):61–72, 2002.

[4] M. Atiquzzaman, S.-Y. Liong, and X. Yu. Alternative Decision Making in Water Distribution Network with NSGA-II. *Journal of Water Resources Planning and Management*, 132(2):122–126, 2006.

[5] A. Babayan, Z. Kapelan, D. Savic, and G. Walters. Comparison of two methods for the stochastic least cost design of water distribution systems. *Engineering Optimization*, 38(3):281–297, 2006.

[6] A. Babayan, D. Savic, G. Walters, and Z. Kapelan. Robust Least-Cost Design of Water Distribution Networks Using Redundancy and Integration-Based Methodologies. *Journal of Water Resources Planning and Management*, 133(1):67–77, 2007.

[7] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, Oxford, UK, 1996.

[8] J. Benjamin and C. Cornell. *Probability, Statistics, and Decision for Civil Engineers.* McGraw-Hill, New York, NY, USA, 1970.

[9] N. Beume. S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem. *Evolutionary Computation*, 17(4):477–492, 2009.

[10] N. Beume, C. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold. On the Complexity of Computing the Hypervolume Indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082, 2009.

[11] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[12] H.-G. Beyer and B. Sendhoff. Evolution Strategies for Robust Optimization. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1346–1353. IEEE Computer Society, 2006.

[13] P. Bhave and R. Gupta. Optimal design of water distribution networks for fuzzy demands. *Civil Engineering and Environmental Systems*, 21(4):229–245, 2004.

[14] L. Bradstreet, L. Barone, and L. While. Maximising Hypervolume for Selection in Multi-objective Evolutionary Algorithms. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1744–1751. IEEE Computer Society, 2006.

[15] L. Bradstreet, L. While, and L. Barone. A Fast Incremental Hypervolume Algorithm. *IEEE Transactions on Evolutionary Computation*, 12(6):714–723, 2008.

[16] J. Branke. Reducing the Sampling Variance when Searching for Robust Solutions. In L. Spector et al., editor, *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, GECCO'01*, pages 235–242. Morgan Kaufmann, 2001.

[17] D. Broad, G. Dandy, and H. Maier. Water Distribution System Optimization Using Metamodels. *Journal of Water Resources Planning and Management*, 131(3):172–180, 2005.

[18] P. Cheung, L. Reis, K. Formiga, F. Chaudhry, and W. Ticona. Multiobjective Evolutionary Algorithms Applied to the Rehabilitation of a Water Distribution System: A Comparative Study. In C. Fonseca et al., editor, *Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003, LNCS 2632*, pages 67–81. Springer, 2003.

[19] P. Cheung, J. van Zyl, and L. Reis. Extension of EPANET for Pressure Driven Demand Modeling in Water Distribution System. In D. Savic, G. Walters, S. Khu, and R. King, editors, *Proceedings of the 8th International Conference on Computing and Control in the Water Industry 2005, CCWI 2005*, pages 311–316, 2005.

[20] C. Coello Coello, G. Lamont, and D. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[21] M. da Conceicao Cunha and J. Sousa. Water Distribution Network Design Optimization: Simulated Annealing Approach. *Journal of Water Resources Planning and Management*, 125(4):215–221, 1999.

[22] G. Dandy, A. Simpson, and L. Murphy. An improved algorithm for pipe network optimization. *Water Resources Research*, 32(2):449–458, 1996.

[23] R. de Neufville, J. Schaake, and J. Stafford. Systems Analysis of Water Distribution Networks. *Journal of the Sanitary Engineering Division*, 97(6):825–842, 1971.

[24] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[25] K. Deb and H. Gupta. Introducing Robustness in Multi-Objective Optimization. *Evolutionary Computation*, 14(4):463–494, 2006.

[26] K. Deb, M. Mohan, and S. Mishra. Evaluating the ϵ-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation*, 13(4):501–525, 2005.

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

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

[29] M. Emmerich. *Single- and Multi-objective Evolutionary Design Optimization Assisted by Gaussian Random Field Metamodels*. PhD Thesis, Department of Computer Science, TU Dortmund University, Germany, 2005.

[30] M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In C. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Proceedings of the 3rd International Conference Evolutionary Multi-Criterion Optimization, EMO 2005, LNCS 3410*, pages 62–76. Springer, 2005.

