TOROS UNIVERSITY
GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OPTIMIZATION OF THE SUSTAINABILITY OF CONTINGENCY LOGISTICS NETWORKS: APPLICATION OF A HYBRID HEURISTIC & A MULTI-OBJECTIVE OPTIMIZATION APPROACHES

M.Sc. THESIS

Havva Esra Dağ

Department of Industrial Engineering

August 2015

Mersin

TOROS UNIVERSITY
GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OPTIMIZATION OF THE SUSTAINABILITY OF CONTINGENCY LOGISTICS NETWORKS: APPLICATION OF A HYBRID HEURISTIC & A MULTI-OBJECTIVE OPTIMIZATION APPROACHES

M.Sc. THESIS

Havva Esra DAĞ
128010001

Department of Industrial Engineering

August 2015

Mersin

Toros Üniversitesi Fen Bilimleri Enstitüsü'nün 128030001 numaralı Yüksek Lisans Öğrencisi "Havva Esra DAĞ'', ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "Optimization of the Sustainability of Contingency Logistics Networks: Application of a Hybrid Heuristic & a Multi-objective Optimization Approaches'' başlıklı tezini aşağıda imzaları olan jüri önünde 31.08.2015 tarihinde sunmuş ve başarılı olduğu oybirliği/~~oyçokluğu~~ ile kabul edilmiştir.

~~Yrd.~~ Doç. Dr. Mehmet MİMAN(Danışman)
Toros Üniversitesi

Prof. Dr. Ali Kokangül (Üye)
Çukurova Üniversitesi

Prof. Dr. D. Tayyar Şen (Üye)
Toros Üniversitesi

Bu Tez Enstitümüz Endüstri Mühendisliği Anabilim Dalında Hazırlanmıştır.

Yrd. Doç. Dr. Onur Boyacıgil GÜNGÖR
Enstitü Müdür V.

NOT: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin , çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ACKNOWLEDGEMENT

Completing this thesis successfully, I want to convey my appreciations to those who contribute to this success. First of all, special thanks go to my committee members: Assist. Prof. Dr. Mehmet Miman, who is one of the best advisors one can ever have. He is also very friendly and helpful to me such that he steps through each stage of my thesis, motivates me all the time during my graduate study and I always felt his invaluable assistance in completion of this thesis.

Prof. Dr. D. Tayyar Şen, whose wisdom and experience have been guiding and encouraging light in my career life, deserves a special place here to mention. I believe the students are very lucky to have such a role model. During my graduate study as a master student, he has given me the opportunity to gain knowledge and share experiences with many people that have contributed to shape my thoughts as a researcher and as a person. I thank him a lot and my sincere gratitude also goes to him. I will always proud of becoming a student of him. It is also worthwhile to note that he has contributed a lot to this thesis, especially in editing.

Prof. Dr. Ali Kokangül provided technical supports related to this study and opened a wide range of future research directions through his expertise in optimization. I appreciate all of his supports and regard myself likely to have him in my committee.

Assist. Prof. Dr. Fikri Ege, who is the head of Department of Industrial Engineering, provides guidelines and recommendation on selecting courses, answers my questions promptly.

Assist. Prof. Dr. Mustafa Deste is one of the special person whose assistance I feel a lot although he is working in another university. His undeniable help motivated me a lot and enabled this thesis to be complete.

Assist. Prof. Dr. Ziya Gökalp Altun who has reviewed my thesis carefully from the beginning to end, and provided necessary corrections to improve the quality of written document is an invaluable place in completion of this study.

Assist. Prof. Dr. Ali Kemal Havare, who contributes a lot to the completion of this thesis with his enthusiastic, student friendly way of conducting the duty of being director of Graduate Schools of Science and Engineering.

I appreciate that Prof. Dr. Pohl have enabled my advisor to develop the model which is a focus of this study. Although I do not meet him face to face, as much as I know him from my advisor, he has such an admirable personality as a scholar that can facilitate the progress of others on the way of their academic career.

Av. Emrah Aslan who has always supported me during my graduate study possess a special place in my heart and I thank to him a lot.

Even though, I could not mention each of their names here, all of the faculty members who provide me a high level of education, hence enable me to achieve having a GPA of 4.00 at the end of my graduate study deserves special thanks.

Finally, with all my heart, I cannot deny the support in terms of all means, encouragement on my graduate study I have been receiving from my all of the family members, especially my dad Fasih Dağ, my mom Şükriye Dağ, my sisters Kübra Dağ and İnaz Dağ. Their presence makes me confident to be successful in my life as well as in my career, where the idea of pursuing a graduate degree was first initiated by Şükriye Dağ and Kübra Dağ, for whom, therefore, I will always become grateful.

I dedicate this thesis to all of members of my family that will always be proud of.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENT iv
TABLE OF CONTENT vi
ABBREVIATIONS ix
LIST OF TABLES x
LIST OF FIGURES xi
LIST OF EXHIBITS xii
ABSTRACT xiii
ÖZET xiv
1. INTRODUCTION AND OVERVIEW 1
1.1. Problem Statement and Research Objectives 2
1.2. Research Motivation 3
1.3. Coverage, Limitations and Assumptions 3
1.4. Thesis Overview 4
2. LITERATURE REVIEW 5
2.1. Contingency Logistics Systems 5
2.1.1. Contingency Construction 7
2.1.2. Formulation and Analysis of Contingency Logistics Network Models 7
2.1.3. Contingency Logistics Networks 12
2.1.3.1. Importance Measures for CLN 12
2.1.3.2. Distorted Risks in a CLN 14
2.1.3.2.1. Distorted Risk Measures (Series and Parallel) 17
2.1.3.2.2. Risk Displacement Factor 18
2.2. Supply Chain and Risk Analysis for Contingencies 20
2.3. Relevant Techniques for Analysis of a CLN 22
2.3.1. Analytical Approaches to CLN Design 22
2.3.2. Simulation Approach to CLN Design 22
2.3.3. Importance Measures and Sensitivity Analysis 23
2.3.4. Optimization-through Knapsack Approach and Multi-Objective Analysis Techniques 24
2.3.5. Metaheuristics Optimization Techniques 28
2.3.5.1. General Optimization Lexicon 29
2.3.5.2. Characteristics of Metaheuristics 31
2.3.5.3. A Brief History of Metaheuristics 31

2.3.5.4. Simulated Annealing 32
2.3.5.5. Genetic Algorithms 34
2.3.5.6. Differential Evolution 36
2.3.5.7. Ant Colony Optimization 37
2.3.5.8. Bee Algorithms 38
2.3.5.9. Particle Swarm Optimization 39
2.3.5.10. Tabu Search 40
2.3.5.11. Harmony Search 40
2.3.5.12. Firefly Algorithm 41
2.3.5.13. Cuckoo Search 43
2.3.5.14. Water Cycle Algorithm 44
2.3.5.15. Other Metaheuristics Algorithms 44
2.3.5.16. A General Design Example for Metaheuristics 45
2.3.6. Multi-Objective Analysis Techniques 46
2.3.6.1. Goal Programming 49
2.3.6.2. Weighted Objective Function 50
2.3.6.2.1. Shannon Entropy Method 51
2.3.6.2.2. MultiMoora Method 53
2.3.6.3. Multi-Attribute Utility Function 54
2.3.6.4. Pareto Optimality 55
2.3.6.5. Analytical Hierarchy Process (AHP) 56
2.3.6.6. Physical Programming 60
2.3.6.7. Utopia Distance 61
2.3.6.7.1. Notation List for Utopia Distance 64
2.3.6.8.2. Procedures of Utopia Distance 64
3. Sustainability Optimization of a Contingency Logistics Network: A multi-dimensional Knapsack Approach 72
3.1. Sustainability Model for a CLN 73
3.1.1. Notation List 73
3.1.2. Modeling Node Reliability 76
3.1.3. Modeling Link Reliability 76
3.1.4. Modeling the Sustainability of a CLN 79
3.2. Heuristic Approaches Investigated 84
3.2.1. Traditional Heuristic Approaches Investigated 85
3.2.1.1. Genetic Algorithm 85
3.2.1.2. Simulated Annealing 88
3.2.2. A Hybrid Heuristic: EDGASA (Genetic Algorithm with Simulating Annealing by Esra Dağ) 91

3.3. Comparisons of Algorithms Investigated 95
3.3.1. Experimental Design.. 96
3.3.2 Analysis of Results...101
3.4. Multi-objective Optimization Modeling for
The Sustainability of a CLN based on Utopia Distance 110
3.4.1. Additional Notation List 110
3.4.2. Multi-Objective Model Developed 111
4. CONCLUSION AND DISCUSSION 114
REFERENCES...117
APPENDICES..125
APPENDIX 1. Packages Used for Coding Algorithms 125
APPENDIX 2. Initial Variables and Parameter Used in Codes..126
APPENDIX 3. makeExperimentalSetup Function 129
APPENDIX 4. Common Functions used in all of the Algorithms 130
APPENDIX 5. Codes of Functions for GA Algorithm...132
APPENDIX 6. Codes of Functions for SA Algorithm 137
APPENDIX 7. Codes of Functions for EDGASA Hybrid Algorithm 139
APPENDIX 8. Results Obtained from GA..143
APPENDIX 9. Results Obtained from SA...145
APPENDIX 10. Results Obtained from EDGASA 147
CURRICULUM VITAE..149

ABBREIVATIONS

ABC	**:**Artificial Bee Colony
ACO	**:**Ant Colony Optimization
AHP	**:**Analytical Hierarchy Process
C.I.	**:**Consistency Index
CI	**:**Confidence Interval
CLN	**:**Contingency Logistics Network
CLS	**:**Contingency Logistics System
CM	**:**Corrective Maintenance
CS	**:**Cuckoo Search
DC	**:**Distribution Center
DP	**:**Dual Power
DE	**:**Different Evolution
EDGASA	**:**Genetic Algorithm with Simulating Annealing by Esra Dağ
FA	**:**Firefly Algorithm
FGP	**:**Fuzzy Goal Programming
GA	**:**Genetic Algorithms
GB	**:**Gamma-Beta
GP	**:**Goal Programming
HBA	**:**Honey Bee Algorithm
HBMO	**:**Honey-Bee Mating Optimization
HS	**:**Harmony Search
IDE	**:**Integrated Development Environment
IPA	**:**Infinitesimal Perturbation Analysis
LP	**:**Linear Programming
MCDM	**:**Multi-Criteria Decision Making
MOORA	**:**Multi-Objective Optimization Technique Using Ratio Analysis
MOP	**:**Multiple Objective Optimization Problem
PH	**:** Proportional Hazard
PM	**:**Preventive Maintenance
PP	**:**Physical Programming
PSO	**:**Particle Swarm Optimization
R.I.	**:**Random Index
SA	**:**Simulated Annealing
TS	**:**Tabu Search
VBA	**:**Virtual Bee Algorithm
WCA	**:**Water Cycle Algorithm
MULRR	**:**Multi Level Ruin and Rebuild Algorithm
GAFTS	**:**Genetic Algorithm fed by Tabu Search

LIST OF TABLES

Page

Table 2.1. Displacement Factors through Distortion..18
Table 2.2. Displacement Factors for Natural Vulnerability 19
Table 2.3. Multi-Objective Optimization Technics...48
Table 2.4. The Fundamental Scale for Pairwise Comparisons
used in the AHP Approach..57
Table 2.5. Random Consistency Index..60
Table 3.1. Two-bit Binary Representation for
Maintenance Alternatives of link *j* 85
Table 3.2. Comparison of Algorithms based on their Mechanisms 95
Table 3.3. Expected Values for Decision Alternatives...97
Table 3.4. Experimental Design Parameters 99
Table 3.5. Descriptive Statistics for Heuristic Solutions
in terms of Average Reliability... 101
Table 3.6. Hypothesis Testing for the Average Reliability of a
CLN by Heuristics 102
Table 3.7. Multiple Comparisons with EDGASA.. 103
Table 3.8. Results for Simultaneous Comparisons of Mean Performances of
Heuristics on Average-R in Minitab 104
Table 3.9. Comparison of Algorithms in ability to find a Better Solution
in terms of Average Reliability... 107
Table 3.10. Comparison of Algorithms in ability to find a Better Solution
in terms of Minimum Reliability... 107
Table 3.11. Comparison of Algorithms in ability to find a Better Solution
in terms of Maximum Reliability.. 108
Table 3.12. Descriptive Statistics for Heuristics in terms of
Average Solution Time 109

LIST OF FIGURES

Page

Figure 2.1. Resource Loading Profile for a CLS 9
Figure 2.2. Basic CLN with Stocks 13
Figure 2.3. Distorted Link's Effect on the System 16
Figure 2.4. Risk Surface under Distortion 19
Figure 2.5. Illustration of Replacement through Distortion 20
Figure 2.6. Genetic Algorithm (Initial Population and Locations) 35
Figure 2.7. Genetic Algorithm (Final Locations) 36
Figure 2.8. Display How All Particles Move Towards the Global Optimum 40
Figure 2.9. Firefly Algorithm and Fireflies Move Towards 4 Global Optima 42
Figure 2.10. Illustration of Domination 50
Figure 2.11. Pareto-Optimal Solutions 56
Figure 2.12. A Hierarchy Structure in AHP 58
Figure 2.13. The Concept and Procedure of the Adaptive Weighted Sum Method 62
Figure 2.14. Adaptive Weighted Sum Method (Utopia Distance) for Multi-Dimensional Problems (2-D Representation) 63
Figure 2.15. Utopia Distance for 3-D Problems 63
Figure 2.16. Adaptive Refinement Procedure 67
Figure 2.17. Configuration of an Additional Equality Constraint for Refinement (3-D Representation) 69
Figure 2.18. Pareto front Obtained by the Usual Weighted Sum Method 70
Figure 3.1. Network Representation of the CLN in Consideration 82
Figure 3.2. CLN Sustainability: Mission Perspective 83
Figure 3.3. Encoding of a Solution for the CLN with *n*-bases 85
Figure 3.4. Next Population Generation in GA 87
Figure 3.5. Steps of Simulated Annealing Algorithm Applied 90
Figure 3.6. Initial Population in EDGASA 92
Figure 3.7. Generation Formation in EDGASA 94
Figure 3.8. The Structure of the CLN Considered for the Experimentation 100
Figure 3.9. Tukey's Simultaneous 95% CIs Plot for Pairwise Differences 105
Figure 3.10. Performance Space for Utopia Distance Model 111
Figure 3.11. Solution Steps for Multi Objective Optimization Model 112

LIST OF EXHIBITS

Page

Exhibit 2.1. General Lexicon for Optimization Problems .... .29
Exhibit 2.2. General Design Optimization Model .... .45
Exhibit 2.3. Sub-Optimization Problem for the *j*th
Normalized Expected Solution .... .68
Exhibit 3.1. Link *j* Reliability Definition in a CLN with Maintenance .... .77
Exhibit 3.2. The Sustainability Optimization Model for a CLN............................ .80
Exhibit 3.3. Sustainability Model as an IP for a CLN in compact form .... .84
Exhibit 3.4. Multi-Objective Optimization Model based on Utopia Distance....... 111

OPTIMIZATION OF THE SUSTAINABILITY OF CONTINGENCY LOGISTICS NETWORKS: APPLICATION OF A HYBRID HEURISTIC & A MULTI-OBJECTIVE OPTIMIZATION APPROACHES

ABSTRACT

Contingencies are unexpected crises or events that cause a major threat to the safety, security and well-being of a certain population. This research effort builds upon the work on contingency logistics reliability models by Miman (2008) who extended the preliminary work conducted by Thomas (2004) that provides the modeling approach which takes a mission success orientation and focuses on the ability to recover from or prevent a contingency logistics failure. Miman (2008) proposes the sustainability model of a contingency logistics network using the concept of selective maintenance. This problem, once formulated, is a non-convex, non-linear, non-separable, multi-dimensional, discrete knapsack problem. These problems are known to be NP hard. Therefore, one needs to explore heuristic solutions in search of robust and effective solution approaches. He developed a memetic algorithm, GAFTS, and proposed this for identifying the best set of maintenance actions to sustain the contingency logistics network. Besides, he used Physical Programming, a multi criteria optimization procedure, to exploit a network manager's preference toward the numerous criteria (reliability, cost, time, resource utilization etc...) judiciously.

This research effort continues the exploration of heuristic techniques for the sustainability model developed by Miman (2008) and develops a hybrid heuristics technique, EDGASA, incooperating simulating annealing (SA) procedure with genetic algorithm (GA). Comparisons of EDGASA with GA and SA reveal that it outperforms in terms of average reliability, best reliability and worst reliability found at an expense of increased solution time.

One of the contributions of this study is a multi-objective modeling approach developed based on utopia distance that aims at minimizing the weighted distance between a solution to the ideal point that could be achieved. The study fills some of the voids in the contingency logistics networks' solution and modeling and highlights potential studies by applying the hybrid heuristic developed as well as multi-objective modeling approach proposed to other problems.

BEKLENMEDİK DURUMLAR LOJİSTİK AĞLARININ SÜRDÜRÜLEBİLİRLİĞİNİN ENİYİLENMESİ: BİR HİBRİT SEZGİSEL YÖNTEMİN VE ÇOK AMAÇLI ENİYİLEME YAKLAŞIMININ UYGULANMASI

ÖZET

Beklenmedik durumlar, belirli bir halkın güvenlik ve zenginliğine büyük bir tehdit oluşturan beklenmedik kriz veya olaylardır. Bu çalışma, Thomas (2004)'ın gerçekleştirdiği ve beklenmedik durumlar lojistiğinin başarısızlığından kurtulma veya istenmeyen durumları önleme için görev başarısı mantığına dayalı öncü çalışmayı geliştiren Miman (2008)'in sağladığı beklenmedik durumlar lojistiği sürdürülebilirliği modelleri üzerine kurulmuş çalışmayı geliştiren bir araştırma çabasının ürünüdür. Miman (2008) seçici bakım kavramını kullanarak bir beklenmedik durumlar lojistik ağı için sürdürülebilirlik modeli önermektedir. Bu problem, formüle edildikten sonra, konveks olmayan, doğrusal olmayan, ayrılamayan, çok boyutlu, ayrık, bir knapsack problemidir. Bu tür problemler çözüm süresi açısından polinomsal olmayan (NP-zor) bilinmektedir. Bundan dolayı esnek ve verimli çözüm yaklaşımlarında sezgisel çözümlerin araştırılmasına ihtiyaç vardır. Miman (2008) bir memetik algoritma (GAFTS) geliştirmiş ve beklenmedik durumlar lojistik ağını sürdürmede en iyi bakım faaliyetlerin kümesini belirlemek için bu algoritmayı önermiştir. Bundan başka, ağın yöneticisinin çok çeşitli kriterler için tercihlerini (güvenilirlik, maliyet, zaman, kaynak kullanım verimliliği… vb.) açıkça değerlendirmek üzere, birçokamaçlı eniyileme tekniği olan fiziksel programlamayı kullanmıştır.

Bu araştırma çabası Miman (2008) tarafından geliştirilen sürdürülebilirlik modeli için sezgisel yöntemlerin araştırılması ve keşfedilmesini devam ettirmekte olup benzetilmiş tavlama (SA)'yı genetik algoritma (GA) içinde kullanarak bir hibrit sezgisel yöntem (EDGASA) geliştirmektedir. EDGASA'nın GA ve SA ile karşılaştırmaları; onun artan çözüm süresi karşılığıyla; bulunan ortalama, en iyi ve en kötü güvenirliliklerde diğerlerinden daha iyi performansa sahip olduğunu göstermektedir.

Bu çalışmanın başka bir katkısı, bir çözümün başarılabilecek ideal noktaya uzaklığını en aza indirmeyi amaçlayan ütopya uzaklık mantığına dayalı geliştirilen birçok amaçlı modelleme yaklaşımıdır. Bu çalışma beklenmedik durumlar lojistik ağının çözümünde ve modellenmesinde bazı boşlukları doldurmakta ve önerilen çok amaçlı modelleme yaklaşımının yanında geliştirilen hibrit sezgisel algoritmanın diğer problemlere uygulanabileceğini vurgulayarak potansiyel araştırmalara ışık tutmaktadır.

1

Introduction and Overview

Contingencies are unexpected events or crisis that cause a major threat to the safety, security and well-being of specific populations. There are variety of situations such as terrorist attacks, military conflicts or natural disasters that can be regarded as a contingency. In the literature there are considerable number of studies that considers the typical logistics systems and supply chain networks, yet, there are limited ones that deals with the contingency logistics networks, especially the reliability of the network. Thomas (2004) explicitly considers the reliability of contingency logistics systems from the perspective of mission success where the missions assigned to the network in the contingency to recover can be achieved in terms of demand and supply interference. He proposes formulas for the base reliability based on the interference between demand and supply such that the reliability of the base is computed by the probability that the demand is less than the supply. He also proposes maximum entropy distributions to derive the base's reliability in case of limited data on the demand and supply distributions. A considerable jump on the literature in reliability of contingency logistics networks (CLNs) was taken by Miman (2008). He considers stocks to be held by base due to uncertainties on the demand and supply for contingency operations. Further, he proposes a sustainability model for the reliability of contingency logistics networks through multi-action selective maintenance on the links between the central depot and operational bases. The resulting sustainability model is a non-linear, non-separable, non-convex model, thus he proposes a set of metaheuristics, MULRR (Multi-level ruin and rebuild algorithm) and GAFTS (Genetic Algorithm fed by Tabu

Search) in addition to traditional methods of TS (Tabu Search) and GA (Genetic Algorithm) as a potential solution approaches to the model to identify the best set of maintenance actions to maximize the network's reliability under the available budget and time limits for performing those actions. He concludes that the memetic algorithm he developed, GAFTS, outperforms the others in consideration along with an increased solution time. One of the contributions in contingency logistics network modeling provided by Miman (2008) is the application of PP as a multi-objective application technique to the sustainability model to exploit the decision makers preference on each of the objectives (reliability, cost and time) through the desirability levels set by the decision maker.

1.1. Problem Statement and Research Objectives

As the main milestones for the literature on the contingency logistics networks and their reliability was mentioned above, this study on focus the investigation of robust and effective solution approaches for the model provided by Miman (2008) and multi-objective modeling approaches for the sustainability of the CLNs.

The research statements for this study can be expressed as:

- How well other traditional metaheuristics, namely SA (simulated annealing) and GA performs on the solution to the model?
- Can a hybrid heuristic based on SA and GA improve the solution quality?
- Is it possible to model the sustainability of CLN through a multi-objective optimization techniques?

According to above research statements, this study aims at investigating the SA and GA, developing a hybrid heuristic, EDGASA based on GA and SA in search of robust and effective solution approaches to the sustainability model provided by Miman (2008). Another purpose of this study is to propose a multi-objective optimization model for the sustainability of the CLNs based on utopia distance. All of the above efforts eventually provided greater insights for the CLN planners in terms of modeling and solution approaches.

1.2. Research Motivation

Today's world, the probability of occurrence of a contingency event and/or the consequences of possible contingencies is not negligible, therefore it has a vital responsibility to get prepared for contingencies. One of the steps of it is to have a CLN design and plan to increase the probability of the network to recover from the contingency, i.e. reliability through the ability to perform the assigned missions. The sustainability model proposed by Miman (2008) identifies the best set of maintenance actions under limited budget and time to perform the actions to maximize the reliability of the CLN. In doing this, solution's approach performances in quality and time of the solution has a significant role as in real life the problem size is likely to be large in addition to the model's inherent non-linear- non-separable and non-convex structure. Therefore, the investigation of solution approaches in search of robustness and effectiveness is very important. Along with the maximization of reliability of the network through selective maintenance, the minimization of cost of maintenance and time to perform maintenance activities can also play a significant role in planning a CLN. Hence, the investigation of multi-objective optimization techniques in the context of contingency logistics networks has also considerable level of importance as well.

1.3. Coverage, Limitations and Assumptions

This study focuses on the sustainability model proposed by Miman (2008). Although this model is a two-dimensional knapsack model, without loss of generality, the solution approaches investigated by this study is applicable to knapsack models with any dimensions. The model assumes that each link in the CLN has a Weibull life distribution. There is a single link between a central depot and each of the operational bases. The lateral shipments among bases are not allowed, hence, no link exists among bases. Each of operational bases supports unique single operation of the assigned mission. There are two main constraints to perform maintenance activities; available time and budget.

1.4. Thesis Overview

To achieve the research goals highlighted above the thesis is arranged as follows: The second chapter of the thesis provides a detailed literature on the contingency logistics networks, solution approaches and multi-objective optimization techniques. In detail, the literature on the contingency logistics systems, supply chain and risk analysis for contingencies, relevant techniques for analysis of a CLN is exhaustively explored and provided. Chapter 3 constitutes the integral part of this study. It first gives the sustainability model in consideration and proposed by Miman (2008) in order to make the study self-standing. Next, it provides the details about the heuristic approaches namely GA, SA and a hybrid heuristic developed, EDGASA, in consideration as a solution approach to the sustainability model. Later, the performances of these algorithms are compared to each other based on an experimental design. Finally, a multi-objective optimization model of the sustainability of a CLN is proposed based on utopia distance approach. The study ends with conclusion and discussion part which highlights the results obtained through this research effort, contribution of this study to the literature and potential research directions the study can provide.

2

Literature Review

This chapter provides the results of exhaustive search on the literature related to the contingency logistics systems and optimization models that are used for them. In particular, it gives an investigation of the instruments and procedures that are appropriate for the analysis, optimization and evaluation of design plans for contingency systems. The first two parts (Section 2.1-2.2) give a general audit of contingency logistics systems and particularly inspect their flexibility in modeling interruptions and more pertinent literature as far as risks, especially those for failure to perform assigned missions to recover from the contingencies. Finally, relevant techniques including metaheuristics algorithms and multi-objective optimization approaches are provided as the optimization models in contingency logistics networks are more likely to be NP hard problems and there is a need for considering competing objectives at the same time.

2.1. Contingency Logistics Systems

A contingency is a randomly happening crisis, for example, a national catastrophe, civil disorder, or military invasion that causes a noteworthy threat to the safety and security of a population that requires a quick response where the development and maintenance of supply chains to give the logistics support capacities should be covered in the design and planning stages of the contingency logistics network (CLN). In the literature, one of the methods available for assessing the reliability of an operational base in a contingency logistics network bases on the interference theory between the demand and supply for that base. From the perspective of mission success, the logistics network reliability is conceptualized as the likelihood of the network to perform the necessary operations required by the mission. This enforces the network to have mission necessities providing enough supplies to the bases that perform the operations (Thomas, 2004) or to perform the best set of selective maintenance actions on the links in the network (Miman, 2008). Both Thomas (2004) and Miman (2008) consider that the set of items required to perform operations in a given mission are supplied from a central distribution center (warehouse) to operational bases (sites) in the network to enable bases to be functioning in terms of equipment. Similarly, according to Miman (2008) a set of selective maintenance actions can be performed on the links in the network so that the network survives the mission. The functioning links transport equipment and materials required by bases, eventually the bases who have enough supplies (through interference theory) transported to them (through the links) can be regarded as functioning (reliable) for the contingency mission.

Basically all means of ways in the CLN should be considered to provide the construction support for the mission as it is generally cumbersome and naturally difficult to satisfy the resources demanded and requests emerged by the tasks due to potential vulnerabilities with the accessibility of these resources and the requests for the tasks.

Commonplace contingency missions are emerged to counter, hinder, or diffuse restricting strengths; save and aid victims; and restore open works foundation. There are set of basic requirements for sending work force, equipment, and supplies for conducting contingency operations to complete the mission in order to recover from the contingency.

Logistics networks must be established for supply chains, replenishment, and distribution; medical treatment facilities; and shelter. In general, all contingency operations include considerable construction requirements for the construction, repair, and maintenance of facilities, roads, airfields, and utilities. Even in case of normal and ideal conditions managing these functions can be challenging. Uncertainties and urgency inherent in contingency operations make it more difficult to the coordination of scheduling, tracking, and allocation of resources for accomplishing the mission.

Thomas (2008) presents a time dependent model for planning activities during sustainment operations. He assesses the mission reliability of a project using the probability of interference between a load measured in the number of days required to completion, and the capacity which is taken as the number of days available for accomplishing the construction mission based on the allotted resources. Although his statement was inspired by military cases, he claims that without loss of generality the models are valid for non-military cases as well.

2.1.1. Contingency Construction

All contingency operations incorporate some level of construction that is necessary for the accomplishment of the general mission. The amount and kind of support for this depends on the specific mission, such as military engagement with a hostile force, rescue and recovery operation or humanitarian relief operations (Thomas *et al.* (2002). They also describe the contingency operations in three district phases as:

1. Mobilization and Deployment,
2. Sustained Operations and
3. Reconfiguration.

According to Kiwus (1990) contingency construction can be categorized into three broad classes: temporary construction (utilization up to 3-6 months), expeditionary construction (utilization for a period of 6 months to 1 year) and permanent construction (not common in contingency operations).

According to Kiwus (1990), the biggest challenge in managing contingency is tracking project performance and the effectiveness in supporting the mission requirements.

2.1.2. Formulation and Analysis of Contingency Logistics Network Models

A contingency logistics system (CLS) is defined by Thomas (2004) as "*a set of processes and methods for providing the procurement, distribution, storage, and transportation of people, supplies, materials, and equipment for supporting contingency operations*". The fundamental characteristics of the contingencies are the uncertainty caused by the unexpected events and the significance of the reaction at the prompt time. This implies the 3-*R* principle for contingency operations: providing right things, at the right place and at the right time. In spite of the fact that a considerable number of studies have as of now, been directed in supply chain in terms of its financial profitability, few have concentrated on the reliability of the contingency logistics networks where guaranteeing mission achievement is much more critical than the expenses.

Thomas (2004) appears to be the first one who explicitly considers the reliability of a contingency logistics system from the perspective of a defined mission where he characterizes a contingency as an unexpected crisis that generate a considerable threat to the wellbeing and security of a population. As he indicates, there is a variety of contingency circumstances ranged from military clashes that oblige engagement with hostile forces, police activities for civil disorders, to evacuate victims from catastrophes, such as earthquakes, hurricanes, tsunami and related disasters.

Contingency events trigger a quick requirement for logistics support functions to sort out and activate individuals, equipment, materials and supplies for conducting contingency operations to recover from the contingency. The importance of *3R* principle for contingency operations can be perceived through the contingency case of Hurricane Katrina in New Orleans in 2005, where the lack of complete reaction to it has been debated for a long time, which also has emphasized the importance of design and planning for contingency logistics networks.

The logistics support functions for contingencies can be organized as a network of supply chain exercises for accepting, transporting, and circulating materials and equipment to guarantee "the right things are at the right place and at the right time." An integral part of contingencies are uncertainties and vulnerabilities they create.

Thomas (2004) models a contingency system from the perspective of the reliability of a mission of interest. He considers a set of operational sites that oblige supplies from a central Distribution Center (DC) in support of the contingency operations. He models the life cycle and relative resource profile, $R(t)$, for a contingency loading with three stages: deployment, sustainment, and reconfiguration as shown in Figure **(2.1)**.

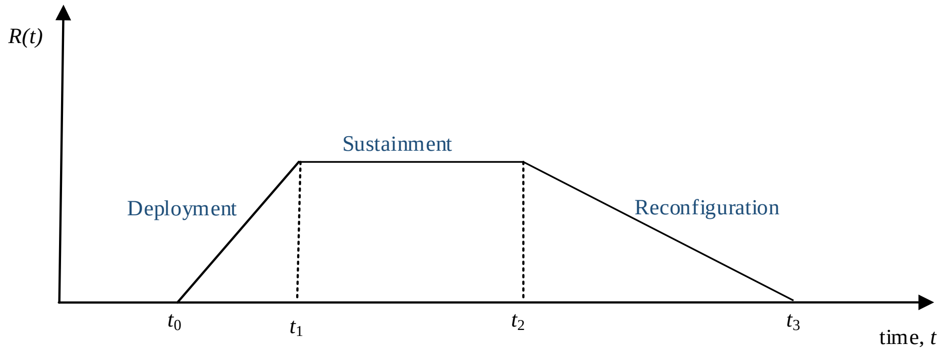


Figure 2.1. Resource Loading Profile for a CLS (Thomas, 2004).

To focus the reliability of the CLS, Thomas (2004) regards each site as an operational node in the network that is expected to perform a set of operations for the mission to recover from the contingency. He uses interference theory between demand and supply to assess the reliability of the site, which is assumed to hold no stocks. The network reliability is, then, computed according to the arrangement of sites in performing contingency operations, which can be characterized by a reliability structure function.

Thomas (2004) investigated the objectives of the logistics networks models considered in past studies and pointed out that most of the models consider the cost of stock allocations, or profitability of networks and there was a void in the literature for a model that explicitly considers the mission success for contingency operations. In general, a network is consisting of a set of nodes and connections between nodes. He considers a specific instance of a CLN, where there is one central distribution center (DC) that supplies material to a set of bases, each of which is fed through a separate link from the DC. His model does not allow links between bases, hence the lateral shipments among bases are not possible. Thomas (2004) first provides the methods to assess the reliability of the nodes (sites, bases) in a CLN. He uses the interference theory between the demand of a base and the supply shipped to the base from the DC. The probability that the demand required by the base is less than amount supplied to the base is assessed as the reliability of the base. He derives set of formulas for the reliability of a base considering a set of probability distributions for the demand of the base, and the supply to the base. In case of limited data about demand and supply, he suggests the use of maximum entropy principle to assess the reliability of the base. Eventually, he computes the network reliability based on the configuration of bases in the network.

Thomas's (2004) preliminary model was extended by Miman (2008) in a variety of ways that provides greater insights for contingency logistics networks planners. First of all, Miman (2008) allowed bases in a CLN to hold stocks before a contingency event occurs. This triggers the stock allocation problem in a CLN to maximize the network's reliability, i.e. the probability of mission success to recover from the contingency in the network subject to available budget and stocks to be used for the allocation. He provides importance measure for a base in terms of the stock it holds for a series and parallel network structure and proposes the use of this importance measures based on the approach defined by Birnbaum (1969) in stock allocation model that can be described as a multi-dimensional knapsack problem. He demonstrates the use of importance measures for a small-size problem.

Since numerous contingency networks have complex structures as well as there may not closed-from expression existing for the reliability of a base due to probability distributions used for the associated demand and supply, assessing the reliability of a site based on interference theory may not be possible or it can be cumbersome. For such cases, Miman (2008) also provides the guidelines of the use of Monte Carlo simulation to assess the node's reliability. He demonstrates that the results of Monte Carlo simulation is matching those obtained from the interference theory for assessing base reliability as well as overall network reliability for the known cases, hence claims that Monte Carlo simulation is a promising approach to assess the network as well as node reliability for cases, especially when the demand and supply distributions do not allow closed form formulas to assess the base's reliability and/or the network structure is too complex to be evaluated. Further, to cope with the uncertainties associated with demand and supply as well as the difficulties associated with the use of interference theory models, an approximation measure is introduced by Miman (2008) based on the mean value first order, second moment technique in modeling the risk of a logistics node. In addition, a decision maker's tolerance for the risk associated with a node failure is incorporated through the use of distortion, whose basis constructed by McLeish and Reesor (2003), particularly dual power (DP) and proportional hazard

(PH) distortion functions that can be regarded as utility functions depending on the decision maker's risk averseness to the specified probability of failure of a node. Distorted risks are basically considering risks higher than they actually are so that one can be more cautious for contingencies in designing the CLN.

Apart from stock allocation models proposed by Miman (2008), a set of optimization model for the acquisition of appropriate risk mitigation systems to improve bases' reliabilities, hence the probability of success in the CLN is also provided by Miman and Pohl (2008). In their approach to risk mitigation models for a CLN, they consider the risk mitigation systems where each has a certain effect on the risk of failure, i.e., the probability of being incapable of performing contingency operations properly and that acquisition the of each risk mitigation system decreased the risk of base's failure by a specific proportion. The goal in their model is to decide what percent of each of the available risk mitigation systems is to be acquired with a given budget as there is a certain cost of acquisition of each risk mitigation system proportional to the amount of it acquired by the decision makers, probably CLN planners.

Another contribution by Miman (2008) is the development of sustainability model for the CLN that considers the reliability of links explicitly in addition to the reliability of nodes. He considers the links existing between the central DC and each of the operational bases that are conceptualized as transportation means of supplies from the DC to the bases. The life time of each link is modeled through the Weibull life distributions and a multi-action selective maintenance model is embedded into the sustainability model. The objective of this model is to identify the best set of maintenance actions to be performed within the available budget and time. His model, whose details are provided in Section (3.1), constitutes the integral part of this thesis that investigates the optimization techniques for it as provided in Section (3.2) and evaluation of these algorithms as provided in Section (3.3).

Miman and Pohl (2012) realized the need for considering the simultaneous optimizations of reliability of the CLN, cost of maintenance actions performed and total time required to perform maintenance actions for the sustainability model in identifying the best set of maintenance actions to perform on each of the links between the DC and each of the bases. In their multi-objective model, the reliability of the network to be maximized, the total cost and total time for the maintenance actions to be minimized through the physical programming (PP) where decision makers set the desirability levels for each of the three objectives explicitly. They argue that this modeling approach can fit to the contingency environment as the inherent uncertainties related to contingencies can be treated through the desirability levels for each criteria to explicitly model where the decision maker wants to find/or does not want to find themselves in terms of these three objectives.

2.1.3. Contingency Logistics Networks

A contingency is any kind of disaster as explained in Section (2.1). Logistic networks involve warehouses, vehicles and every kind of transportation tools. When these are used for a possible contingency it is called a Contingency Logistics Network (CLN). In this study, CLN and CLS can be used interchangeably as it is done the Section (2.1.2). This section provides the details of some of the previous studies conducted such as importance measures, distorted risks and so on to make conceptualization of this study more concrete.

2.1.3.1. Importance Measures for CLN

A representation of a supply chain network for a basic supply component, for example, ammo, is given in Figure **(2.1)** under the concept of CLN design based on Miman (2008).

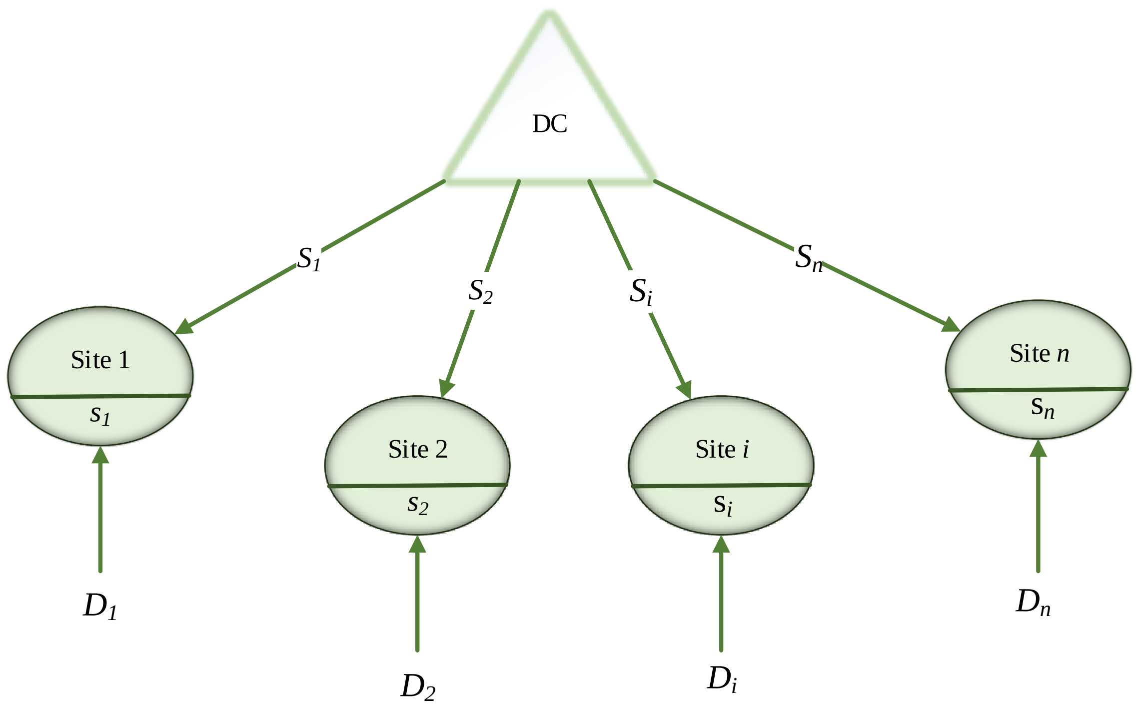


Figure 2.2. Basic CLN with Stocks.

In Figure **(2.2)**, a supply component, for instance ammo, is circulating from a central distribution center that constitutes a higher echelon level, and provides control support for the ammo to *n* operational sites. In case of contingency event, specifically a terrorist attack, there is a mission of removing this threat which requires performing a set of operations by bases. Therefore, some amount of ammo is demanded by bases, which are uncertain in the design stage of the CLN. Similarly, during the contingency the amount that can be supplied from DC to each base is also uncertain.

Thomas (2004) deals these uncertainties in demand and supply through probability distributions with enough data, and maximum entropy principle with limited data. The use of this principle enables him to maximize the reliability of the logistic network under chaotic circumstances and for the most pessimistic scenario. Miman (2008) allows bases to hold "safety" stock. That is, operational sites are provided with a few materials before a real crisis happens. In fact, during the contingencies characterized by uncertainties, it is easy to have materials allocated before the contingency rather than to ship them from a DC to the base at the time of contingency.

The risk associated with the node i of the network, which is assigned to perform a set of operations, is modeled by utilizing the interference theory where the risk, ρ_i, $\rho_i = Pr\ \{ S_i < D_i \}$, for the node i is defined as the probability that the node i has lack of supplies i.e., $\{ S_i < D_i \}$. By conditioning on D_i, then the probability of failure of site i is obtained as it is in Equation **(2.1)**.

$$\rho_i = \mathrm{P_r}\{S_i < D_i\} = \begin{cases} \sum_y X_i(y) g_i(y) & \text{for discrete } (X_i, Y_i) \\ \int_y X_i(y) g_i(y) & \text{for continuous } (X_i, Y_i) \end{cases} \qquad \textbf{(2.1)}$$

The importance measure for site i, regarding the parameter p_i, can be characterized utilizing Birnbaum's (1969) methodology where the first order derivative is utilized:

$$\zeta_i = \frac{\partial R}{\partial p_i} \qquad \textbf{(2.2)}$$

As ζ_i shows the impact of a unit difference in the failure parameter for site i, p_i on the reliability of the supply chain, R. This is a measure of the sensitivity of the whole system to differences in the parameter p_i. Note that, Miman (2008) provides importance measures for site i in terms of mean demand, mean, supply and stock hold at that site.

2.1.3.2. Distorted Risks in a CLN

The consequences of a disaster in a contingency are considerably more severe than those of ordinary everyday events. Therefore risks related to contingencies cannot be ignored. Miman and Pohl (2008) handles this by applying the distortion functions to the risk of failure of each node in the network designed to perform missions to recover from the contingency. This enables the CLN planners to reflect decision makers' attitudes towards the risk of failure similar to utility functions.

The most general form of distortion functions is gamma-beta (GB) distortion as provided by Offut *et al.*, (2006) in Equation **(2.3)**. GB distortion family, by setting its parameters, provides specific distortions such as proportional hazard (PH) and dual power (DP) in Equation **(2.4)** and Equation **(2.5)** respectively.

$$
\begin{aligned}
& h(x) = d_{GB}(h(x)) = \int_0^{h(x)} K\, t^{a-1} (1-t)^{b-1} e^{-t/c}\, dt \\
& \textit{where} \\
& K^{-1} = \int_0^1 t^{a-1} (1-t)^{b-1} e^{-t/c}\, dt
\end{aligned}
\tag{2.3}
$$

$$
h^{PH}(x) = d_{PH}(h(x)) = h^{a}(x), \quad 0 \le a \le 1 \tag{2.4}
$$

$$
h^{DP}(x) = d_{DP}(h(x)) = 1-(1-h(x))^{b}, \quad b \ge 1 \tag{2.5}
$$

Miman and Pohl (2008) applies PH and DP distortions to the risk of failure of node *i* in the CLN, and consequently gets distorted risks as in Equation **(2.6)** to reflect the decision maker's risk tolerance. It is noteworthy that a distorted risk is higher than the actual probability of failure, as the perceived risks can be higher for a risk-averse decision maker. The use of distortion to the risks in the CLN, introduced by Miman and Pohl (2008), provides a greater flexibility to plan the CLN.

$$
\begin{aligned}
\rho_i^{PH} &= (pi)^{a}, \qquad 0 \le a \le 1 \\
\rho_i^{DP} &= 1-(1-pi)^{b}, \qquad b \ge 1
\end{aligned}
\tag{2.6}
$$

The illustration of application of distortion is presented in Figure **(2.3)**.