[31] M. Emmerich, A. Deutz, R. Li, and J. Kruisselbrink. Getting Lost or Getting Trapped: On the Effect of Moves to Incomparable Points in Multiobjective Hillclimbing. In J. Branke et al., editor, *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO'10*. ACM Press, to appear.

[32] M. Emmerich, K. Giannakoglou, and B. Naujoks. Single- and Multi-objective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

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

[34] R. Farmani, D. Savic, and G. Walters. "EXNET" Benchmark Problem for Multi-Objective Optimization of Large Water Systems. In *IFAC Workshop on Modelling and Control for Participatory Planning and Managing Water Systems*, 2004.

[35] R. Farmani, D. Savic, and G. Walters. Evolutionary multi-objective optimization in water distribution network design. *Engineering Optimization*, 37(2):167–183, 2005.

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

[37] R. Farmani, G. Walters, and D. Savic. 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, 2006.

[38] Y. Filion. Single-objective deterministic versus multi-objective stochastic water network design: practical considerations for the water industry. *Proc IMechE, Part O: Journal of Risk and Reliability*, 222(4):655–665, 2008.

[39] M. Fleischer. The Measure of Pareto Optima: Applications to Multi-objective Metaheuristics. In C. Fonseca et al., editor, *Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003, LNCS 2632*, pages 519–533. Springer, 2003.

[40] C. Fonseca and P. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann, 1993.

[41] C. Fonseca and P. Fleming. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In H.-M. Voigt, W. Ebeling, I. Rechenberg, , and H.-P. Schwefel, editors, *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, LNCS 1141*, pages 584–593. Springer, 1996.

[42] C. Fonseca, L. Paquete, and Manuel López-Ibáñez. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1157–1163. IEEE Computer Society, 2006.

[43] K. Formiga, F. Chaudhry, P. Cheung, and L. Reis. Optimal Design of Water Distribution System by Multiobjective Evolutionary Methods. In C. Fonseca et al., editor, *Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003, LNCS 2632*, pages 677–691. Springer, 2003.

[44] O. Fujiwara and D. Khang. A Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks. *Water Resources Research*, 26(4):539–549, 1990.

[45] Z. Geem. Optimal cost design of water distribution networks using harmony search. *Engineering Optimization*, 38(3):259–277, 2006.

[46] A. George and J. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1981.

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

[48] H. Gomes, S. de Tarso Marques Bezerra, P. de Carvalho, and M. Salvino. Optimal dimensioning model of water distribution systems. *Water SA*, 35(4):421–431, 2009.

[49] I. Goulter. Systems Analysis in Water-Distribution Network Design: From Theory to Practice. *Journal of Water Resources Planning and Management*, 118(3):238–248, 1992.

[50] C. Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962.

[51] L. Jourdan, D. Corne, D. Savic, and G. Walters. Hybridising Rule Induction and Multi-Objective Evolutionary Search for Optimising Water Distribution Systems. In *Proceedings of the 4th International Conference on Hybrid Intelligent Systems, HIS'04*, pages 434–439. IEEE Computer Society, 2004.

[52] L. Jourdan, D. Corne, D. Savic, and G. Walters. Preliminary Investigation of the 'Learnable Evolution Model' for Faster/Better Multiobjective Water Systems Design. In C. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Proceedings of the 3rd International Conference Evolutionary Multi-Criterion Optimization, EMO 2005, LNCS 3410*, pages 841–855. Springer, 2005.

[53] Z. Kapelan. *Calibration of Water Distribution System Hydraulic Models.* PhD Thesis, School of Engineering and Computer Science, University of Exeter, UK, 2002.

[54] Z. Kapelan, D. Savic, and G. Walters. Multiobjective design of water distribution systems under uncertainty. *Water Resources Research*, 41(11), 2005.

[55] E. Keedwell and S.-T. Khu. A novel evolutionary meta-heuristic for the multi-objective optimization of real-world water distribution networks. *Engineering Optimization*, 38(3):319–333, 2005.

[56] J.-W. Klinkenberg, M. Emmerich, A. Deutz, O. Shir, and T. Bäck. A Reduced-Cost SMS-EMOA using Kriging, Self-Adaptation, and Parallelization. In M. Ehrgott, B. Naujoks, T. Stewart, and J. Wallenius, editors, *Proceedings of the 19th International Conference on Multiple Criteria Decision Making, MCDM 2008, LNEMS 634*. Springer, 2008.