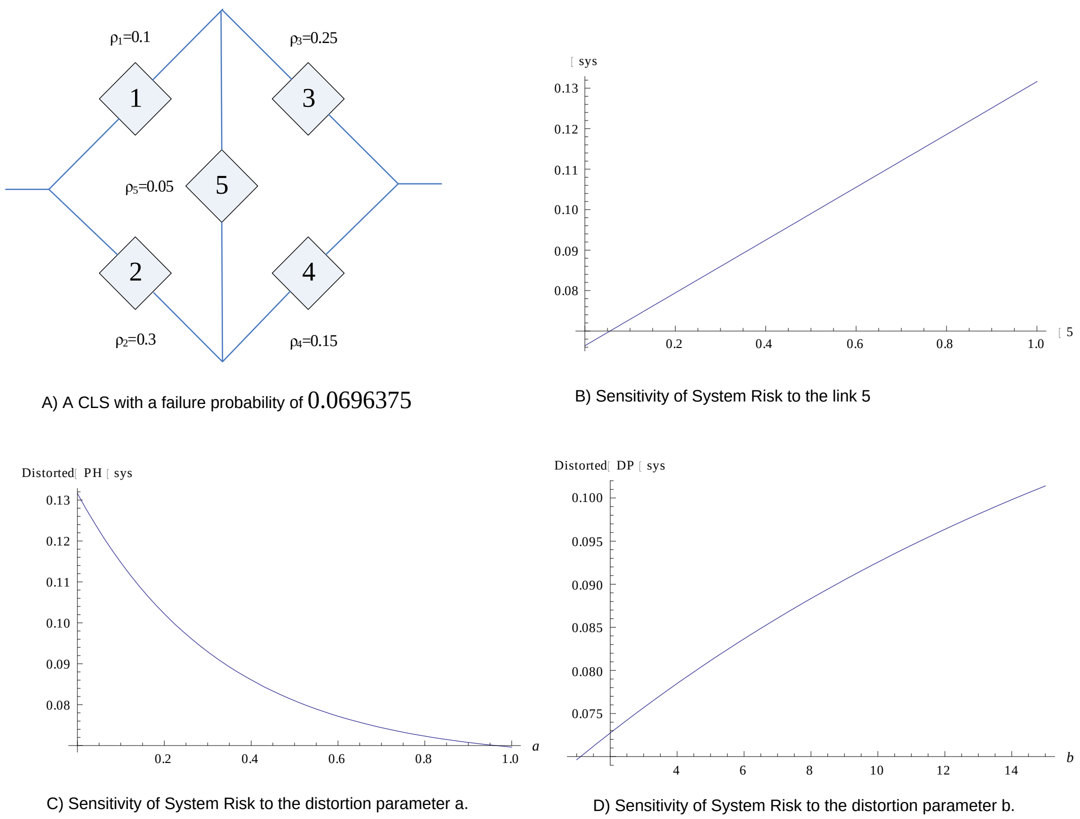


Figure 2.3. Distorted Link's Effect on the System (Miman, 2008).

Figure **(2.3B)** represents the linear relationship between the system's risk and the risk of component 5 for the network structure demonstrated in Figure **(2.3A)**. In the event that the component is more vital for the logistics network planner, or the vulnerabilities on that component are not evaluated as completely as could be allowed, the distortion can be utilized to reflect the CLN planner's attitude towards the risk of failure of that component.

When a PH distortion is applied, as it is in Figure **(2.3C)**, as the distortion parameter *a* decreases, the perceived risk increases. Miman and Pohl (2012) provide how the values of distortion parameters can be set according to decision-maker's attitude towards risks as follows:

For as risk-neutral decision maker: $a = 1$; $b = 1$

For a risk-seeking decision maker: $a > 1$; $b < 1$

For a risk-averse decision maker: $a < 1$; $b > 1$

They indicate that the selection of distortion parameters for a risk of failure of a base depends on;

- Criticality, usefulness of the node;
- Level of uncertainties associated with supply and demand
- Decision maker's attitudes towards risk

2.1.3.2.1. Distorted Risk Measures (Series and Parallel)

Miman and Pohl (2008) provide importance measures for the distorted risks, as ordinary importance measures may be misleading for low-probability events that have severe consequences depending on the network structures where sites are arranged either in series (Equations **(2.7)-(2.8)**) or in parallel (Equations **(2.9)-(2.10)**):

Series are;

$$\left\{ i^* \middle| \max\left\{\zeta_i^{PH}\right\}, i = 1,2,\ldots,n \right\} = \left\{ i^* \middle| \max\left\{ \frac{a_i}{1-\rho_i^{a_i}} \rho_i^{a_i-1} \right\}, \quad i = 1,2,\ldots,n \right\} \qquad \textbf{(2.7)}$$

and

$$\left\{ i^* \middle| \max\left\{\zeta_i^{DP}\right\}, i = 1,2,\ldots,n \right\} = \left\{ i^* \middle| \max\left\{ \frac{b_i}{1-\rho_i} \right\}, \quad i = 1,2,\ldots,n \right\} \qquad \textbf{(2.8)}$$

Parallels are;

$$\left\{ i^* \middle| \max\left\{\zeta_i^{DP}\right\}, i = 1,2,\ldots,n \right\} = \left\{ i^* \middle| \max\left\{ \frac{a_i}{\rho_i} \right\}, \quad i = 1,2,\ldots,n \right\} \qquad \textbf{(2.9)}$$

and

$$\left\{ i^* \middle| \max\left\{\zeta_i^{DP}\right\}, i = 1,2,\ldots,n \right\} = \left\{ i^* \middle| \max\left\{ \frac{b_i\,(1-\rho_i)^{b_i-1}}{1-(1-\rho_i)^{b_i}} \right\}, \quad i = 1,2,\ldots,n \right\} \qquad \textbf{(2.10)}$$

2.1.3.2.2. Risk Displacement Factor

Risk displacement factor associated with a distortion is computed through Equation **(2.11)** and Miman and Pohl (2008) provides the risk and risk displacement factors associated with proportional hazard and dual power distortions applied to risks emerged from a variety of probability distributions for demand and supply as tabulated in Table **(2.1)**.

$$R_d = \frac{\{x > 0 : h(x) = 0.5\}}{\{x > 0 : h(x) = 0.5\}} \tag{2.11}$$

Table 2.1. Displacement Factors through Distortion (Miman and Pohl, 2008).

Case	*Supply*, $f(x)$	*Demand*, $g(y)$	*Failure Probability*, r	R_d^a	R_d^b
a	$\mu e^{-\mu x}$, $x \geq 0$	$\lambda e^{-\lambda y}$, $y \geq 0$	$\frac{\mu}{\mu+\lambda}$	$\frac{1}{e^{\ln 2/a}-1}$	$e^{\ln 2/b}\left(1-0.5^{\frac{1}{b}}\right)$
b	$\frac{1}{\sigma_X\sqrt{2\pi}} e^{-\frac{x-\mu}{2\sigma_X^2}}$, $-\infty < x < \infty$	$\frac{1}{\sigma_Y\sqrt{2\pi}} e^{-\frac{y-\lambda}{2\sigma_Y^2}}$, $-\infty < y < \infty$	$\Phi\left(\frac{\lambda-\mu}{\sqrt{\sigma_Y^2+\sigma_X^2}}\right)$	$1-\frac{\sqrt{2}\left(\sigma_Y^2+\sigma_X^2\right)\text{InverseErf}[1-2(}{\mu\sqrt{\sigma_Y^2+\sigma_X^2}}$	$1-\frac{\sqrt{2}\left(\sigma_Y^2+\sigma_X^2\right)\text{InverseErf}[2(0.}{\mu\sqrt{\sigma_Y^2+\sigma_X^2}}$
c	$\beta(1-\beta)^x$, $x = 0,1,\ldots$	$\alpha(1-\alpha)^y$, $y = 0,1,\ldots$	$\frac{\alpha}{\alpha+\beta-\alpha\beta}$	$\frac{1}{e^{\ln 2/a}-1}$	$e^{\ln 2/b}\left(1-0.5^{\frac{1}{b}}\right)$
d	$\frac{\mu^r x^{r-1} e^{-\mu x}}{(r-1)!}$, $r = 1,2,..,;x \geq 0$	$\lambda e^{-\lambda y}$, $y \geq 0$	$\left(\frac{\mu}{\mu+\lambda}\right)^r$	$\frac{e^{\ln 2/r}-1}{e^{(\ln 2)r/a}-1}$	$\frac{e^{\ln 2/r}-1}{\left(1-0.5^{\frac{1}{b}}\right)^{-\frac{1}{r}}-1}$
e	$\mu e^{-\mu x}$, $x \geq 0$	$\frac{\lambda^r y^{r-1} e^{-\lambda y}}{(r-1)!}$, $r = 1,2,..,;y \geq 0$	$1-\left(\frac{\lambda}{\mu+\lambda}\right)^r$	$\frac{\left(1-0.5^{\frac{1}{a}}\right)^{-\frac{1}{r}}-1}{\left(1-0.5^{\frac{1}{r}}\right)e^{\ln 2/r}}$	$\frac{1-e^{\ln 2/br}}{1-e^{\ln 2/r}}$
f	$\frac{\mu^r x^{r-1} e^{-\mu x}}{(r-1)!}$, $r = 1,2,..,;x \geq 0$	$\frac{\lambda^r y^{r-1} e^{-\lambda y}}{(r-1)!}$, $r = 1,2,..,;y \geq 0$	$1-\ldots$ $\ldots\frac{\Gamma(2r)}{(\Gamma(r))^2}.*\ldots.$ $\ldots.Beta[-\frac{\lambda}{\mu},r,1-2r]$	*NoClosedForm*	*NoClosedForm*

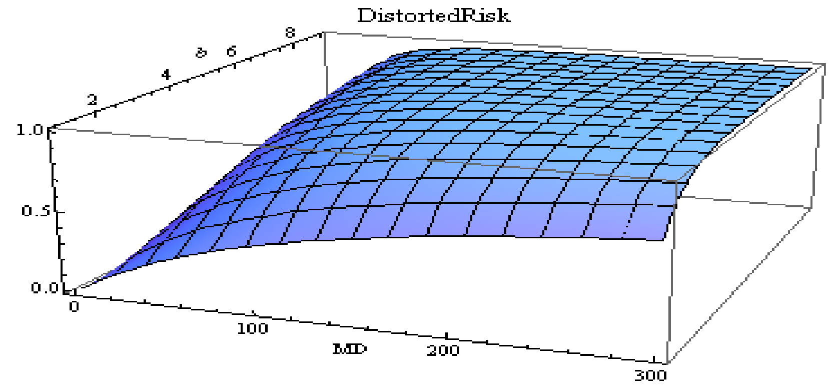


Figure 2.4. Risk Surface under Distortion (Miman, 2008).

Miman (2008) further introduce a new metric, called as natural vulnerability, which is defined as the risk of not accomplishing assigned operations by a node that has only security stock allowed to hold and the extra supply in case of contingency cannot be provided that node. This can be also interpreted through the previous definition of the risk of failure of the node where supply is set to zero.

$$\upsilon i = 1 - G_i\,(s_i) \tag{2.12}$$

Table 2.2. Displacement Factors for Natural Vulnerability (Miman, 2008).

Case	R_d^a	R_d^b
a , d	$\dfrac{1}{a}$	$\dfrac{\ln\left(1-0.5^{\frac{1}{b}}\right)}{\ln(0.5)}$
b	$1+\dfrac{\sqrt{2}\sigma_Y\ \mathrm{InverseErf}[1-2(0.5^{\frac{1}{a}})]}{\lambda}$	$1+\dfrac{\sqrt{2}\sigma_Y\ \mathrm{InverseErf}[2(0.5^{\frac{1}{b}})-1]}{\lambda}$
c	$\dfrac{\dfrac{\ln(2)}{a}+\ln(1-\beta)}{\ln(2-2\beta)}$	$\dfrac{\ln\left(\dfrac{\beta-1}{0.5^{\frac{1}{b}}-1}\right)}{\ln(2-2\beta)}$

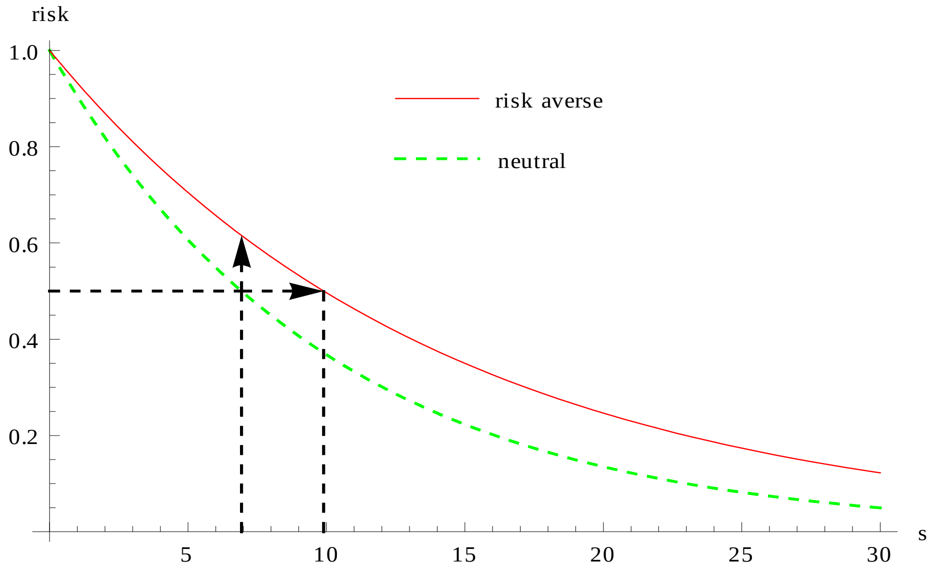


Figure 2.5. Illustration of Replacement through Distortion (Miman, 2008).

2.2. Supply Chain and Risk Analysis for Contingencies

There have been several studies for the general emergency situations and studies that address the risk and uncertainty assessment in contingency networks in addition to the cost benefit analysis and one-for-one replenishment of spare parts in the emergency operations (Miman, 2008). For instance, Gupta and Bhattacharjee (2012) give more prominent experiences on adaptable logistics networks where the qualitative risk assessment of the supply chains is considered, hence, that can be connected with contingency operations.

Parlar (1997) considered a stochastic inventory model in which the supply available is subject to randomness that may result from machine breakdowns, strikes, embargoes, and so on. Their model assumes that the inventory managers deals with two suppliers who are either individually ON (available) or OFF (unavailable) where each supplier's availability is modeled as a semi-Markov (alternating renewal) process.

Archibald *et al.* (1997) considered transshipments for contingency cases using the general inventory methodology. Specifically, they considered a multi-period and periodic review model of two location inventory systems in which emergency transshipments can occur in case of stock outs.

The contingency oriented administration is a powerful view of arranging, sorting out, and controlling that is customary to the settings and specific circumstances confronted by the organization (Wren, 1994). For this, there would not be a "one-best path" to oversee. However, there are a few promoters of the "best practice" approach (Lee and Siau, 2001).

The contingency approach connected to supply chain administration would accept that there is no general response to accomplish as relevant elements and circumstances change and they change over the long run. That is, discovering or keeping up the best procedure is troublesome in today's quickly changing business environment. As indicated by Chow *et al.* (1995), lack of the one-best path way to deal with disruptions in a supply chain shows that alternatives, such as the contingency theory connected to supply chain administration could end up being more helpful case for investigation (Bowersox *et al.*, 1999 and Bowersox and LaHowchic, 2008).

One considerable and exceedingly noticeable commitment to deal with supply chain administration from the perspective of contingencies originates from Fisher (1997). Fisher argues that the supply chain structure and administration has to be anticipated for the types of products being produced and delivered through them. He categorizes the products as *functional* or *innovative*. Commoditized products that normally satisfy basic needs such as staples can be given as an example of functional products. The main characteristics of functional product are not changing much over time, having lower profit margins, longer life cycles and low forecast uncertainty. Innovative products possess just opposite characteristics having frequent product launches and higher profit margins, shorter life cycles and less predictable demand. Electronics products such as cell phones provide good examples for such innovative items (Lee *et al.*, 2007).

The recent researches are generally based upon the pioneer models by Gross (1963) which determine the optimal redistribution and replenishment policies for a two store inventory system. Later, Krishan and Rao (1965) determine the optimal stock levels which minimize the one period inventory costs and transportation costs with emergency transshipments.

2.3. Relevant Techniques for Analysis of a CLN

This section provides a set of techniques, including analytical approaches, simulation approaches, importance measures, optimization as a knapsack problem, multi-objective optimization, heuristic optimizations in analysis and modeling of contingency logistics networks.

2.3.1. Analytical Approaches to CLN Design

Kukreja and Schmidt (2005) analyze a model for lumpy demand parts in a multi-location inventory system with transshipment by using analytical and simulation techniques. They provide results for the mean and variance of the lead-time demand at various locations analytically and then use simulation methodology to determine inventory control policies for such a system Wong *et al.*, 2006 propose an integer-programming problem with a nonlinear objective function and non-linear constraints for multi-item multi-location spare parts systems with lateral transshipment and waiting time constraints. They conclude that among the four different heuristics they developed, greedy-type heuristic has the best performance in terms of their total costs and computation times.

In general, it can be said that the analytical approaches handle costs and times for the logistics operations. Mathematical models and Markov process are some of the methods that can be encountered in the literature. Confronted with randomness and uncertainties by the decision-maker, fuzzy logic can appear as a promising way in decision making (Bevilacqua *et al*., 2006). Fuzzy theory, a theory to act in situations of uncertainty provides important tools for decision makers based on probability theory (Analytis *et al*., 2014), can be considered in case of contingency.

2.3.2. Simulation Approach to CLN Design

In general, multi-echelon supply chain problems involves with high level of complexity, that makes analytical approaches impossible or cumbersome to be applied. This triggers the need for simulation techniques to put more realism to the models. Needham and Evers (1998) investigate the interaction of relevant costs and transshipment policies via simulation study and present a method for determining a threshold value at which the benefits of transshipments outweigh their costs. They also propose a meta-model to provide greater insights into cases where emergency transshipments should be employed. Their solution procedures based on infinitesimal perturbation analysis (IPA), is claimed to be an efficient simulation-based optimization technique (Ho. *et al.*, 1979).

2.3.3. Importance Measures and Sensitivity Analysis

In the literature there is a variety of sensitivity analysis techniques based on the first order differential analysis developed by Birnbaum (2000), Lambert *et al.*, (1998) and variations of these techniques, available for analyzing the reliability of a system. Some of the other techniques provide rank improvement actions (Xie and Shen, 1989) and an importance measure for the reliability of a *k*-out-of *n* system (Barlow and Proschan, 1975).

Variance decomposition methods that partition the variance of the systems reliability estimate among the affects due to variance of the components-reliability estimate are other techniques used in the literature to identify which component contributes most to the variance of the system reliability estimate, hence needs improvement. Coit and Smith (1996b) uses these methods for systems of mutually independent components arranged either in series or parallel. Further, Coit and Smith (1996a) also analyze redundancy allocation problem through Genetic Algorithm as well. In a similar way, Coit and Jin (2001) provide a ranking methodology to prioritize testing so as to minimize the system variance.

Miman (2008) indicates that the maintenance of transportation links along with the operational nodes can affect the reliability and availability of the logistic network. Cassady *et al.* (2004) develop an importance measure for the availability estimate of a two state (functioning and failed) repairable system whose components have exponential failure and repair distributions. Miman and Pohl (2006) provide an assessment technique which utilizes the delta-method to provide an estimate of the variance for the system's availability, consequently, they construct a variance importance measure so that one can have an improvement in the variance of the system level availability estimate through the reduction of the variance of the component availability estimates using this importance measure. In addition, they developed a cost model that trades-off cost and uncertainty. They conclude that the variance importance measure provides results that are consistent with reliability importance measures developed by Coit and Jin (2001).

Simulation is another technique for the sensitivity analysis of a system. For instance, Snyder and Shen (2006) used simulation to investigate the differences between demand and supply uncertainty in multi-echelon supply chains. They concluded that the two types of uncertainty are mirror images of each other. That is the optimal strategy for dealing with supply uncertainty and may be just the opposite for the strategy used to cope with demand uncertainty. Brauers and Zavadskas (2010) considered a simple multi-echelon supply chain, first under demand uncertainty and then under supply uncertainty in the form of disruptions in search of best order frequency, inventory placement, or supply chain structure.

2.3.4. Optimization-through Knapsack Approach and Multi-Objective Analysis Techniques

Miman (2008) indicates that the stock allocation along with the selective maintenance problem requires an investigation of appropriate optimization approaches used for resource allocation problems. In many industrial and military agencies, one of the major cost component is money that is being spent on equipment maintenance activities. Cassady and Nachlas (1998) indicated that productivity can be increased up to thirty percent by implementing better maintenance policies.

Cassady *et al.* (2001) define the selective maintenance problem as "*the process of identifying a set of maintenance actions to be performed from a set of potential maintenance actions, given a set of limited resources*". In general, this problem can arise for a system that performs a sequence of missions with a certain amount of resources (time, budget, etc.) allocated for maintenance between successive missions. Robertson and Ulrich (1998) classify these maintenance activities as either corrective maintenance (CM) that refers to restoring failed equipment to a functioning condition or preventive maintenance (PM) that refers to performing maintenance on functioning equipment for the purpose of delaying future failures.

A list of all potential corrective and preventive maintenance actions is prepared at the end of each mission, and after which, system managers should identify the best set of actions that are particularly desirable due to the limited resources that do not allow actions to be performed practically. There are various studies in the literature that consider selective maintenance problem. Rice *et al.* (1998) developed a mathematical programming model, whose objective is to maximize the system reliability for a single mission, with maintenance time and maintenance cost as constraints, and use total enumeration to find an exact solution to the basic selective maintenance problem for a single mission.

Cassady *et al.* (2001) extend the preliminary selective maintenance problem to permit any system structure considering reliability, maintenance cost and maintenance time. They provide two new extensions by introducing time-dependent failure rates for components and multiple maintenance options for which life time of the components are characterized by Weibull probability distributions. They consider three different type of maintenance options – minimal CM, perfect PM and replacing failed components –in their models.

Cassady and Pohl (2002) provide two extensions to the basic selective maintenance model by Rice *et al.* (1998) by introducing the acquisition of extra maintenance time at some cost during the break period and setting the optimum maintenance time between breaks for a new system where the objective is to identify this extra time with a minimal cost. The second extension they provide is the use of Monte Carlo simulation to determine the optimum time allocated between the breaks.

Pohl *et al.* (2005) model the selective maintenance problem by expanding upon earlier models such that there is an opportunity for a decision maker to replace a component with an upgraded component. In their model, one has the option of replacing with current technology vs. the original technology during maintenance.

The original selective maintenance problem of Rice *et al.* (1998) is modeled as a non-separable, non-linear knapsack problem with discrete decision variables. This class of knapsack problems presents a tremendous challenge with respect to identifying optimal solutions even when the decision variables are continuous (Bretthauer and Shetty, 2002). This is the reason explained by Rice *et al.* (1998) why they use total enumeration algorithm to solve the selective maintenance problem they formulated.