[57] J. Knowles. A Summary-Attainment-Surface Plotting Method for Visualizing the Performance of Stochastic Multiobjective Optimizers. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, ISDA 2005*, pages 552–557. IEEE Computer Society, 2005.

[58] J. Knowles and D. Corne. On Metrics for Comparing Non-Dominated Sets. In *Proceedings of the 2002 World Congress on Computational Intelligence, WCCI 2002*, pages 711–716, 2002.

[59] J. Knowles and D. Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, 2003.

[60] J. Kruisselbrink. Robust optimization. Optimization under uncertainty and noise. Technical report, Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands, to appear.

[61] J. Kruisselbrink, M. Emmerich, and T. Bäck. An Archive Maintenance Scheme for Finding Robust Solutions. In *PPSN XI: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature, LNCS*. Springer, to appear.

[62] K. Lansey. Uncertainty in Water Distribution Network Modeling. *Water Resources Update, Universities Council on Water Resources*, (103):22–26, 1996.

[63] R. Li. *Mixed-Integer Evolution Strategies for Parameter Optimization and Their Applications to Medical Image Analysis.* PhD Thesis, Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands, 2009.

[64] S.-Y. Liong and M. Atiquzzaman. Optimal Design of Water Distribution Network Using Shuffled Complex Evolution. *Journal of the Institution of Engineers, Singapore*, 44(1):93–107, 2004.

[65] I. Lippai, J. Heaney, and M. Laguna. Robust Water System Design With Commercial Intelligent Search Optimizers. *Journal of Computing in Civil Engineering*, 13(3):135–143, 1999.

[66] H. Maier, A. Simpson, A. Zecchin, W. Foong, K. Phang, H. Seah, and C. Tan. Ant Colony Optimization for Design of Water Distribution Systems. *Journal of Water Resources Planning and Management*, 129(3):200–209, 2003.

[67] L. Mays. Review of reliability analysis of water distribution systems. In K. Tickle et al., editor, *Stochatic Hydraulics '96: Proceedings of the Seventh IAHR International Symposium*, pages 53–62. Balkema, 1996.

[68] I. Montalvo, J. Izquierdo, R. Pérez, and M. Tung. Particle Swarm Optimization applied to the design of water supply systems. *Computers and Mathematics with Applications*, 56(3):769–776, 2008.

[69] L. Murphy, A. Simpson, and G. Dandy. Pipe network optimization using an improved genetic algorithm. Technical Report R109, Department of Civil and Environment Engineering, University of Adelaide, Australia, 1993.

[70] H. Nielsen. Methods for Analyzing Pipe Networks. *Journal of Hydraulic Engineering*, 115(2):139–157, 1989.

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

[72] E. Reehuis and T. Bäck. Mixed-Integer Evolution Strategy Using Multiobjective Selection Applied to Warehouse Design Optimization. In J. Branke et al., editor, *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO'10*. ACM Press, to appear.

[73] L. Rossman. EPANET 2 Users Manual. Technical Report EPA/600/R-00/057, Water Supply and Water Resources Division, National Risk Management Research Laboratory, U.S. Environmental Protection Agency, Cincinnati, OH, USA, 2000.

[74] D. Savic. Single-objective vs. Multiobjective Optimisation for Integrated Decision Support. In A. Rizzoli and A. Jakeman, editors, *Proceedings of the 1st Biennial Meeting of the International Environmental Modelling and Software Society, Integrated Assessment and Decision Support, iEMSs 2002*, pages 7–12, 2002.

[75] D. Savic and G. Walters. Genetic Operators and Constraint Handling for Pipe Network Optimization. In T. Fogarty, editor, *AISB Workshop on Evolutionary Computing, LNCS 993*, pages 154–165. Springer, 1995.

[76] D. Savic and G. Walters. Genetic Algorithms for Least-Cost Design of Water Distribution Networks. *Journal of Water Resources Planning and Management*, 123(2):67–77, 1997.

[77] J. Schaake and D. Lai. Linear Programming and Dynamic Programming Application to Water Distribution Network Design. Technical Report 116, M.I.T. Hydrodynamics Laboratory, Department of Civil Engineering, Cambridge, MA, USA, 1969.