As the total enumeration strategy is not practical for reasonably sized problems, in the literature there are a variety of studies that explore approaches to address this issue. For example, Mathur *et al.* (1986) consider a general non-linear knapsack problem with a concave objective function and a single convex constraint and suggest the use of an implicit enumeration technique as a solution procedure. Kodialam and Luss (1998) deal with a simple resource allocation problem with a single resource constraint where the objective as well as resource usage functions are separable and convex. Ohtagaki *et al.* (2000) propose a heuristic approached, namely Smart Greedy, to solve a multi-dimensional nonlinear knapsack class of reliability optimization problems. Kevin and Sancho (2001) suggest the use of a hybrid 'dynamic programming/depth-first search' algorithm to solve non-linear programming problems resulted from redundancy allocation in reliability optimization. Overall, all of these approaches that can be employed to optimize the CLN design can be categorized into three groups: heuristics, approximations and exact methods for which Miman (2008) provide an extensive literature review as described in the remaining of this section.

As explained by Miman (2008), heuristics are intuitive methods to find a local or a near optimal solution (approximate optimal solution) by gradually improving an incumbent solution. Steepest ascent methods using sensitivity factors, boundary region search, increasing redundancy on minimal path sets, and metaheuristics can be listed in this category. The main characteristics of them are having a relatively short time of execution and providing reasonable solutions, which generally cannot be guaranteed to be global optimal. Authors who suggest the use of heuristic solutions often use metaheuristics, particularly, simulated annealing (SA), and genetic algorithms (GAs). (This is, in fact one of the motivation of this research to apply them as well as to develop a hybrid heuristic based on these two algorithms). For instance, Coit and Smith (1996b) developed a problem specific GA to analyze series parallel systems to determine the optimal design configuration when there are multiple component choices available for each of several k-out-of-n: G subsystems in terms of reliability allocation. Pohl *et al.* (2007) developed metaheuristics solutions to the multi-action selective maintenance problems of practical size. They conclude that MULRR is more robust and efficient while GAs require more tuning, though they are likely to have considerable potential with problem specific subroutines in mutation and crossover operators.

For close estimation methods, a casual problem which has a bigger possible district contrasted with the first problem is determined and its ideal arrangement is adjusted off to a number arrangement. To tackle the casual problem, scientific programming strategies, for example, linear and nonlinear programs are regularly utilized. The principle weakness of this kind of calculation is that it obliges superb data, e.g., subordinates and scientific reformulation and accordingly, execution discriminatingly relies upon the system structure.

For approximation techniques, a relaxed problem of larger feasible region than that of the original problem is solved and its optimal solution is rounded to an integer solution (Miman, 2008). Generally, mathematical programming techniques, such as linear, nonlinear, and geometric programming are used to solve the relaxed problem. He claims that the main disadvantage of these types of algorithms is the requirement of high quality information, such as derivatives and mathematical reformulation; hence, the system structure affects the performance of these algorithms considerably.

Combinatorial optimization techniques can be used for global optimization methods, some examples of which are exact methods including dynamic programming, implicit enumeration, and branch-and-bound. The main characteristics of these approaches are guaranteeing global optimal solutions, however, being very exhaustive, their efficiency depends on the efficiency of the search space elimination. Ha and Kuo (2006) present an efficient branch-and-bound approach to solve non-linear, non-convex and non-separable types of discrete knapsack problems, where the system of their interest is coherent, i.e., the objective and constraint functions have monotonic increasing properties.

Apart from them, column generation has been known to be one of the most successful approaches for solving large scale integer programming problems over the last decade as used by Gilmore and Gomory (1961), Minoux (1987), Vance *et al.*, (1994), Desrosiers *et al.* (1990), Vanderbeck and Wolsey (1996) and Zia and Coit (2006).

2.3.5. Metaheuristics Optimization Techniques

This section provides information about metaheuristics optimization techniques including general optimization lexicon; characteristics of metaheuristics; a brief history of metaheuristics; metaheuristics that are used often such as simulated annealing (SA), genetic algorithms (GA), different evolution (DE), ant colony optimization (ACO), bee algorithms, particle swarm optimization (PSO), tabu search (TS), harmony search (HS), firefly algorithm (FA), cuckoo search (CS), water cycle algorithm (WCA), other metaheuristics algorithms and a general design example for metaheuristics based on work by Yang (2011).

The metaheuristics optimization treats optimization problems utilizing metaheuristics algorithms. Optimization using metaheuristics can be conducted for a wide range of areas from building configuration to financial aspects. The main considerations for their application are to provide a best evaluated solution with a reasonable objective function value by identifying the best set of decision variable under the limited resources such as available time and budget.

The general form of realistic optimization problem is known to be nonlinear and multi-modal, under different complex constraints. Many times there can be competing objectives such as maximizing reliability of a CLN while minimizing cost of maintenance actions to be performed and minimizing the time required by selected maintenance activities. This makes the process of identifying the best set of decision variables difficult and not straight forward.

One of the simplest optimization problems can be considered as a minimization or maximization problem. For instance, the function, $f(x) = x^2$, has an optimal value $f_{min} = 0$ at the solution of $x = 0$ in the domain of $-\infty < x < \infty$. For a sufficiently simple function, the first order differential $f'(x) = 0$ can be utilized to focus the potential areas in the solution space, the second differential $f''(x)$ can be used to check if the solution is a most extreme or least. However, for nonlinear, multi-modal, multi-variate functions, this approach may not be easy to use or may not be appropriate to be undertaken. Similarly, some of the functions may have discontinuities, and accordingly differentials is not easy to obtain.

2.3.5.1. General Optimization Lexicon

Dantzig (1963) provides a general lexicon for an optimization problem as illustrated in Exhibit **(2.1)**.

$$f_1(x),\ldots,f_i(x),\ldots,f_I(x), \qquad x=(x_1,\ldots,x_d)$$

$$\textit{subject to}$$

$$h_j(x) \;=\; 0,\; (j=1,2,\ldots,J)$$

$$g_k(x) \;\le\; 0,\; (k=1,2,\ldots,K)$$

Exhibit 2.1. General Lexicon for Optimization Problems, (*P1*), (Dantzig, 1963).

In Exhibit **(2.1)**, $f_1,\ldots,f_I$ are the objectives, while h_j and g_k are the equality and inequality constraints, respectively. The special case of $I = 1$ is called as single-objective optimization while the case where $I \ge 2$ makes the optimization problem multi-objective problem whose optimal solution procedures are different than those for a single objective problem. He concerns single-objective optimization problems. If at least one of the functions f_i, h_j and g_k in the optimization problem are nonlinear in terms of decision variables represented by *x*, the optimization problem is said to be nonlinear. In the special case where each of these functions is linear in terms of decision variables, the optimization problem turns into a linear programming problem which can be solved through the standard simplex strategy (Dantzig 1963).

Most of the metaheuristics optimization is concerned about nonlinear optimization problems. In general the maximization representation of model can be turned to the minimization representation or vise versa easily. That is, one modeling paradigm can also imply the other. As it is expected, the most straightforward instance of an optimization is unconstrained function optimization.

To treat optimization problems, effective approaches or optimization algorithms are required. In the literature, there are numerous optimization algorithms that can be classified in a variety of ways according to their focus and characteristics.

Gradient-based algorithms where the derivative or gradient of a function is the focus such as hill-climbing, use derivative information efficiently. On the other hand, derivative-free algorithms such as Nelder-Mead downhill simplex do not use any information but the values of the function itself instead. They are very useful in case where derivatives do not exist or it is very cumbersome and expensive to calculate derivatives accurately. Depending on the number of agents/solutions an algorithm uses and, hence the number of paths it traces out as the iteration continues, algorithms can be classified into trajectory-based (ones with a single solution such as hill-climbing and SA) or population based (ones interacting multiple solutions such as PSO) according to Kennedy and Eberhardt (1995).

An algorithm can also be classified as either deterministic (such as hill-climbing and downhill simplex) if it works without any random nature, hence, it always find the same solution provided that it starts with the same initial solution or stochastic (such as GA or PSO) if there is a randomness in the algorithm, hence, in general it is likely that the algorithm end up with a different solution every time it is executed even when the same initial solution is used for each execution. Besides, depending on its search capacity, the algorithm can be classified as local search (such as hill-climbing) if it converges towards a local optimum or global search if it converges towards a global optimum. In general, randomization is regarded to be an efficient component for global search algorithms.

An algorithm may not necessarily to fit into one of the categories specified above, but, it can be a combination of one algorithm with another, regarded as hybrid algorithm, in order to design more efficient algorithms. This idea motivates the development of the hybrid algorithm, EDGASA, whose details are provided in Section (3.2).

2.3.5.2. Characteristics of Metaheuristics

According to convention introduced by Glover (1986) and Glover and Kochenberger (2003), all modern nature-inspired algorithms can be called as metaheuristics. The word *meta* means *beyond* and *metaheuristic* is used for an algorithm to imply that the modified algorithm is developed to perform better (generating better solutions) than simple heuristics (local points) (Glover and Laguna 1997). In addition, metaheuristics are likely to generate quality solutions to difficult optimization problems, but cannot guarantee an the optimal solution. Though, Voss (2001) claims that they tend to be suitable for global optimization.

Blum and Roli (2003) indicate two major components of any metaheuristics: exploration, which means generating diverse solutions generally through randomization to escape from local optima and exploitation, which means focusing the search in a local neighborhood that is likely to produce good solutions. The global optimality is more likely to be achievable through the good combination of these two components.

2.3.5.3. A Brief History of Metaheuristics

Although heuristics methods had been used from the 1940s to 1960s in various applications, the first landmark in the history is provided through the development of evolutionary strategies and evolutionary programming. The first study on genetic algorithms was published by Holland (1975), who developed them in 1960s. Kirkpatrick *et al.* (1983) provided a big step for metaheuristics through the development of simulated annealing (SA) based on the inspiration by the annealing process of metals. This was followed by the development of artificial ımmune systems by Farmer *et al.* in 1986. Glover introduced the use of memory in metaheuristics and tabu search (TS) based on this idea was published by Glover and Laguna (1997). Dorigo (1992) developed ant colony optimization (ACO) technique that was inspired by the swarm intelligence of social ants using pheromone. In 1995, particle swarm optimization (PSO) was developed by Kennedy and Eberhart (1995). Differential evolution (DE) algorithm was developed by Storn and Price in 1997.

This was followed by the development of harmony search (HS) algorithm in 2001 by Geem *et al.* More recently, Nakrani and Tovey proposed the honey bee algorithm and its application for optimizing internet hosting centers in 2004, while was a novel bee algorithm was developed by Pham *et al.*, in 2005 and the artificial bee colony (ABC) was developed by Karaboga in 2005. In 2008, the firefly algorithm (FA) was developed by Yang. Finally, Yang and Deb introduced an efficient cuckoo search (CS) algorithm (Yang and Deb, 2009; Yang and Deb, 2010), demonstrating that CS is far more effective than most existing metaheuristics algorithms.

The most recently, Eskandar *et al.* (2012) developed a water cycle algorithm in 2012 based on the observation of water cycle process and rivers and streams flow to the sea in nature for constrained optimization.

2.3.5.4. Simulated Annealing

Simulated annealing is originated from mimicking the metal annealing processing (Kirkpatrick *et al.*, 1983) and has the ability to escape from local optima by allowing the acceptance of inferior solutions with a certain probability. Dissimilar to the gradient-based routines and other deterministic hunt strategies, the fundamental favorable position of simulated annealing is its capacity to obtain from being caught in neighborhood optima. The formulation of SA presented in this part below is based on (Kirkpatrick *et al.*, 1983).

The acceptance probability p is shown by Equation **(2.13)**.

$$p = \exp\left[-\frac{\Delta E}{k_B T}\right] \qquad \textbf{(2.13)}$$

where k_B is the Boltzmann's constant, T is the temperature for managing the annealing process and ΔE is the difference in energy. This transition probability is built on the Boltzmann distribution in statistical mechanics. The difference in the objective function, Δf, can be bonded to ΔE as in Equation **(2.14)**.

$$\Delta E = \gamma \Delta f \qquad \textbf{(2.14)}$$

where γ is a real constant (typically, $\gamma = 1$ for keeping it simple). From Equation **(2.14)**, it is obvious that $p \to 0$ as $T \to 0$. The way to control the temperature variations also control the behavior of the algorithm, hence its efficiency.

There are numerous approaches to control the cooling rate or the temperature decrease which can be either through a linear cooling schedule or geometric cooling schedule. For a linear cooling timetable, the current temperature, *T*, is provided by Equation **(2.15)**.

$$T = T_0 - \beta t \qquad \textbf{(2.15)}$$

where T_0 is the beginning temperature and *t* is a pseudo time that replaces the iterations. β is the cooling rate and should be selected in such a method that $T \to T_f$ when $t \to t_f$ (or the maximum number N of iterations). This generally provides Equation **(2.16)**;

$$\beta = \left(T_0 - T_f\right) / t_f \qquad \textbf{(2.16)}$$

On the other hand, a geometric cooling time schedule decreases the temperature by a cooling factor $0 < \alpha < 1$ so that T is replaced by αT or

$$T(t) = T_0 \alpha^t, \qquad t = 1,2,...,t_f \qquad \textbf{(2.17)}$$

The advantage of the geometric cooling schema is there is no need to set the maximum number of iterations as $T \to 0$ when. Therefore, the geometric cooling schedule is used more often. To enamel cooling process to be slow enough to let the system to stabilize, the alpha constant is generally set such $\alpha \in [0.7, 0.99]$.

SA evaluates the objective function multiple times in a given temperature. Either performing many evaluations at a few temperature or doing a few evaluations at many temperature levels are suggested due to the balance between the number of evaluations and solution quality. Number of iterations can be set by two major ways: through the fixed number of iterations at each temperature or in a variable way such that number of iterations at lower temperatures is increased to fully explore local minima.

2.3.5.5. Genetic Algorithms

Genetic algorithms (GAs) can be regarded as the most well-known evolutionary algorithms based on a population with a wide range of applications from discrete systems (such as travelling salesman problem) to continuous systems (efficient design of airfoil in aerospace engineering).

The base of genetic algorithms consisting of encoding of solutions as arrays of bits or chromosomes and manipulation of these solutions selected based on their fitness and a set of genetic mechanisms. The procedure followed by genetic algorithms can be outlined through five steps (Yang, 2011) :

1. Encoding scheme;
2. Fitness function or selection criterion;
3. Generation of a population of chromosomes;
4. Evaluation of the fitness of each chromosomes in the population;
5. Generation of next population based on the fitness;

Steps 4 and 5 are repeated till stopping criterion (which is generally a certain number of generations) is satisfied.

One of the ways to define the fitness function is the assessment of an individual fitness relative to the entire population as shown in Equation **(2.18)**.

$$F\left(x_i\right) = \frac{f\left(\xi i\right)}{\sum_{i=1}^{N} f\left(\xi_i\right)} \qquad \textbf{(2.18)}$$

where ξi is the phenotypic value of individual *i*, and N is the population size. The basic mechanisms of the genetic algorithms are crossing over, elitism and mutation. Mutation is the change in some parts of chromosomes before they are transferred to the next population. Generally, mutation with a simple site is known to be inefficient. The selection of genetic algorithms' parameters such as population size and rates for crossing over, elitism and mutation are very important and generally requires considerable efforts for tuning. Figure **(2.6)** and Figure **(2.7)** illustrate the initial solutions used and final points obtained in a genetic algorithm application based on Yang (2011).

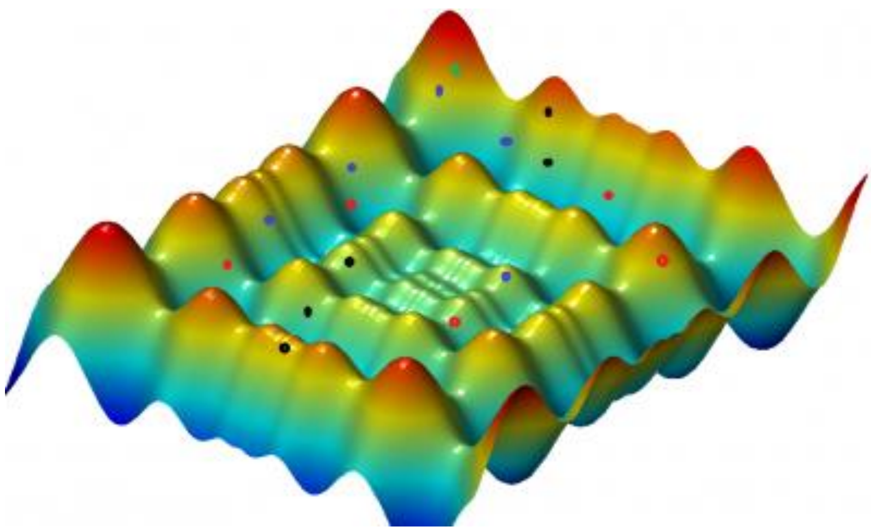


Figure 2.6. Genetic Algorithm (Initial Population and Locations) (Yang, 2011).

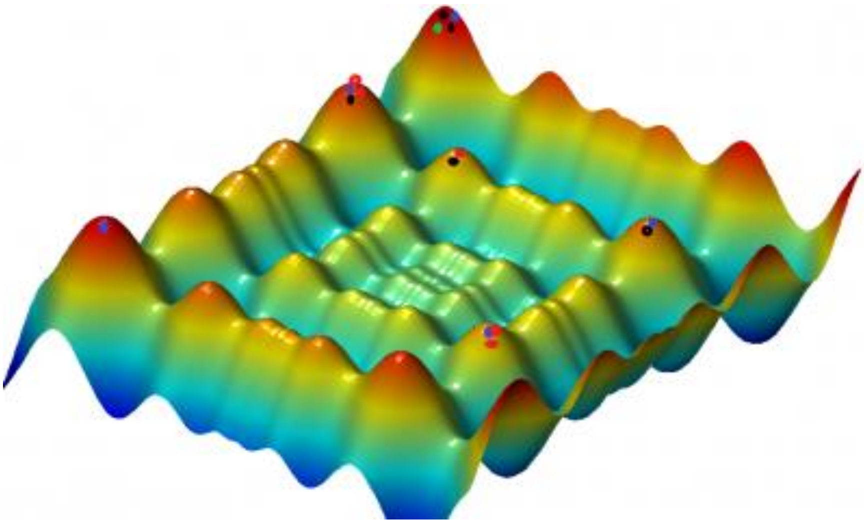


Figure 2.7. Genetic Algorithm (Final Locations) (Yang, 2011).

2.3.5.6. Differential Evolution

Differential evolution (DE) is a vector-based evolutionary algorithm that was developed in 1996 and 1997. It can be considered as a further development of genetic algorithm. It differs from genetic algorithms such that it carries out operations over each component (dimension of the solution) and almost everything is performed in terms of vectors.

For a *d*-dimensional optimization problem with *d* parameters, a population of *n* solution vectors x_i, $\left(i = 1, 2, ..., n\right)$ is generated first and for each x_i at any generation *t*, the vector x_i^t is used, which in turn represents a chromosome. (Storn and Price, 1997)

$$\mathrm{x}_i^t = (\mathrm{x}_{1,i}^t, \mathrm{x}_{2,i}^t, ..., \mathrm{x}_{d,i}^t) \qquad \textbf{(2.19)}$$

This vector can be considered as a chromosome.

DE consists of three main steps of three principle steps: mutation, crossover and selection. Mutation is performed through randomly selected three distinct vectors; x_p, x_q and x_r for each vector x_i at any time (generation) t. Then, the donor vector is generated through the mutation scheme presented in Equation **(2.20)**.

$$\mathrm{v}_i^{t+1} = \mathrm{x}_p^t + F(\mathrm{x}_q^t - \mathrm{x}_r^t) \quad \textbf{(2.20)}$$

where $F \in [0,2]$ is a differential weight parameter. Despite the fact that, as a principle, $F \in [0,2]$; in practice a scheme with $F \in [0,1]$ is more efficient and stable. In general the minimum population size $n \geq 4$.

The crossover, which is controlled by a probability $C_r \in [0,1]$, can be performed in two schemes: binomial scheme performs crossovers on each of the *d* components (variables, parameters). Replacement of a component using the donor vector is decided randomly by generating a uniformly distributed random $r_i \in [0,1]$, and the *j*th component of x_i is set according to Equation **(2.21)**.

$$x_{j,i}^{t+1} = v_{j,i}^{t+1} \quad \text{if} \quad r_i \leq C_r, \quad \left(j = 1, 2, ..., d \right) \quad \textbf{(2.21)}$$

Interested readers can find the details about DE, especially for different schemes used in DE, in the study of Price *et al.* (2005).

2.3.5.7. Ant Colony Optimization

Ant colony optimization was proposed by Dorigo in 1992 mimic the behaviors of ants. Ants use pheromone as a chemical messenger such that each ant lay pheromone to communicate with others, where each ant is able to follow the route marked with pheromone laid by other ants.

When a food source is found by an ant, the trail to and from it is marked with pheromone, whose concentration ϕ decays (evaporates) at a constant rate γ such that $\phi(t) = \phi_0 e^{-\gamma t}$ where ϕ_0 is the initial concentration at t = 0. Evaporation affects the possibility of convergence. Starting from following randomly route, the pheromone concentration varies and the route with higher pheromone concentration is followed by ants often, in turn, the pheromone is enhanced by increasing number of ants that follow the same route. This route eventually becomes a favored path.

2.3.5.8. Bee Algorithms

Bee algorithms, a few variant of which are known to be honeybee algorithm, artificial bee colony, bee algorithm, virtual bee algorithm and honeybee mating algorithms, are inspired by foraging behavior of bees. When bees find a good food source and bring some nectar back to the hive, they recruit more bees by using directional dancing with varying strength in order to communicate the direction and distance of the food source. In case of multiple food sources, such as flower patches, bee colony seems to be able to allocate forager bees among different flower patches to maximize their total nectar intake (Moritz and Southwick 1992).

The historical developments on such algorithms can be summarized: Honey Bee Algorithm (HBA) was first formulated by Tovey and Nakrani in 2004 to allocate computers among different clients and web-hosting servers. Yang (2005) developed Virtual Bee Algorithm (VBA) to solve optimization problems. Pham *et al.* (2005) developed the bee algorithms which was followed by the development of Honey-bee mating optimization (HBMO) in 2005 by Haddad and Afshar (Haddad *et al.*, 2006). At the same times, Artificial Bee Colony (ABC) algorithm was developed by Karaboga (2005).

Ant and bee algorithms are more suitable for discrete and combinatorial optimization, and have a wide range of applications (Yang, 2011).

2.3.5.9. Particle Swarm Optimization

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart (1995) based on observed swarm (such as fish and bird) behavior in nature. It has a wide range of applications in optimization (such as computational intelligence, and design/scheduling applications and their variants).

PSO searches solution space by adjusting the trajectories of particles (individual agents), each of which traces a piecewise path that can be modelled as a time-dependent positional vector. The movement of a swarming particle consists of a stochastic component and a deterministic component. That is, each particle is attracted toward the position of the current global best, g^*, and its own best known location, x_i^*, although they have a tendency to move randomly. There is a current best for all particles at any time *t* at each iteration. The goal of PSO is to find the best global among as the current best solutions until the objective improves no longer or after a certain number of iterations. Given x_i is the position vector and v_i is the velocity vector of particle *i*, which in practice takes a value between [0, v_{max}], the new velocity vector can be computed through the Equation **(2.22)**.

$$v_i^{t+1} = v_i^t + \alpha\epsilon_1 [g^* - x_i^t] + \beta\epsilon_2 [x_i^* - x_i^t] \qquad \textbf{(2.22)}$$

where $\in_1$ and $\in_2$ are two random vectors, whose each entry takes a value between 0 and 1. α and β are learning parameters (acceleration constant) that are generally set taken as "≈ 2".

In general, the initial velocity of a particle can be set to zero, i.e., $v_i^{t=0} = 0$. The new position can be computed through the Equation **(2.23)**.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \qquad \textbf{(2.23)}$$

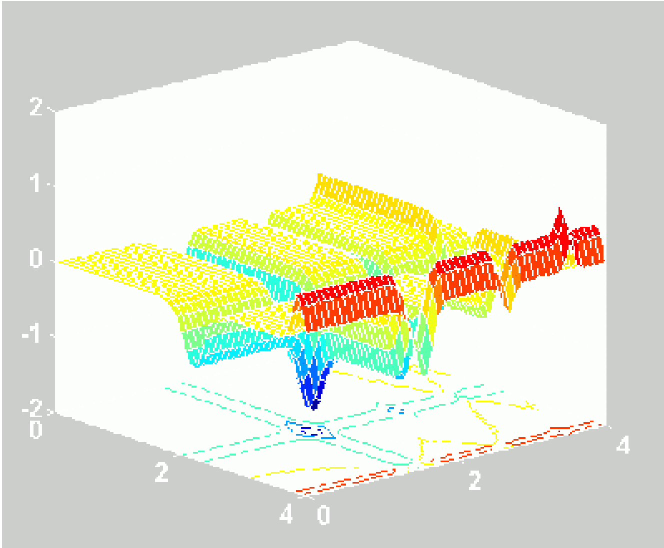


Figure 2.8. Display How All Particles Move Towards the Global Optimum (Yang, 2011).

2.3.5.10. Tabu Search

Tabu search was developed by Fred Glover in the 1970s, was published first in 1997 (Glower and Laguna, 1997). The major component of the method is the use of memory and search history. Therefore, it can be considered as an intensive local search, and appropriate use of search history avoids revisiting local solutions through the records of recently tried solutions in tabu lists, that is likely to save considerable computing time, hence improve the search efficiency.

2.3.5.11. Harmony Search

Harmony Search (HS) was developed by Geem *et al.* in 2001 based on the improvisation process of a musician that indicates three choices:

1. Musician can play any famous piece of music (a series of pitches);
2. Musician can play something similar to a known piece (adjusting the pitch slightly);
3. Compose new or random notes.

From the perspective of Markov chain, the pitch adjustment can be considered to be a random walk that generates a new solution from the current solution x_{old} according to Equation **(2.24)**:

$$x_{new}^{t+1} = x_{old}^{t} + b_p e_i^t \ , \qquad \textbf{(2.24)}$$

where e_i^t is a random number obtained from a uniform distribution [−1, 1] and b_p is the bandwidth, which controls the local range of pitch adjustments.

2.3.5.12. Firefly Algorithm

The Firefly Algorithm (FA) was produced by Yang (2009) and is based on the flashing patterns and behavior of fireflies:

1. Fireflies are unisex so that one firefly will be attracted to other regardless of their sex.

2. The attractiveness is proportional to the brightness. Thus, for flashing two fireflies, the brighter firefly attract the other one. If neither one is brighter, a random move is performed. The attractiveness decreases as the distance between two fireflies increases.

3. The brightness of a firefly is determined by the landscape of the objective function.

A firefly’s attractiveness is proportional to the light intensity seen by adjacent fireflies, thus, the variation of attractiveness, β, with the distance r can be presented by Equation **(2.25)**.

$$\beta = \beta_0 e^{-\gamma r 2} \qquad \textbf{(2.25)}$$

where β_0 is the attractiveness at r = 0.

The movement of firefly i, attracted to another more attractive (brighter) firefly j, is determined according to Equation **(2.26)**.

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r^2_{ij}} (x_j^t - x^t_i) + \alpha e^t_i \qquad \textbf{(2.26)}$$

where α represents randomization parameter and e_i^t is a vector of random numbers that is obtained generally from either a Gaussian distribution or uniform distribution at time t. In the special case of $\beta_0 = 0$ represents the random walk. Figure **(2.9)** demonstrates an illustrative.

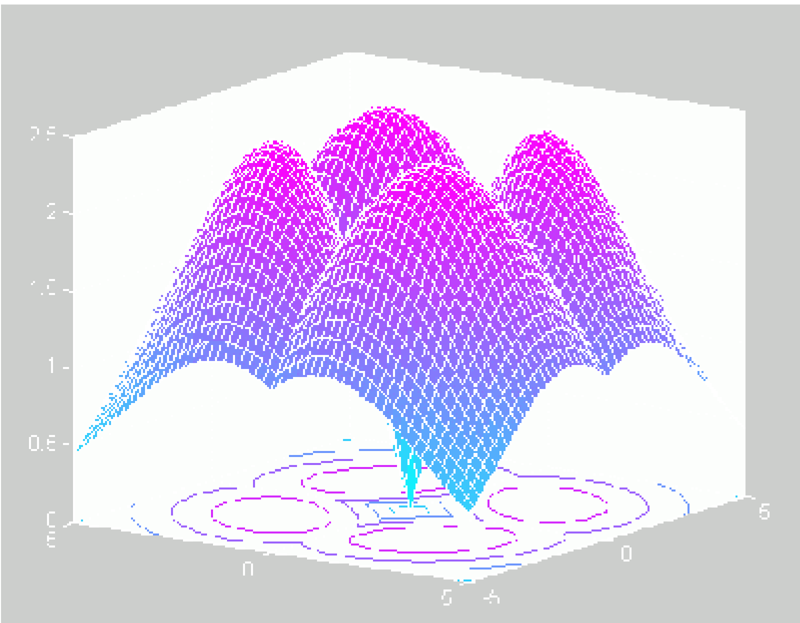


Figure 2.9. Firefly Algorithm and Fireflies Move Towards 4 Global Optima. (Yang, 2011)

2.3.5.13. Cuckoo Search

Cuckoo search (CS) was developed by Yang and Deb in 2009 based on the brood parasitism of some cuckoo species (Yang and Deb, 2010). One of the characteristics of cuckoos is having an aggressive reproduction strategy. For example, some species lay their eggs in communal nests. Even they can remove others' eggs to increase the probability of their own eggs to hatch. Some of species lay their eggs in the nests of other host birds (often other species). The reproduction mechanism can be outlined in three stage (Yang, 2011):

1. Each cuckoo lays one egg at a time in a randomly chosen nest.
2. The best nests with high-quality eggs are remained in next generations.
3. The number of available nest is fixed. There is a probability $p_a \in [0,1]$ for the egg laid by a cuckoo to be discovered by the host bird. In this case, the host bird can either eliminate the egg or simply abandon the nest and build a new nest.

A Lévy is performed according to Equation **(2.27)** to generate a cuckoo *i*, i.e. new solutions, x^t.

$$x_i^{(t+1)} = x_i^{(t)} + \alpha L(s, \lambda) \quad \textbf{(2.27)}$$

where α > 0 is the step size to be scaled according to the problem of interest. Note that Equation **(2.27)** represents a stochastic equation for a random walk, which is a Markov chain whose next status/location only depends on the current location, $x_i^{(t)}$, with a transition probability, $\alpha L(s, \lambda)$. The step length, *s*, of the random walk is obtained from a Lévy distribution; Equation **(2.28)**:

$$L(s, \lambda) \sim s^{-\lambda}, (1 < \lambda \le 3) \quad \textbf{(2.28)}$$

Yang (2011) indicates that a Lévy flight is more efficient in the long run. He also claims that CS is potentially far more efficient compared to PSA and GA (Yang and Deb, 2010).

2.3.5.14. Water Cycle Algorithm

One of the most recent metaheuristics inspired from the nature is water cycle algorithm (WCA) developed by Eskandar *et al.*, (2012) based on water cycle process for constrained optimization. WCA. WCA begins with an initial population (consisting of raindrops). The best individual (raindrop/solution) is chosen as sea. Then a number of good raindrops are chosen as rivers and the rest of the raindrops are regarded as streams flowing to rivers and sea. Eskandar *et al.* (2012) indicate that WCA are competitive to other metaheuristics.

2.3.5.15. Other Metaheuristics Algorithms

Apart from algorithms mentioned above, there are many other metaheuristics in the literature, for instance, artificial immune systems based on the characteristics of the immune system of mammals, which was first proposed by Farmer *et al.* 1986 followed by work of Bersini and Varela (1990) on immune networks. In 1989 Moscato proposes the memetic algorithm, which has a characteristics of multi-generation, co-evolution and self-generation. Cross-entropy method, which can be regarded as generalized Monte Carlo method, developed by Rubinstein in 1997. Cross-entropy algorithm is consisting of generation of random samples and parameter updates to minimize cross entropy. Another algorithm that is inspired by nature is bacterial foraging optimization developed by Passino around 2002.

Eventually, there are much more metaheuristics available in the literature, which cannot be possibly to mentioned all of them in this study. However, some of them are introduced with their key characteristics to enable readers to have enough knowledge about them so that they can conceptualize the contribution of this study better.

2.3.5.16. A General Design Example for Metaheuristics

For the purpose of the illustration of design problems for which metaheuristics are used to optimize, this part presents a model used by Yang (2011). He considers the design of a compressional and tensional spring that involves three decision variables; wire diameter x_1, coil diameter x_2, and the length of the coil x_3. According to engineering specifications, the design model presented by Yang (2011) can be seen through Exhibit **(2.2)**.

minimize $f(x) = x_1^2 x_2 (2 + x_3)$

subject to the following constraints

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^3 x_3} \leq 0$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

The limits of the variables are;

$$d_1 : 0.05 \leq x_1 \leq 2.0, \qquad d_2 : 0.25 \leq x_2 \leq 1.3, \qquad d_3 : 2.0 \leq x_3 \leq 15.0$$

Exhibit 2.2. General Design Optimization Model, (*P2*), (Yang, 2011).

Depending on the engineering specification, (***P2***) seeks the minimization of the objective function $f(x)$ subject to four constraints, $g_i(x) \quad \mathrm{i} = 1..4$, depending on the relationships among the decision variables, $x_i \quad i = 1..3$, by identifying the best values for decision variables (i.e. design parameters of wire diameter, coil diameter, and the length of the coil) that have specified domains of $d_i \quad i = 1..3$. Yang (2011) uses simulated annealing starting from an initial solution of $x_0 = (1.0, 1.0, 14.0)$. He also uses a population based algorithm, specifically PSO, starting from an initial population that is consisting of n solutions vector. The best solution found is represented by Equation **(2.29)**.

The accompanying best arrangement can be discovered effectively (Yang, 2011).

$$x_* = (0.051690, 0.356750, 11.28126), \qquad f(x_*) = 0.012665 \qquad \textbf{(2.29)}$$

This small engineering design application is presented in this section to let readers understand the optimization model presented in Section (3.1) and for which a hybrid heuristics as well as traditional metaheuristics are to be investigated in Section (3.2) better. Interested reader can refer the studies of Glover and Kochenberger 2003, Talbi 2009, Yang 2010 for further metaheuristics applications on engineering design optimization.

2.3.6. Multi-Objective Analysis Techniques

A multiple objective optimization problem (MOP) deals with the minimization or maximization of objective functions (more than one) subject to a set of constraints. This section provides a detailed information about some of the multi-objective optimization techniques such as goal programming (GP), weighted objective function, shannon entropy method, multi-moora method, fuzzy logic, multi-attribute utility function, pareto optimality, analytical hierarchy process (AHP), physical programming (PP) and utopia distance that are used often in the literature. Therefore, once can better conceptualize the multi-objective modeling approach presented in Section (3.4).

In general, a solution to MOP indicates a Pareto optimal solution due to tradeoffs between its competing objectives it has. The image of this solution in the space of objectives is termed as Pareto front. In fact, the resolution of a MOP can be regarded as an approximation of the Pareto front.

As far as CLNs and the reliability optimization of them considered, there are several approaches that are listed in Table **(2.3)** and often used in the applications such as multi-action selective maintenance, redundancy allocation and sustainability modeling. In supply chain networks, there is a variety of objectives such as increasing service level, decreasing warehouse costs, fixed costs, variable costs and total costs, decreasing lead time, consolidating supplier's bases, improving base's reliability and increasing the total quality of supply while setting the values for decision variable in the stage of decision making.

In the literature, Liao and Rittscher (2007) consider optimization of total cost, the level of rejection rate, late delivery, flexibility rate in terms of supplier selection under the constraints of demand satisfaction and capacity utilization. Shannon and Weaver (1947) develops a multi-objective transshipment planning model for the petroleum industry where the objectives are minimizing total transshipment cost, maximizing the production, and conforming to demands. Coello (1999) investigates a multi-location inventory problems that have multiple objectives such as minimization of cost and maximization of fill rate.

In terms of CLN design optimization in view of mission success, Miman and Pohl (2012) provides a multi-objective sustainability optimization model that tries the identify the best set of maintenance activities to perform on the links of the network with multiple objectives; maximizing the reliability of the CLN (mission success), minimizing the total cost of selective maintenance activities performed and minimizing the total time required by maintenance activities.

In general, there it is likely that managers may have numerous goals to achieve, depending on the situation of managers. Managers want to satisfy the needs of the situation as desired by them.

Table 2.3. Multi-Objective Optimization Technics (Based on Miman, 2008).

Method	Final Solution	Suitability	Applicability
Goal Programming	unique	directly	rely on subjective weights
			final solution may not be the best compromise
Weighted Objective Function	unique	directly	rely on subjective weights
			final solution may not be the best compromise
Multi-attribute Utility Functions	unique	directly	utility function is difficult to determine
Pareto Optimality	set of non-dominated solutions	indirectly	optimization algorithms can not be directly applied
			set of solutions yielded for further considerations
AHP	unique	not-suitable	not appropriate for very large solution space
Utopia Distance	unique	directly	rely on relative scales for each measure, which act as weights
			final solution may not be the best compromise
Physical Programming	unique	directly	based on decision maker's preferences
			acceptance parameter may need some small tunings
			final solution reflects the decision maker preference

The details about each of the methods presented in Table **(2.3)** are provided in successive sub-sections.

2.3.6.1. Goal Programming

Goal Programming is a pragmatic programming method that is able to choose from an infinite number of alternatives. In general it can be regarded as an extension or generalization of linear programming to handle multiple, normally conflicting objectives. Each of these objective (criteria) is given a goal (target) value to be achieved. It minimizes the deviations from this set of target values. In goal programming, each of the criteria is considered with a target value where deviations from the targets are minimized through the weighted penalties. In general Goal programming is used to perform three types of analysis (URL-1):

1. *Determine the required resources to achieve a desired set of objectives.*
2. *Determine the degree of attainment of the goals with the available resources.*
3. *Providing the best satisfying solution under a varying amount of resources and priorities of the goals.*

One of its advantages is its capacity to handle large-scale problems and to produce infinite alternatives depending on the situation. A major disadvantage is its ability to weigh coefficients for which many applications uses other methods such as analytical hierarchy process (AHP). GP has a wide range of applications from production planning and scheduling to health care and portfolio selection. DeCroix and Arreola-Risa,(1998) and Gigerenzer et al. (2005) indicate that multi-criteria decision making (MCDM) methods such as AHP are most often utilized in applications that avoid most of GPs disadvantages.

The basis of goal programing (GP) originated in 1955 by Charnes *et al*. (1955) despite the fact that the name “goal programming” was used first time by Charnes and Cooper (1961). Tan (2001) provides the application of goal programming to diverse issues. The main disadvantages of goal programming, as indicated in Table **(2.3)**, is the ambiguity in the determination of coefficients (weights) (Vollmann and Cordon, 1998).

The most meaningful approach towards the mathematical formalization of fuzziness was pioneered by Zadeh (1965). Preliminary works on fuzzy decision making can be found in Kickert (1978) and Zimmermann (1987). Zimmermann (1991) provides applications of fuzzy theory in management, business and operational research. Hannan (1981) develops the fuzzy goal programming (FGP), which is later used in multi-objective optimization problems.

2.3.6.2. Weighted Objective Function

In general all the multi-objective optimization techniques seek non-dominated solutions represented in Pareto front and illustrated in Figure **(2.10)** considering the tradeoffs among the objectives.

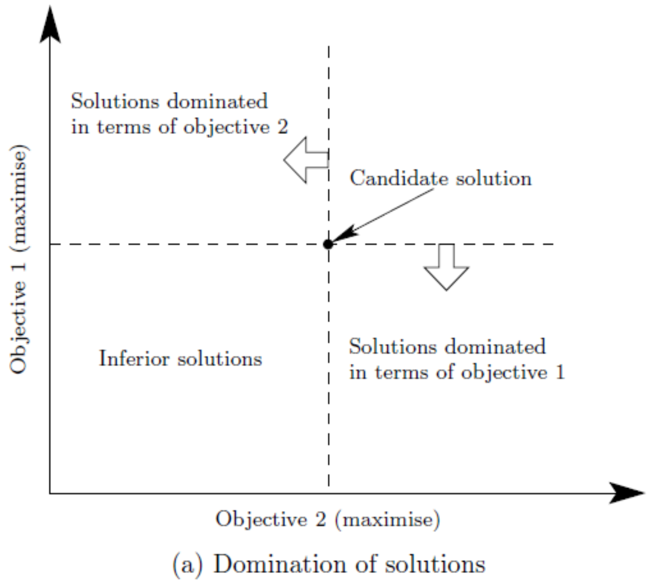


Figure 2.10. Illustration of Domination (Das, 2005).

If a solution is dominated with respect to all the objectives when compared to another solution (having more desirable performance for all of the objectives), the former is dominated by the latter and considered to be inferior. Weighted objective function techniques combine all objectives together in one function that is in the form of a linear combination of the objective function values. Different weights are assigned to the coefficients in the combination with the intention of approximating the Pareto optimal set of multiple objective solutions closely.

As it is described in Table **(2.3)**, this technique also has disadvantages of tedious process of setting weights. Das (2005) describes complexities of the weighted sum of objectives and claims that the method is efficient in getting points from all parts of the Pareto set only when the Pareto curve is convex.

Recently, Dağ and Miman (2015) provide a multi-objective optimization model for the reliability of a CLN through a weighted objectives function of the reliability of a network with distorted risks of failure of bases; cost of stock allocation and total stocks to allocate. The goal of them is maximizing the reliability of the network while minimizing the cost and total number of stocks for allocation. They illustrate their model with an example and its solution through Excel Solver, which indicates a Pareto optimality.

2.3.6.2.1. Shannon Entropy Method

Shannon and Weaver (1947) propose the use of the idea of entropy, also emphasized by Zeleny (1982), in order to determine the weight (relative significance) of criteria. Entropy, which is a fundamental concept in physical sciences, social sciences and systems, represents the amount of uncertainty raised from the content of a message (Farzamnia and Babolghani, 2014). The goal of this method is to provide more information to decision makers. Given *m* alternatives and *n* criteria, the data can be represented in the form of decision matrix as in Equation **(2.30)**.

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_n \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \end{bmatrix} \end{array} \quad \textbf{(2.30)}$$

The steps of determining the weight of every indicators are presented in Equation **(2.31)-(2.35)** according to Shannon and Weaver (1947):

$$P_{ij} = \frac{x_{ij}}{\sum_{t=1}^{m} x_{ij}} \tag{2.31}$$

The level of the *j*th index is obtained according to Equation **(2.32)**:

$$E_j = -K\sum_{i=1}^{m}\left[P_{ij} \,.LnP_{ij}\right] \forall j, \quad K = \frac{1}{Ln\,(m)} \tag{2.32}$$

Level of uncertainty or degrees of knowledge generated for the *j*-th index (d_j) is computed through Equation **(2.33)**.

$$d_j = 1 - E_j; \quad \forall_j \tag{2.33}$$

Finally, the weight (w_j) of the indicators are computed through Equation **(2.34)-(2.35)**.

$$w_j = -\frac{d_j}{\sum_{j=1}^{n} d_j} \; ; \quad \forall j \tag{2.34}$$

$$w'_j = \frac{\lambda_j w_j}{\sum_{i=1}^{n} \lambda_i w_i} \tag{2.35}$$

2.3.6.2.2. MultiMoora Method

One of the power techniques that can be used in determining weights for criteria known as multi-objective optimization technique using ratio analysis (MOORA) developed by Brauers (2004). Later, MULTIMOORA, wider version of MOORA, is proposed by Brauers and Zavadskas (2010). Considering the same decision matrix *D* where there are *m* alternatives $\{A_i \mid i = 1, 2, ...m\}$ and *n* criteria $\{C_j \mid j = 1, 2, ..., n\}$ and knowing the relative weight of each criterion (w_j), one can apply the MULTIMOORA technique in four steps (Balezentis and Shouzhen, 2013):

<u>*Step One:*</u> The data contained in the choice matrix (*D*) is scaled according to Equation **(2.36)**:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \qquad \textbf{(2.36)}$$

<u>*Step Two:*</u> Positive indicators (such as profits) and negative ones (such as material costs) are separated according to Equation **(2.37)**:

$$A^* = \left\{ A_i \mid Max\left(\sum_{j=1}^{g} r_{ij}\ w_i - \sum_{j=g+1}^{n} r_{ij}\ w_j \right) \right\} \qquad \textbf{(2.37)}$$

where $j = 1, 2... g$ positive standards and $j = g+1, g+2, ..., n$ are negative standards.

<u>*Step Three:*</u> Options based on the "reference point approach" upward (from low to high) are evaluated:

$$A^* = \{A_i \mid Min\left(Max \mid v_j^* - r_{ij}\ w_j \mid \right) \qquad \textbf{(2.38)}$$

where

$$v_j^* = \max_i r_{ij} \qquad \textbf{(2.39)}$$

Step Four: "perfect product" rankings are:

$$A^* = \left\{ A_i \mid Max \left(\frac{\prod_{j=1}^{g} r_{ij} w_j}{\prod_{j=g+1}^{n} r_{ij} w_j} \right) \right\} \quad \textbf{(2.40)}$$

where $j = 1, 2 \ldots g$ indices are positive and $j = g + 1, g + 2, \ldots, n$ indices are negative.

2.3.6.3. Multi-Attribute Utility Function

Most of the times decisions makers have an overview of the utility of the option or alternative available on the random bases. However, in multi-attribute utility optimization the decision makers are assumed to have perfect knowledge about the utilities that represent the boundaries in which the solution lies (Analytis *et al*., 2014). In their multi-attribute utility optimization model, the decision makers estimate the utility of the available alternatives and then prioritize them in the order of their desirability. There are three models of estimating utility: the linear multi-attribute model, equal weighting of attributes and a single attribute heuristic model. The performance of these models in 12 real world experiments which ranged from consumer choice to industrial experiments is evaluated.