[78] M. Schütz. Eine Evolutionsstrategie für gemischt-ganzzahlige Optimierungsprobleme mit variabler Dimension. Technical Report SYS-1/96, Department of Computer Science, TU Dortmund University, Germany, 1996.

[79] H.-P. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation.* John Wiley & Sons, Inc., New York, NY, USA, 1993.

[80] F. Spellman. *Handbook of Water and Wastewater Treatment Plant Operations, Second Edition.* CRC Press, Boca Raton, FL, USA, 2009.

[81] T. Tanyimboh, M. Tabesh, and R. Burrows. Appraisal of Source Head Methods for Calculating Reliability of Water Distribution Networks. *Journal of Water Resources Planning and Management*, 127(4):206–213, 2001.

[82] E. Todini. Looped water distribution networks design using a resilience index based heuristic approach. *Urban Water*, 2(2):115–122, 2000.

[83] E. Todini and S. Pilati. A Gradient Algorithm for the Analysis of Pipe Networks. In B. Coulbeck and C. Orr, editors, *Computer Applications in Water Supply, Volume I: Systems Analysis and Simulation*, pages 1–20. John Wiley & Sons, Inc., 1988.

[84] B. Tolson, M. Asadzadeh, H. Maier, and A. Zecchin. Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization. *Water Resources Research*, 45(12), 2009.

[85] B. Tolson, H. Maier, A. Simpson, and B. Lence. Genetic Algorithms for Reliability-Based Optimization of Water Distribution Systems. *Journal of Water Resources Planning and Management*, 130(1):63–72, 2004.

[86] J. Tospornsampan, I. Kita, M. Ishii, and Y. Kitamura. Split-Pipe Design of Water Distribution Network Using Simulated Annealing. *International Journal of Computer, Information, and Systems Science, and Engineering*, 1(3):153–163, 2007.

[87] J. Tospornsampan, I. Kita, M. Ishii, and Y. Kitamura. Split-Pipe Design of Water Distribution Networks Using a Combination of Tabu Search and Genetic Algorithm. *International Journal of Computer, Information, and Systems Science, and Engineering*, 1(3):164–174, 2007.

[88] K. Vairavamoorthy and M. Ali. Optimal Design of Water Distribution Systems Using Genetic Algorithms. *Computer-Aided Civil and Infrastructure Engineering*, 15(5):374–382, 2000.

[89] K. Vairavamoorthy and J. Lumbers. Leakage Reduction in Water Distribution Systems: Optimal Valve Control. *Journal of Hydraulic Engineering*, 124(11):1146–1154, 1998.

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

[91] M. van Dijk, S. van Vuuren, and J. van Zyl. Optimising water distribution systems using a weighted penalty in a genetic algorithm. *Water SA*, 34(5):537–548, 2008.

[92] J. van Zyl and C. Clayton. The effect of pressure on leakage in water distribution systems. *Proceedings of the ICE – Water Management*, 160(2):109–114, 2007.

[93] J. van Zyl, D. Savic, and G. Walters. Operational Optimization of Water Distribution Systems Using a Hybrid Genetic Algorithm. *Journal of Water Resources Planning and Management*, 130(2):160–170, 2004.

[94] A. Vasan and S. Simonovic. Optimization of Water Distribution Network Design Using Differential Evolution. *Journal of Water Resources Planning and Management*, 136(2):279–287, 2010.

[95] T. Walski. The Wrong Paradigm—Why Water Distribution Optimization Doesn't Work. *Journal of Water Resources Planning and Management*, 127(4):203–205, 2001.

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

[97] D. Yates, A. Templeman, and T. Boffey. The Computational Complexity of the Problem of Determining Least Capital Cost Designs for Water Supply Networks. *Engineering Optimization*, 7(2):143–155, 1984.

[98] E. Zitzler, D. Brockhoff, and L. Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In S. Obayashi et al., editor, *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007, LNCS 4403*, pages 862–876. Springer, 2007.

[99] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

[100] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK) – Swiss Federal Institute of Technology (ETH) Zürich, Switzerland, 2001.

[101] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, LNCS 1498*, pages 292–301. Springer, 1998.

[102] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.