Weitzman (1979) propose a general model for the selection of alternatives where the decision makers have low information about the alternatives in consideration initially but learn the exact utility contribution after paying a search cost. A recent study conducted by Dzyabura (2014) explicitly deals with how the step by step search can be guided by a multi-attribute utility model.

According to Analytis *et al.*, (2014), the decision makers' preferences can be described by a linear utility model which play a significant role in cognitive search engines in generating the orders in which the alternatives are ranked. Luan *et al.*, (2014) emphasize the need to integrate the decision making theories in psychology.

2.3.6.4. Pareto Optimality

In general, there are several response variables or different objectives that will define the quality attributes of the final product which may be conflicting, non-conflicting or partially conflicting in the nature of industrial processes. The conflicting objectives can be referred to those situations in which a particular objective can only be improved by comprising other objectives. The similar case is valid for the partially conflicting objectives. However, the partially conflicting objectives can be termed in those circumstances where the variables are conflicting on some common grounds. Carmon *et al*. (1994) indicate that there are various definitions of the terms conflicting with multi-objective optimization and decision making in the literature. According to Deb (2001), a set of objectives regarding as conflicting if no solution is able to achieve the optimal value for each defined criteria and vice versa. Tan (2001) defines a conflict as the existence of incomparable solutions in the search space. In the context of a multi-objective optimization accompanied by a conflicting or a partially conflicting objectives, solution approaches yield not only one but different multiple solutions which are termed as non-dominated or Pareto optimal solutions (Vance *et al.*, 1994).

As previously mentioned, in multi-objective decision making environments, objectives are often likely to be conflicting in various areas, consequently there are tradeoffs among the performances obtained for multiple objectives for a given solution. Therefore only non-dominated solutions should be desired to be obtained while inferior solutions are ignored. The collection of non-dominated solutions is called as the Pareto optimal set and represented as a Pareto front as described in Figure **(2.11)**. The goal of multiple objective approach is to find points in the Pareto front. As the number of objectives increases, the number of non-dominated solutions may increase a lot. This is due to the fact that, in case of more objectives, the likelihood of domination with respect to all the objectives decreases. In such cases, the multi-objective methods may struggle to converge and may be out performed by the single objective solutions as described in detail by recent studies such as Hughes (2003), Purshouse (2003) and so on.

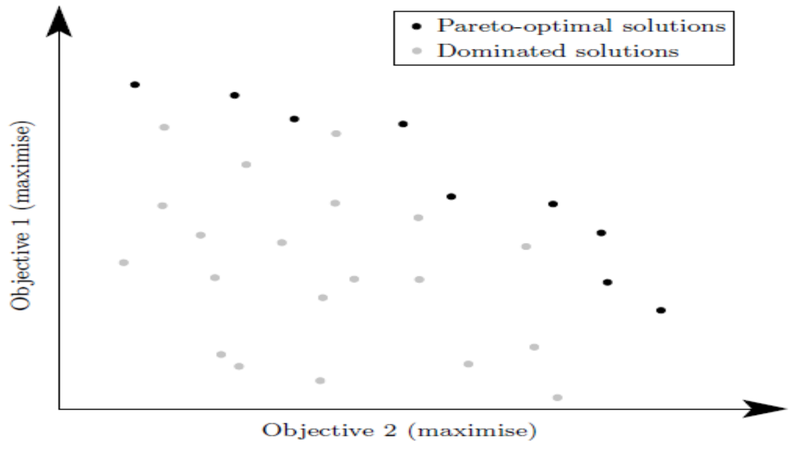


Figure 2.11. Pareto-Optimal Solutions (Belton and Stewart, 2002).

The most important task in multi-objective optimization is to come up with a subset of Pareto optimal solutions which represents the search space. Once the Pareto optimal solutions are found, the further step is the multi-criteria decision making analysis, such as AHP, which involves further consideration to arrive at an optimal solution.

2.3.6.5. Analytical Hierarchy Process (AHP)

The analytic hierarchy process (AHP) proposed by Saaty (1990) is a strong and executable multi-criteria decision making approach to deal with complex problems in which both qualitative and quantitative objectives are taken into account. The AHP helps analysts to structure crucial parts of a problem into a hierarchy similar to that of a family tree. This reduces complex decisions to a series of simple comparable ones which are evaluated at each level of hierarchy, and then synthesized to reach the final results (Bevilacqua et al., 2006).

An AHP approach starts by defining the decision criteria in the form of a hierarchy of objectives. The hierarchy is structured on different levels starting from the top level i.e. the overall objective through the intermediate levels involving the criteria and sub criteria on which the subsequent levels depend to the lowest level i.e. the alternatives. After setting the hierarchy structure, a set of square pairwise comparison matrices for each of the lower levels with seperate matrix for each element in the above level is constructed (Parlar and Berkin, 1991).

AHP uses the simple pairwise comparisons to determine the weights and the ratings enabling the analyst can concentrate on only two attributes at a time. One of the strengths of the AHP approach is that it allows the decision makers to specify their own preferences using the verbal scale as indicated in Table **(2.4)**. These verbal judgments are then converted into a score using the discrete 9-point scale. This hierarchy synthesis function is used to weight eigenvectors by the weights of the criteria and the sum is taken over all weighted eigenvector entries conforming to those in the next lower level of the hierarchy.

Table 2.4. The Fundamental Scale for Pairwise Comparisons used in the AHP Approach (Saaty, 1980).

Intensity of importance	Definition
1	Equal importance
3	Moderate importance
5	Strong importance
7	Very strong importance
9	Extreme importance
2, 4, 6, 8	For compromises between the above

One of the disadvantages of the AHP is the large number of AHP pairwise comparisons which are required to produce the optimal solution for high dimensional decision making problems (Carmon et al. 1994). In addition, even under the best circumstances, the respondents of the survey is very likely to suffer from information overload in a very long discussion or interview resulted from a large number of pairwise comparisons, which makes the decision makers confused (Miller, 1956). However, when there are only few levels and sublevels, the AHP becomes a powerful tool that can be applied straightforward to obtain the weights or relative preferences of the alternatives.

An AHP hierarchy is a structured means of describing the problem at hand, which consists of an overall goal, a group of options or alternatives for reaching the goal, and a group of factors or criteria that relate the alternatives to the goal. In most cases the criteria are further broken down into sub criteria, sub-sub criteria, and so on, in as many levels as the problem requires. The hierarchy can be visualized as a diagram like the one in Figure **(2.12)** where the goal is at the top, the alternatives at the bottom, and the criteria filling up the middle. Specifically, Figure **(2.12)** represent a hierarchical structure consisting of three levels: one goal (at the top), 5 criteria (in the middle) and three alternatives (on the bottom).

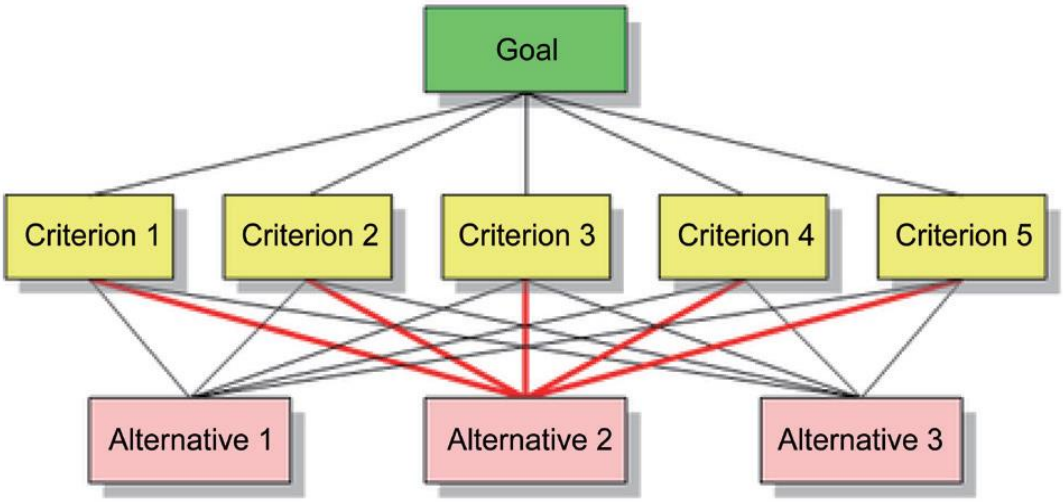


Figure 2.12. A Hierarchy Structure in AHP (Saaty, 1990).

Saaty (1990 and 1994) describe steps for applying AHP:

i. Define the decision making problem and its overall goal,

ii. Structure the hierarchy with the decision maker’s goal at the top with the criteria at the intermediate levels, and alternatives on the bottom,

iii. Construct the set of pairwise comparison matrices for each to the lower levels with one matrix for each element in the level immediately above. The pairwise comparisons that capture the preference of the decision maker are performed using the relative measurement scale presented in Table **(2.4)**.

iv. For a comparison matrix of size *n*, there are *n* (*n*-1)/2 judgments required to develop the matrix. Reciprocals are automatically assigned in each pair wise comparison.

v. The hierarchy synthesis function is used to weight the eigenvectors by the weights of the criteria and the sum is taken over all weighted eigenvector entries corresponding to those in the next lower level of the hierarchy.

vi. After all the pairwise comparisons are completed, the consistency of the comparisons is assessed by using the eigenvalue, λ, through the consistency index, C.I .computed according to Equation **(2.41)** (Saaty,1980).

$$C.I. = (\lambda - n) / (n-1) \tag{2.41}$$

The consistency of pairwise judgements can be checked through the consistency ratio (C.R) computed by Equation **(2.42)** proportion ($C.R.$) (Saaty, 1980).

$$\text{C.R.} = \frac{\text{C.I.}}{\text{R.I.}} \tag{2.42}$$

where R.I. stands for Random Consistency Index, for which the appropriate values are provided in is given in Table **(2.5)** depending on the size of the comparison matrix. Saaty (1980) suggests that the C.R. is acceptable if it does not exceed 0.10. Otherwise, the judgment matrix is considered to be inconsistent, hence, the judgments should be reviewed and repeated to obtain a consistent matrix.

Table 2.5. Random Consistency Index (Triantaphyllou & Mann, 1995).

n	1	2	3	4	5	6	7	8	9	10	11
R.I.	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51

2.3.6.6. Physical Programming

Physical programming (PP) developed by Messac (1996) is a multi-objective optimization technique that enables decision makers to express their preference on different objectives through the use of class functions. Class functions are functions of the corresponding objective values and are classified into four classes: smaller is better (i.e., minimization), larger is better (i.e., maximization), value is better, and range is better.

Messac (1996) claims that his PP approach is an effective and computationally-efficient approach for design optimization. He emphasizes the deficiencies of the traditional approaches such that weight-based (Archimedian) approaches as very difficult to implement in practice for realistic problems that comprise many weights as these methods generally require many iterations on the choice of weights; and there is no clear guidance provided by them to converge to the right set of weights. Besides, preemptive approaches implicitly rank the objectives such that priority-1 objective is infinitely more important than a priority-2 objective, which is infinitely more important than a priority-3 objective and so on. Messac (1996) indicates that such preemptive approaches are unrealistic with their prioritizing under the implicit assumption that one objective is infinitely more important than another. PP applies "One vs. Others Criteria Rule (OVO) rule" which can be stated as a full reduction for one criterion (objective) across a given region (k = 3, 4, 5) is preferable to the full reduction for all the other criteria across the next better region, (k-1).

Some of its applications include an optimization-based production planning model (Messac *et al.*, 2002) and optimal redundancy allocations for multi-state series parallel systems (Tian *et al.*, 2008). Recently, Miman and Pohl (2012) considers provides a multi-objective optimization model for the sustainability of a CLN where objectives are maximization of the sustainability and minimization of cost of maintenance activities and time required by maintenance activities. Their model constitute the first time use of PP, applied to the area of contingency logistic design and selective maintenance optimization. They argue that the concept of PP provides greater flexibility and reflects the decision makers preferences explicitly hence facilitates the design of a CLN.

2.3.6.7. Utopia Distance

Kim and De Weck (2012) provide detailed information about utopia distance approaches. The fundamental philosophy of the utopia distance is to adaptively refine the Pareto front. It starts from the determination of a rough profile of the Pareto front, continues with the estimation of the size of each Pareto patch (line segment in the case of two-dimensional problems). Later and the regions for further refinement, which are specified as feasible domains for sub-optimization by assigning additional constraints, in the objective space are determined.

In the bi-objective adaptive weighted sum method, the feasible domain for further exploration is determined by specifying two inequality constraints. The usual weighted sum method is then performed as sub-optimization procedure in these feasible domains obtaining more Pareto optimal solutions. The procedure is repeated for further refinement until a termination criterion is met. Figure **(2.13)** compares the typical weighted sum method and the bi-objective adaptive weighted sum method for a sample problem that has a relatively flat and non-convex region.

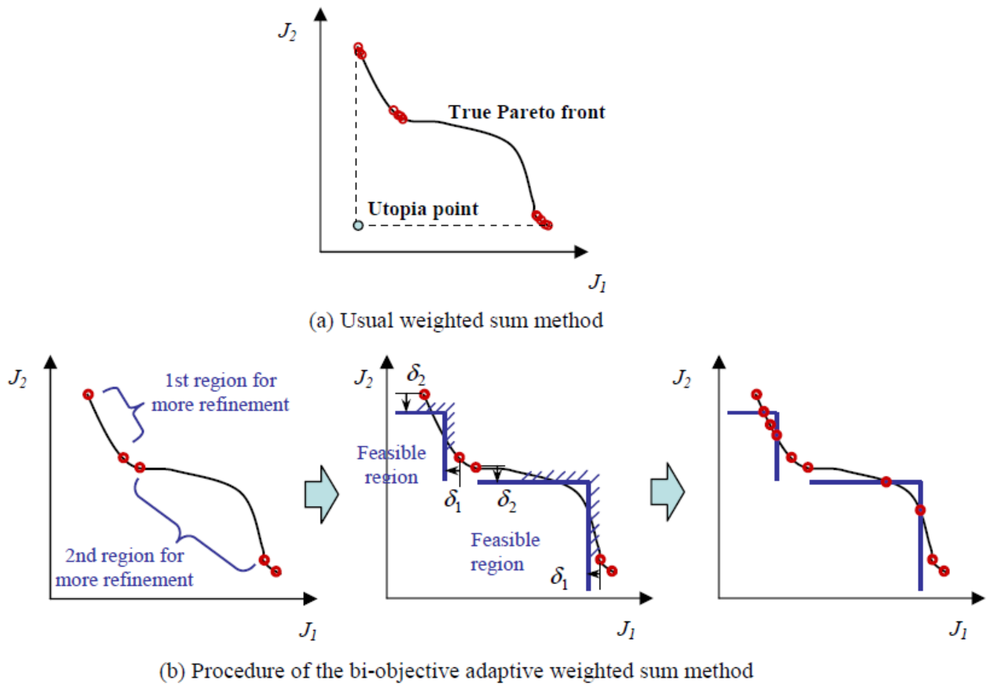


Figure 2.13. The Concept and Procedure of the Adaptive Weighted Sum Method (Kim and De Weck, 2012).

In the first stage of Utopia Distance for multi-dimensional problems, the approximate shape of the Pareto front is determined by using the usual weighted sum method, then Pareto front patches are identified, and patches for further refinement are selected on the basis of the patch size. Figure **(2.14)** illustrates the concept of the multi-objective adaptive weighted sum method with equality constraints for multi-objective optimization. In the three-dimensional case, the Pareto front becomes a surface, and the linearized Pareto front patch is represented by four line segments that connect four vertices, as shown in Figure **(2.15)**.

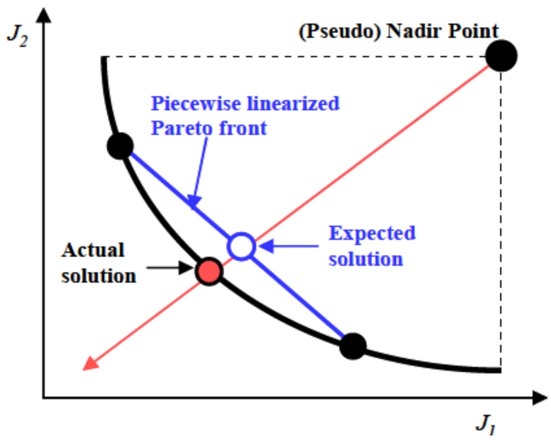


Figure 2.14. Adaptive Weighted Sum Method (Utopia Distance) for Multi-Dimensional Problems (2-D Representation) (Kim and De Weck, 2012).

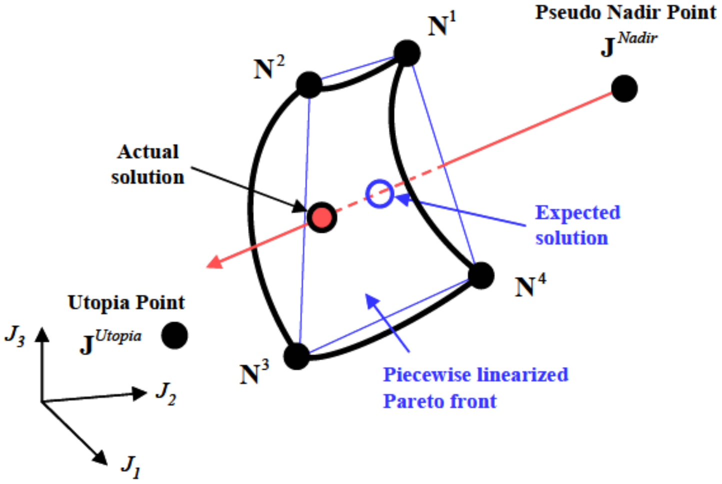


Figure 2.15. Utopia Distance for 3-D Problems (Kim and De Weck, 2012).

The complete and detailed procedure is presented in the following sub-sections.

2.3.6.7.1. Notation List for Utopia Distance

This section provides the list of notations that are used to describe the procedures and mathematical formulation for a multi-objective formulation based on utopia distance according to Kim and De Weck (2012).

J = objective function vector
x = design vector
p = vector of fixed parameters
g = inequality constraint vector
h = equality constraint vector
m = number of objectives
α_i = *i*th weighting factor
$\overline{J}_i$ = normalized objective function
J^{Utopia} = utopia point
J^{Nadir} = nadir point
J^{i*} = ith anchor point J^{i*}
p^j = position vector of the *j*th expected solution on the piecewise linearized plane

2.3.6.8.2. Procedures of Utopia Distance

Procedures for executing the multi-objective utopia distance method are provided, based on Kim and De Weck (2012).

[Step 1] Stage = 1. Normalization of the objective functions. When $\mathbf{x}^{i*}$ is the optimal solution vector for the single objective optimization of the *i*th objective function J_i, the utopia point J^{Utopia} is defined by Equation **(2.43)**:

$$J^{Utopia} = [J_1(x^{1*})\ J_2(x^{2*}) \ldots J_m(x^{m*})] \quad \textbf{(2.43)}$$

And the pseudo nadir point J^{Nadir} is given by Equation **(2.44)**.

$$J^{Nadir} = [J_1^{Nadir}\ J_2^{Nadir} \ldots J_m^{Nadir}] \quad \textbf{(2.44)}$$

where m is the number of objective functions or the dimension of the objective space, and each component J_i^{Nadir} is determined by Equation **(2.45)**:

$$J_i^{Nadir} = \max[J_i\,(x^{1*})\; J_i(x^{2*}) \ldots J_i(x^{m*})] \tag{2.45}$$

The ith anchor point J^{i*} is defined in Equation **(2.46)**:

$$J^{i*} = [J_1(x^{i*})\; J_2(x^{i*}) \ldots J_m(x^{i*})] \tag{2.46}$$

Normalized objective function $\bar{J}_i$ is calculated through Equation **(2.47)**:

$$\bar{J}_i = \frac{J_i - J_i^{Utopia}}{J_i^{Nadir} - J_i^{Utopia}} \tag{2.47}$$

[Step 2] Perform multi-objective optimization using the standard weighted sum approach with a small number of divisions, $n_{initial}$, for three objective functions, the weighted single objective function J_{Total} is obtained via Equation **(2.48)**:

$$\begin{aligned} J_{Total} &= \alpha_2\,[\alpha_1\,J_1 + (1-\alpha_1)\,J_2] + (1-\alpha_2)\,J_3 \\ &= \alpha_1\alpha_2\,J_1 + (1-\alpha_1)\,\alpha_2\,J_2 + (1-\alpha_2)\,J_3, \qquad \alpha_i \in [0,1] \end{aligned} \tag{2.48}$$

where α_i is the ith weighting factor. As a general form, the weighted single objective function of m objective functions, J_{Total}^m , is computed according to Equation **(2.49)**:

$$J_{Total}^{m} = \alpha_{m-1}\, J_{Total}^{m-1} + (1-\alpha_{m-1})\, J_m, \qquad m \geq 2 \qquad \textbf{(2.49)}$$

where $J_{Total}^{1} \equiv J_1$. Be aware of that (*m*-1) weighting factors are necessary to explore an *m*-dimensional objective space.

The uniform step size of the *i*th weighting factor α_i is determined by the number of initial divisions along the *i*th objective dimension as given by Equation **(2.50)**:

$$\Delta\alpha_i = \frac{1}{n_{initial,i}}, \qquad i = 1,\ldots,m-1 \qquad \textbf{(2.50)}$$

One can use the same step size for all weighting factors. In fact, there is a scheme that systemically determines each weighting factor and helps produce well-distributed solutions. In the adaptive weighted sum method, however, the usual step size strategy can be used because this approximate multi-objective optimization is conducted only once, after which adaptive refinement is conducted (Kim and De Weck, 2012).

[Step 3] Delete nearly overlapping solutions, which are obtained often when the weighted sum method is used. The Euclidian distances between these solutions are nearly zero, and among these, only one solution is needed to represent the Pareto front. In the computer implementation, if the distance among solutions in the objective space is less than a predetermined distance (ε), then all solutions except one are deleted.

[Step 4] Identify Pareto-front patches. Patches of any shape can be used, for example Kim and De Weck (2012) use quadrilateral patches in three-dimensional problems. In their case, four Pareto-optimal solutions become the four nodes of each patch, and edges are line segments that connect two neighboring nodes of each patch. They argue that constructing and maintaining meshes on the Pareto front may be tedious, but there are two advantages of using a mesh, for which interested reader can refer the work of

Kim and De Weck (2012). In general, patches play the role of primitives for further refinement for subsequent stages, as will be seen in Step 5, and if only non-dominated solution points are displayed, it is difficult to visualize and interpret the shape of the Pareto front. A mesh representation makes it very easy to visualize the Pareto surface as in the case of finite element meshes.

[Step 5] Stage = Stage + 1. Determine the layout for further refinements in each of the Pareto-front patches.

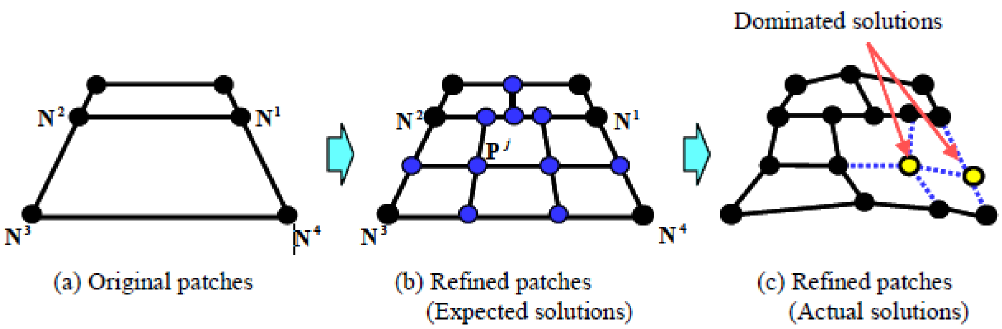


Figure 2.16. Adaptive Refinement Procedure (Kim and De Weck, 2012).

The larger the patch is, the more it needs to be refined. Figure **(2.16)** shows an example of refinement, in which a patch is composed of four nodes in three dimensional objective space. Kim and De Weck (2012) indicate that, because the lower patch is larger, it is refined more than the upper one. In each mesh, the locations of expected solutions are determined by interpolation, and sub-optimizations are conducted along the lines that connect the pseudo nadir point and the expected solutions. The actual solutions may be different from expected solutions, and there can be dominated solutions, which must be deleted by a Pareto filter.

The position vector of the jth expected solution on the piecewise linearized plane (P^j) is obtained as the weighed sum of the four vectors of the nodal solutions through Equation **(2.51)**:

$$p^j = \beta_1 N^1 + \beta_2 N^2 + \beta_3 N^3 + \beta_4 N^4, \qquad \beta_i \in [0,1] \tag{2.51}$$

where N^i is the position vector the ith node of a Pareto-front patch, and β_i is a weighting factor for interpolation.

The normalized vector of P^j is obtained according to Equation **(2.52)**:

$$\overline{P}_i^j = \frac{P_i^j - J_i^{Utopia}}{J_i^{Nadir} - J_i^{Utopia}} \tag{2.52}$$

where P_i^j is the ith coordinate of the jth expected solution on the piecewise linearized (hyper-) plane. The refinement level, which is represented by the step size of weighting factors, is determined based on the relative average length of the patch in each direction.

[Step 6] Impose an additional equality constraint (as illustrated in Figure **(2.17)** visiually) for each expected solution and conduct a sub-optimization with the weighted sum method. For the jth normalized expected solution, $\overline{P}^j$, the sub-optimization problem is provided in Exhibit **(2.3)** according to Kim and De Weck (2012).

minimize	$w_i * \overline{J}(x)$
subject to	$\dfrac{(\overline{P}^j - \overline{J}^{Nadir}).(\overline{J}(x) - \overline{J}^{Nadir})}{\left\lvert\overline{P}^j - \overline{J}^{Nadir}\right\rvert \left\lvert\overline{J}(x) - \overline{J}^{Nadir}\right\rvert} = 1$
	$\overline{h}(x) = 0$
	$\overline{g}(x) \leq 0$

Exhibit 2.3. Sub-Optimization Problem for the jth Normalized Expected Solution, ($P3$), (Kim and De Weck, 2012).

where $w_i = -(\bar{P}^j - \bar{J}^{Nadir})$ is a vector of weighting factors, $\bar{h}(x)$ and $\bar{g}(x)$ are normalized equality and inequality constraint vectors. The normalized nadir point $\bar{J}^{Nadir}$ is a vector whose components are one, i.e. $\bar{J}^{Nadir} = (1,1,...,1)$.

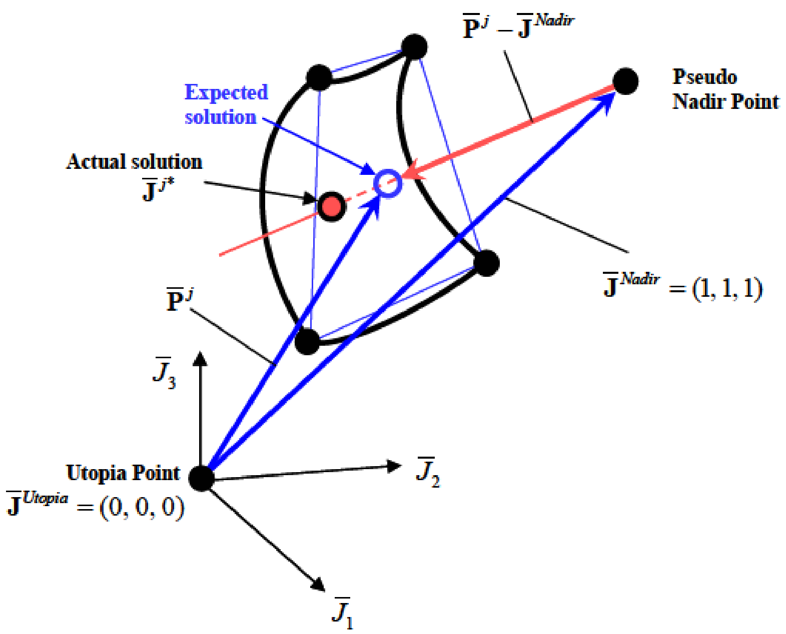


Figure 2.17. Configuration of an Additional Equality Constraint for Refinement (3-D Representation) (Kim and De Weck, 2012).

Kim and De Weck (2012) explain the model represented in Exhibit **(2.3)** as follows: The equality constraint $\left((\bar{P}^j - \bar{J}^{Nadir}).(\bar{J}(x) - \bar{J}^{Nadir})\right) / \left(\left|\bar{P}^j - \bar{J}^{Nadir}\right|\left|\bar{J}(x) - \bar{J}^{Nadir}\right|\right) = 1$ makes the two vectors $\bar{P}^j - \bar{J}^{Nadir}$ and $\bar{J}(x) - \bar{J}^{Nadir}$ be collinear in the objective space. Therefore, this constraint ensures that the solution is obtained only along the line $\bar{P}^j - \bar{J}^{Nadir}$, which connects the expected solution on the piecewise linearized plane and the pseudo nadir point. The objective function $-(\bar{P}^j - \bar{J}^{Nadir}).\bar{J}(x)$ is a scalar function to be minimized determining the solution that is nearest to the utopia point in the direction of $-(\bar{P}^j - \bar{J}^{Nadir})$. The actual solution obtained for the *j*th normalized expected solution, $\bar{P}^{j*}$, would be different from the expected solution as In Figure **(2.17)**, the origin of the vector $\bar{P}^j - \bar{J}^{Nadir}$ is actually (0,0,0) but moved for better visualization.

[Step 7] Perform Pareto filtering. Bi-objective adaptive weighted sum method, where non-Pareto optimal solutions are automatically rejected, does not require filtering. Contrary, in the multi-objective adaptive weighted sum method, any solution that lies on the equality constraint is feasible, and non-Pareto optimal solutions may be obtained. In each step, it is necessary to perform Pareto filtering to obtain the true Pareto front (as illustrated in Figure **(2.18)**).

[Step 8] Delete overlapping solutions. Identify Pareto-front patches with all Pareto optimal solutions including newly obtained solutions in the previous steps. If a termination criterion is met, stop; otherwise go to Step 5. Kim and De Weck (2012) indicate several types of termination criteria:

1. The number of stages reaches a prescribed number;
2. The size of largest Pareto-front patch falls below a prescribed value;
3. The standard deviation among the sizes of all Pareto-front patches falls below a prescribed value.

They use the maximum number of stages as the termination criterion in their work.

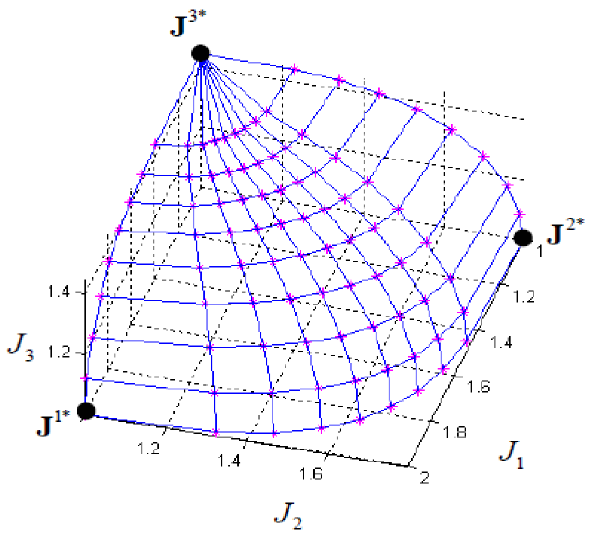


Figure 2.18. Pareto front Obtained by the Usual Weighted Sum Method (Kim and De Weck, 2012).

Kim and De Weck (2012) claim that the utopia distance method solves multi-objective optimization problems with more than two objective functions effectively. While bi-objective adaptive weighted sum method, which is applicable to only optimizations with two objective functions, uses inequality constraints to specify regions for further refinement; the utopia distance method, which is scalable to *n*-dimensional problems, uses equality constraints, allow modelers to decide where to obtain additional solutions, and this makes the Pareto front mesh well conditioned.

Recently, Dağ and Miman (2014) propose a multi-objective optimization model for the sustainability of a CLN based on utopia distance approach. Their model tries to minimize the weighted utopia distance between any point objective space and ideal solution in an effort to maximizing the sustainability while minimizing the cost of selective maintenance and time required to perform maintenance activities. They take the ideal point as (1,0,0) which indicates having a CLN with reliability of one, and zero cost and time.

3

Sustainability Optimization of a Contingency Logistics Network: A Multi-Dimensional Knapsack Approach

This chapter constitutes the fundamental part of this study including the contribution to the literature on contingency logistics systems, filling the some of the voids there providing metaheuristics solutions, developing hybrid solutions to the sustainability model as well as a different multi-objective optimization approach based on utopia distance for this model as proposed by Miman (2008). To achieve this the rest of this chapter is organized as follows: First, the necessary details on the sustainability model is provided, based on the study of Miman (2008) to enable this study to be self-standing so that who reads the study better comprehend the contributions and make use of them. Next, the details on the application of traditional techniques; genetic algorithm (GA) and simulated annealing (SA) along with a hybrid heuristic, EDGASA, is provided in Section (3.2), whose robustness and effectiveness are compared to each other and evaluated based on an experimental design presented in Section (3.3). The last part of this chapter introduces utopia distance approach to model the sustainability of the CLN as a multi-objective optimization technique where objectives are maximizing the sustainability while minimizing the total costs and total time required by maintenance activities.

3.1. Sustainability Model for a CLN

The sustainability of a CLN was defined by Miman (2008) to model the maintenance alternatives in the contingency logistics network settings where the life of each link between central warehouse and operational sites is modeled though a Weibull distribution and the bases' reliabilities are obtained through interference between demand and supply. The rest of this section provides the necessary details about the sustainability optimization model for a CLN as proposed by Miman (2008), which is the focus of this study in order to make this study self-standing.

3.1.1. Notation List

As defined by Miman (2008) following the notation lists to model the sustainability of the network.

V : set of nodes in the CLN

V^{-} : set of nodes except the central depot, i.e., operational sites in the CLN

V^{o} : set of bases that can support the operation o

n : number of independent operational sites (nodes) = $\left\| V^{-} \right\|$

A : set of links, i.e. arcs in the CLN

A^{-} : set of links, from the central depot to operational nodes (bases), i.e, $\{(i, j): i = 0, j = 1, 2 \ldots n\}$

l_{ij} : denotes the link (i, j) between node $i \in V$ and node $j \in V$ such that $i \neq j$ and $(i, j) \in A$

$X_{ij}(k)$: binary variable that indicates the status of link (i, j) at the beginning of mission k such that

$$X_{ij}(k) = \begin{cases} 1 & \text{, if link } (i, j) \text{ is functioning at the beginning of mission } k \\ 0 & \text{, if link } (i, j) \text{ is failed at the beginning mission } k \end{cases}$$

$X_{ij}^{+}(k)$: binary variable that indicates the status of link (i, j) at the end of mission k such that

$$X_{ij}^{+}(k) = \begin{cases} 1 & \text{, if link } (i, j) \text{ is functioning at the end of mission } k \\ 0 & \text{, if link } (i, j) \text{ is failed at the end of mission } k \end{cases}$$

$x_j(k)$: binary variable that indicates the status of the node for mission k such that

$$x_j(k) = \begin{cases} 1 & \text{, if node } j \text{ is functioning} \\ 0 & \text{, if node } j \text{ is failed} \end{cases}$$

$x_j^c(k)$	: binary variable indicating the status of the cell j for mission k; $\min\left\{X_j^+(k), x_j(k)\right\}$ such that $x_j^c(k) = \begin{cases} 1 & \text{, if cell } j \text{ is functioning} \\ 0 & \text{, if cell } j \text{ is failed} \end{cases}$
$\vec{x}_c(k)$	: n-tuples of status vector for the cells for misison k; $\left(x_1^c(k), x_2^c(k), ..., x_n^c(k)\right)$
$\phi^{\text{CLN}}\left(\vec{x}_c(k)\right)$	: structure function for the CLN
F_{ij}	: time to failure distribution for the link (i, j)
$\eta_{ij}(k)$	: Weibull life distribution scale parameter for the link (i, j) for mission k
$\beta_{ij}(k)$	: Weibull life distribution shape parameter for the link (i, j) for mission k
$\eta_{ij}^+(k)$	: Weibull life distribution scale parameter for the upgraded link (i, j) for mission k
$\beta_{ij}^+(k)$	: Weibull life distribution shape parameter for the upgraded link (i, j) for mission k
$r_{ij}^l(k)$	: probability of link (i, j) survival during mission k given it exists at the beginning of the mission k
$R_{ij}^l(k)$	: probability that the link (i, j) survives, mission k
$R_j^n(k)$	: probability that the node j is mission-k capable based on the inference of demand and supply
$v_{ij}(k)$	: virtual age of the link (i, j) at the begining of mission k
$v_{ij}^+(k)$	: virtual age of the link (i, j) at the end of mission k
$e_{ij}(k)$	: repair effectiveness for the link (i, j) after mission k
$\tau_{ij}(k)$	: amount of time, transportation time, the link (i, j) is used during the mission k
$a_{ij}(k)$	: imperfect repair adjustment factor for the link (i, j) during the mission k
$R_j^c(k)$	: probability that the cell j is operable for the mision k
$R(k)$	: probability that the CLN is capable, i.e. reliable, for the mission k
D_j	: random variable that represents the demand emerged at the node j
S_j	: random variable that represents the amount suppliable to the node j

λ_j	:	rate of demand at the node j for exponential distribution
μ_j	:	rate of supplied to the node j for exponential distribution
ρ_j	:	failure probability of node j based on the inference
cmr_{ij}	:	cost to repair a failed link (i, j)
crl_{ij}	:	cost to replace a failed link (i, j) with an identical new link
crl_{ij}^{+}	:	cost to replace a failed link (i, j) with an improved type link
$crfl_{ij}$	:	cost to replace an existing old link (i, j) with an identical new link
$crfl_{ij}^{+}$	:	cost to replace an existing link (i, j) with an improved type link
tmr_{ij}	:	time required to repair a failed link (i, j)
trl_{ij}	:	time required to replace a failed link (i, j) with an identical new link
trl_{ij}^{+}	:	time required to replace a failed link (i, j) with an improved type link
$trfl_{ij}$	:	time required to replace an existing link (i, j) with an identical new link
$trfl_{ij}^{+}$	:	time required to replace an existing link (i, j) with an improved type link
$C_{ij}(k)$	:	cost associated with the maintanenace activity peformed after mission k for the link (i, j)
$T_{ij}(k)$	:	time associated with the maintanenace activity performed after mission k for the link (i, j)
$C(k)$	:	total cost of maintanenace activities performed after the mission k in the CLN
$T(k)$	:	total time required for maintanenace activities performed after the mission k in the CLN
$C_0(k)$	:	Available budget for maintanenace activities to be performed after the mission k in the CLN
$T_0(k)$	:	Available time for maintanenace activities to be performed after the mission k in the CLN
$W_{ij}(k)$	:	binary variable indicating whether the link (i, j) is repaired after mission k
$V_{ij}(k)$	:	binary variable indicating whether the link (i, j) is replaced with an new identical new one after mission k
$Z_{ij}(k)$	:	binary variable indicating whether the link (i, j) is replaced with an new improved one after missionk

3.1.2. Modeling Node Reliability

The node reliability of base *j* in a CLN represents the ability of the operational base *j* to be able to perform the mission with critical supplies that are required by the base. Considering the random nature of a contingency, the node reliability can be modeled through interference theory between demand and supply, which are given by Equation **(3.1)** and Equation **(3.2)**.

$$\begin{aligned} R_j^n(k) &= P\left(x_j(k)=1\right) = 1 - P\left(x_j(k)=0\right) = 1-\rho_j(k) \\ &= 1 - P\left(S_j(\mathrm{k}) < D_j(k)\right) \end{aligned} \tag{3.1}$$

$$\rho_j = \Pr\{S_j < D_j\} = \begin{cases} \sum_y S_J(y) g_j(y) & \text{for discrete } (S_j, D_j) \\ \int_y S_J(y) g_j(y) & \text{for continuous } (S_j, D_j) \end{cases} \tag{3.2}$$

If $S_j(\mathrm{k}) \sim Expo(\mu_j)$ and $D_j(\mathrm{k}) \sim Expo(\lambda_j)$ are assumed, then Equation **(3.1)** can be reduced to Equation **(3.3)**.

$$R_j^n(k) = \frac{\lambda_j}{\lambda_j + \mu_j} \tag{3.3}$$

3.1.3. Modeling Link Reliability

Generally, the link (*i, j*) can be modeled in a variety of ways in order to capture a variety of insights for a CLN in terms of disruptions in the network. For example, the link can be regarded as the reliability of a specific transportation mode from node *i* to node *j*. This section models the reliability of links under the assumption that the life of the link in a contingency operation may be a function of how long it has been in operation before it fails or is disrupted. That is, the link is assumed to follow a

Weibull distribution (note that by specifying parameters of the Weibull distributions different maintenance characteristics can be captured) and, upon conclusion of a mission, the decision maker has multiple options: minimally repair a failed or broken link, replace a failed or disconnected link, replace a functioning link, or upgrade a link in order to improve its performance. Considering the links only between the depot and operational bases, i.e. the domain of links is A^- and the lateral shipments are ignored, the subscript i in the link reliabilities can be omitted as it is zero for all the links in consideration. The final link j reliability can be expressed by Exhibit **(3.1)** based on Miman (2008):

$$R_j^l(k+1) = P\left(X_j^+(k+1)=1\right) = r_j^l(k+1)X_j(k+1)$$

$$= \exp\left[-\left(\frac{\tau_j(k+1)+\upsilon_j(k+1)}{\eta_j(k+1)}\right)^{\beta_j(k+1)} + \left(\frac{\upsilon_j(k+1)}{\eta_j(k+1)}\right)^{\beta_j(k+1)}\right]X_j(k+1) \quad \text{(o)}$$

where

$$X_j(k+1) = X_j^+(k) + W_j(k)\left(1-X_j^+(k)\right) + V_j(k)\left(1-X_j^+(k)\right) + Z_j(k)\left(1-X_j^+(k)\right) \quad \text{(e1)}$$

$$\upsilon_j(k+1) = \upsilon_j^+(k) - \upsilon_j^+(k)V_j(k) - \upsilon_j^+(k)Z_j(k) - \left(1-a_j(k)\right)\upsilon_j^+(k)W_j(k) \quad \text{(e2)}$$

$$a_j(k) = \left(a_j(k-1) + e_j(k)W_j(k)\right)\left(1-V_j(\mathrm{k})\right)\left(1-Z_j(\mathrm{k})\right) \quad \text{(e3)}$$

$$\beta_j(k+1) = \beta_j(k)\left(1-Z_j(\mathrm{k})\right) + \beta_j^+(k)Z_j(\mathrm{k}) \quad \text{(e4)}$$

$$\eta_j(k+1) = \eta_j(k)\left(1-Z_j(\mathrm{k})\right) + \eta_j^+(k)Z_j(\mathrm{k}) \quad \text{(e5)}$$

subject to

$$W_j(k) + V_j(k) + Z_j(k) \le 1 \quad \text{(c1)}$$

$$W_j(k) \quad \text{binary} \quad \text{(d1)}$$

$$V_j(k) \quad \text{binary} \quad \text{(d2)}$$

$$Z_j(k) \quad \text{binary} \quad \text{(d3)}$$

Exhibit 3.1. Link j Reliability Definition in a CLN with Maintenance, (*P4*).

In Exhibit **(3.1)**;

(o) represents the reliability of the link (0 ,j) -*survival probability of link between the depot and base j for mission* (k+1).

(e1) represents the updated link j status for mission k+1 (at the beginning of the mission (k+1)) base on the status of the link j at the end of the mission k and the maintenance action taken at the end of the mission k.

(e2) represents the updated age of the link j at the beginning of the next mission (k+1) based on the age of the link j at the end of mission k, the percentage of improvement after repair is decreased during mission k, and the maintenance action taken at the end of the mission k.

(e3) represents the updated percentage of improvement after repair is decreased by a percentage for mission (k+1) repair based on percentage of improvement after repair is decreased for mission k, and the maintenance action taken at the end of the mission k.

(e4) represents the updated Weibull life distribution shape parameter for link j for mission (k+1) based on Weibull life distribution shape parameter for link j for mission k and Weibull life distribution shape parameter for upgraded link j for mission k and the maintenance action taken at the end of the mission k.

(e5) represents the updated Weibull life distribution scale parameter for link j for mission (k+1) based on Weibull life distribution scale parameter for link j for mission k and Weibull life distribution scale parameter for upgraded link j for mission k and the maintenance action taken at the end of the mission k.

(c1) represents the constraint that at most one of the maintenance actions (repair, replacement with the new identical one and replacement with the upgraded, i.e., superior or improved, one) on link j can be performed at the end of the mission k.

(d1) represents the decision variable concerning the repair of link j at the end of the mission k such that;

$$W_j(k)=\begin{cases}1 & \text{if the link } j \text{ is repaired after mission } k \\ & \text{but before mission } (k+1) \\ 0 & \text{otherwise}\end{cases} \qquad \textbf{(3.4)}$$

(d2) represents the decision variable concerning the replacement of link *j* with the new identical one at the end of the mission *k* such that;

$$V_j(k)=\begin{cases}1 & \text{if the link } j \text{ is replaced with a new identical one} \\ & \text{after mission } k \text{ but before mission } (k+1) \\ 0 & \text{otherwise}\end{cases} \qquad \textbf{(3.5)}$$

(d3) represents the decision variable concering the replacement of link *j* with the new improved (upgraded, superior) one at the end of the mission *k* such that;

$$Z_j(k)=\begin{cases}1 & \text{if the link } j \text{ is replaced with an improved one} \\ & \text{after mission } k \text{ but before mission } (k+1) \\ 0 & \text{otherwise}\end{cases} \qquad \textbf{(3.6)}$$

Note that, the link reliability is highly non-linear, in terms of decision variables.

3.1.4. Modeling the Sustainability of a CLN

As explained by Miman (2008) the links can be conceptualized as the transportation means from the depot to the base and operational base together with the link to the base can be regarded as an operational cell whose status can be expressed as a binary variable, $x_j^c(k)$. Then the CLN can be expressed as the arrangements of the *n*-cells, which constitute the *n*-tupelos of the status vector, composed of individual cell statuses expressed by $\vec{x}_c(k)$. The optimization model for the sustainability of the CLN developed by Miman (2008) as a multi-dimensional knapsack model, which is the focus of this study is presented in Exhibit **(3.2)** and Figure **(3.1)-(3.2)** In order to enable readers to have more concrete conceptualization of this study.

$$\max \quad R(k+1) = \underset{x_j^c \in \phi^{\mathrm{CLN}}(\vec{x}_c(k))}{f} \left(R_j^c(\mathrm{k}+1)\right) = \phi^{\mathrm{CLN}} \left(R_j^l(k+1) R_j^n(k+1)\right) \qquad (0)$$

subject to

$$C(k) \le C_0(k) \qquad (1)$$

$$T(k) \le T_0(k) \qquad (2)$$

$$W_j(k) + V_j(k) + Z_j(k) \le 1 \text{ for } j = 1, \dots, n \qquad (3)$$

$$W_j(k), V_j(k), Z_j(k) \quad \text{binary for } j = 1, \dots, n \qquad (4)$$

where for $\forall j \in A^-$:

$$R_j^l(k+1) = P\left(X_j^+(k+1) = 1\right) = r_j^l(k+1) X_j(k+1)$$

$$= \exp\left[-\left(\frac{\tau_j(k+1) + \upsilon_j(k+1)}{\eta_j(k+1)}\right)^{\beta_j(k+1)} + \left(\frac{\upsilon_j(k+1)}{\eta_j(k+1)}\right)^{\beta_j(k+1)}\right] X_j(k+1) \qquad (0.1)$$

$$X_j(k+1) = X_j^+(k) + W_j(k)\left(1 - X_j^+(k)\right) + V_j(k)\left(1 - X_j^+(k)\right) + Z_j(k)\left(1 - X_j^+(k)\right) \qquad (0.2)$$

$$\upsilon_j(k+1) = \upsilon_j^+(k) - \upsilon_j^+(k) V_j(k) - \upsilon_j^+(k) Z_j(k) - \left(1 - a_j(k)\right) \upsilon_j^+(k) W_j(k) \qquad (0.3)$$

$$a_j(k) = \left(a_j(k-1) + e_j(k) W_j(k)\right)\left(1 - V_j(\mathrm{k})\right)\left(1 - Z_j(\mathrm{k})\right) \qquad (0.4)$$

$$\beta_j(k+1) = \beta_j(k)\left(1 - Z_j(\mathrm{k})\right) + \beta_j^+(k) Z_j(\mathrm{k}) \qquad (0.5)$$

$$\eta_j(k+1) = \eta_j(k)\left(1 - Z_j(\mathrm{k})\right) + \eta_j^+(k) Z_j(\mathrm{k}) \qquad (0.6)$$

$$C(k) = \sum_{j \in A^-} C_j(k) \qquad (1.1)$$

$$C_j(k) = CMR_j(k) + CRL_j(k) + CRL_j^+(k) + CRFL_j(k) + CRFL_j^+(k) \qquad (1.1.1)$$

$$CMR_j(k) = cmr_j W_j(k) \qquad (1.1.1.1)$$

$$CRL_j(k) = crl_j V_j(k)\left(1 - X_j^+(k)\right) \qquad (1.1.1.2)$$

$$CRL_j^+(k) = crl_j^+ Z_j(k)\left(1 - X_j^+(k)\right) \qquad (1.1.1.3)$$

$$CRFL_j(k) = crfl_j V_{ij}(k) X_j^+(k) \qquad (1.1.1.4)$$

$$CRFL_j^+(k) = crfl_j^+ Z_j(k) X_j^+(k) \qquad (1.1.1.5)$$

$$T(k) = \sum_{j \in A^-} T_j(k) \qquad (2.1)$$

$$T_j(k) = TMR_j(k) + TRL_j(k) + TRL_j^+(k) + TRFL_j(k) + TRFL_j^+(k) \qquad (2.1.1)$$

$$TMR_j(k) = tmr_j W_j(k) \qquad (2.1.1.1)$$

$$TRL_j(k) = trl_j V_j(k)\left(1 - X_j^+(k)\right) \qquad (2.1.1.2)$$

$$TRL_j^+(k) = trl_j^+ Z_j(k)\left(1 - X_j^+(k)\right) \qquad (2.1.1.3)$$

$$TRFL_j(k) = trfl_j V_{ij}(k) X_j^+(k) \qquad (2.1.1.4)$$

$$TRFL_j^+(k) = trfl_j^+ Z_j(k) X_j^+(k) \qquad (2.1.1.5)$$

Exhibit 3.2. The Sustainability Optimization Model for a CLN, (*P5*), (Based on Miman, 2008).

In Model (*P5*);

(0): represents the sustainability of the CLN (the CLN survives for mission (k+1)), which depends on the structure of the logistics network, which can be expressed using the structure function $\phi^{\text{CLN}}\left(\vec{x}_c(k)\right)$, to be maximized. Note that (0.1)-(0.6) have the same meanings as those in Exhibit **(3.1)**.

(1) : represents that the total cost of maintenance activities performed on links in the CLN at the end of mission k, should not exceed the available budget at the end of mission k for maintenance activities.

(1.1): represents the total cost associated with the maintenance activities on the CLN while (1.1.1) represents the cost of maintenance activity performed on link j that is composed of the mutually exclusive five cost components; cost due to the link repair, cost associated with replacing a failed link with a new identical one, cost due to replacement of failed link with a new improved one, cost due to replacement of the functioning link with a new identical link and cost due to replacement of functioning link with a new improved one as represented by (1.1.1.1)-(1.1.1.5) respectively.

Similarly,

(2) : represents the total time required to perform maintenance activities on links in the CLN at the end of mission k, should not exceed the available time at the end of mission k for maintenance activities.

(2.1): represents the total time associated with the maintenance activities on the CLN while (2.1.1) represents the time of maintenance activity performed on link j that is composed of the mutually exclusive five time components ; time required to repair the link, time associated with replacing a failed link with a new identical one, time required to replace the failed link with a new improved one, time required to replace the functioning link with a new identical link and time required to replace functioning link with a new improved one as represented by (2.1.1.1)-(2.1.1.5) respectively.

It is apparent from Exhibit **(3.2)** that the sustainability model which is focused on this study is a non-linear, non-convex and non-separable mathematical program modeled as a multi-dimensional knapsack problem, which motivates this study in search of effective and robust optimization techniques whose details are provided in Section (3.2).

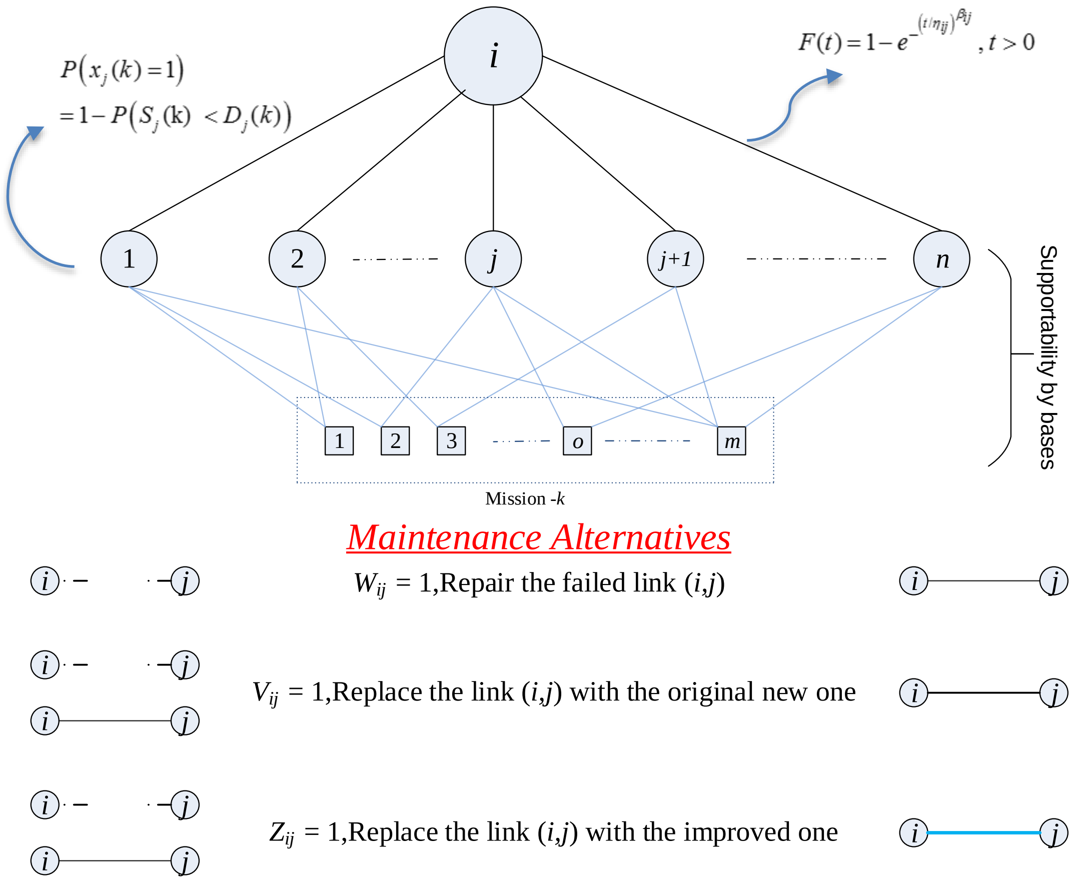


Figure 3.1. Network Representation of the CLN in Consideration.

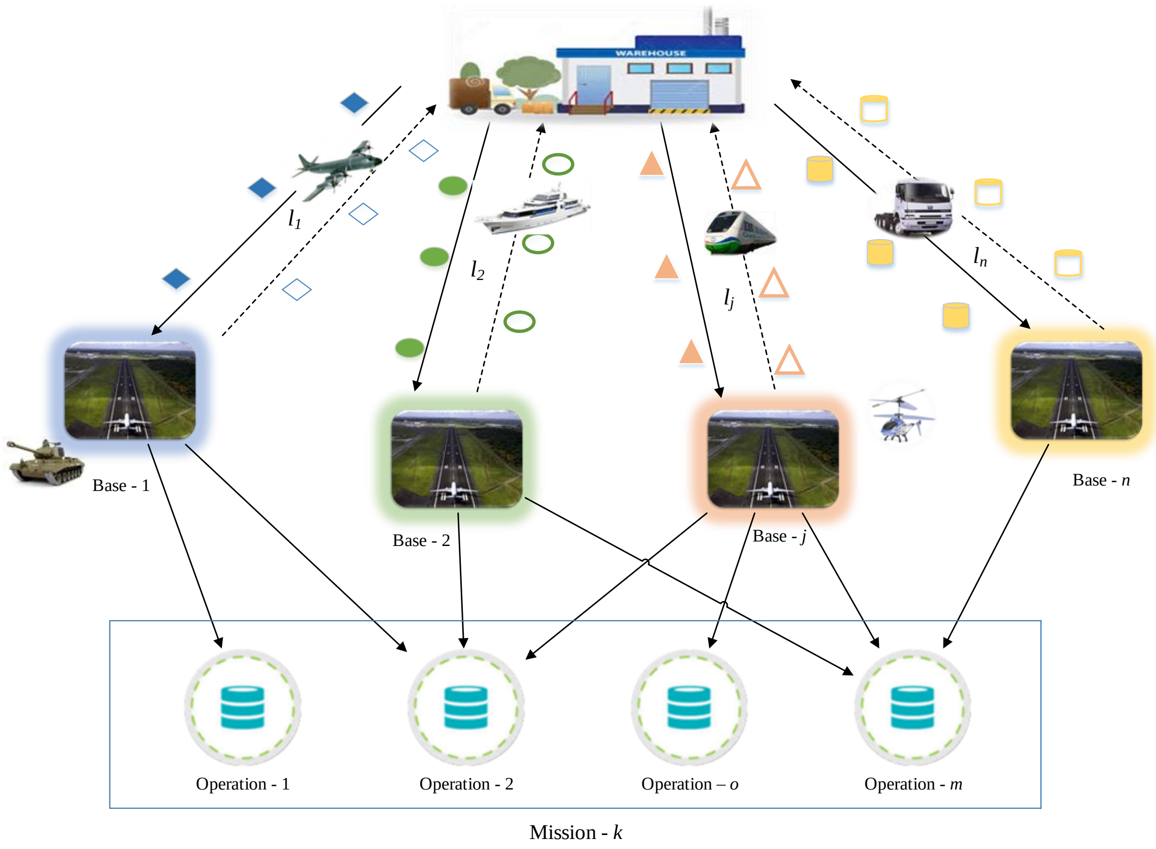


Figure 3.2. CLN Sustainability: Mission Perspective.

In Figure **(3.2)**, a mission-k emerges as a result of a contingency event and consists of sets of operations 1 to m that need to be accomplished to recover from the contingency. Each operation can be supported by a variety of bases provided that the bases (nodes) are mission capable, i.e. have enough supplies to perform the assigned task. If each base j can support only one operation o, which can eventually be supported by several bases and in order to mission-k to be achieved all operations o = 1..m must be achieved, then the reliability of the system can be calculated using the Equation **(3.7)**.

$$R(k) = \prod_{o} \coprod_{j \in V^o} (R_j^c) \tag{3.7}$$

3.2. Heuristic Approaches Investigated

This section provides the details about the traditional heuristic algorithms, GA, SA as well as the hybrid heuristic developed EDGASA, which are in consideration for investigation in search of robust and effective solution approach to the sustainability model which is presented in Exhibit **(3.3)** one more time to make conceptualization more concrete.

$$
\begin{array}{|lr|}
\hline
\max \quad R(k) & (0) \\
s.t. & \\
C(k) \le C_o(k) & (1) \\
T(k) \le T_o(k) & (2) \\
W_j(k) + V_j(k) + Z_j(k) \le 1 \quad , j=1..n & (3) \\
W_j(k) \quad \text{binary} \quad , j = 1,\ldots,n & (4) \\
V_j(k) \quad \text{binary} \quad , j = 1,\ldots,n & (5) \\
Z_j(k) \quad \text{binary} \quad , j = 1,\ldots,n & (6) \\
\hline
\end{array}
$$

Exhibit 3.3. Sustainability Model as an IP for a CLN in compact form, (*P6*).

In model (*P6*), (0) represents the network reliability to maximize based on the maintenance actions taken. (1) and (2) represents the constraints related to resources, budget and time respectively, available for the maintenance to perform. (3) represents that at most one of the maintenance alternatives (repair, replace with an original link and replaced with an improved link) can be selected to be performed for each link j=1,…,n. (4-6) represent the binary decision variables for each link j=1,…,n. Note that including do nothing there are four possibilities (do nothing, repair, replace with an identical link, replace with an improved link) which can be modeled as a two-bit system in a solution representation for a base depending on the values of decision variables (4-6 in (*P6*)) as shown in Table **(3.1)**.

Table 3.1. Two-bit Binary Representation for Maintenance Alternatives of link *j*.

Encoding	Action Taken
$\boxed{0}\boxed{0}$ $W_j = 0$; $V_j = 0$; $Z_j = 0$	None
$\boxed{0}\boxed{1}$ $W_j = 1$; $V_j = 0$; $Z_j = 0$	repair
$\boxed{1}\boxed{0}$ $W_j = 0$; $V_j = 1$; $Z_j = 0$	replacement with a new identical link
$\boxed{1}\boxed{1}$ $W_j = 0$; $V_j = 0$; $Z_j = 1$	replacement with a new improved link

Note that this two-bit representation of a solution for a base yields a solution encoded binary with a length of 2*n* for a CLN consisting of *n* operational sites (bases) as displayed in Figure **(3.3)**.

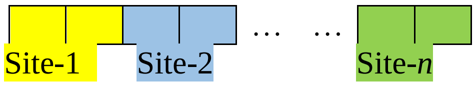


Figure 3.3. Encoding of a Solution for the CLN with *n*-bases.

3.2.1. Traditional Heuristic Approaches Investigated

This section provides the details on the pure GA and SA, which have been extensively used in the literature for a variety of problem structures where the problems are NP hard. In this study their details are provided so that one can better conceptualize the hybrid heuristic, EDGASA, developed and presented in Section (3.2.2). The selection of parameters for algorithms under investigation is based on the findings of Miman (2008), who first time investigated meta-heuristics for the presented sustainability model.

3.2.1.1. Genetic Algorithm

GA is an evolutionary technique where in this study the evaluation started from an initial population consisting of 50 individuals generated randomly. In each generation, the fitness of every individual in the population is computed. Infeasible solutions are penalized with a large penalty for each unit of excess cost and/or time over available amounts where the fitness of such infeasible solutions is assigned as one over the resulting absolute performance after the penalty is imposed. (For SA and EDGASA, the same penalty scheme is applied) Before generation of next population starts, the chromosomes are arranged according to their fitness according to which each chromosome in the population has a likelihood -through computeFR function- to be selected for crossing over mechanism as a parent. Then, the next generation in GA is obtained through the eliciting, crossing-over and mutation mechanisms as displayed in Figure **(3.4)**.

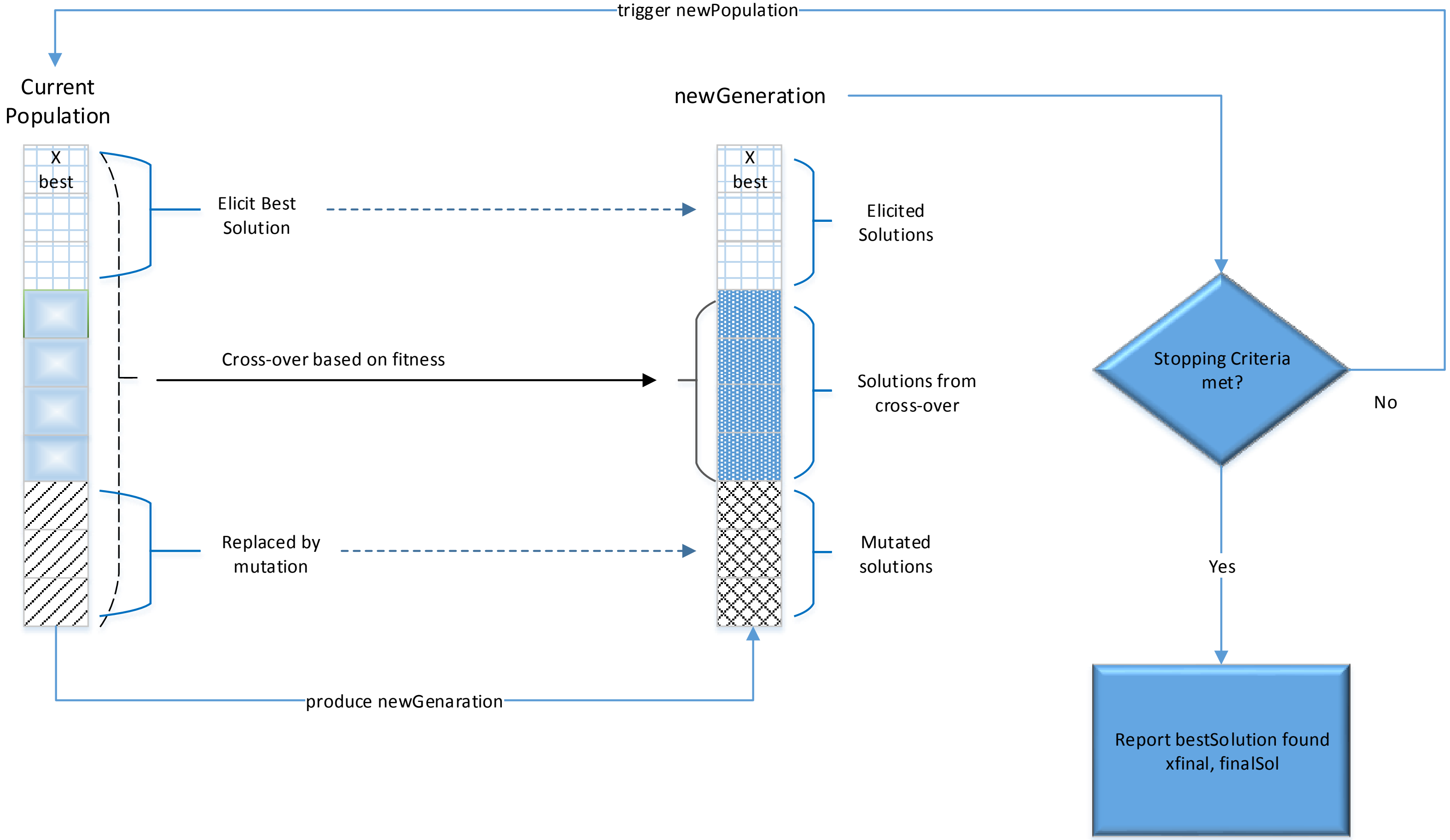


Figure 3.4. Next Population Generation in GA.

In eliciting, the 4% of the best chromosomes which has the highest fitness are transferred to the next generation as they are. In crossing-over, the 56 % the next population are obtained through a single crossing over mechanisms where parents are stochastically selected based on their fitness out of 50 Individuals from the current population and modified using a single crossover mechanism. In the single cross over mechanism a parents Exchange the solution parts from the single point that are selected randomly yielding two off-springs to be placed into next generation. Single bit mutation is applied (the value of the bit is reversed, i.e. one turns to zero, zero turns to one) to a random subset of the population 40% - and added to the new population. The new population is then used in the next iteration (generation) of the algorithm. The algorithm terminates when a maximum number of generations (250) have been produced. The essential codes of the GA are presented in Appendix 5. Note that the rates of eliciting, crossing over or mutation rates can be arranged depending on the problem size as well as dynamically during the execution time.

3.2.1.2. Simulated Annealing

SA is a metaheuristics originated by mimicking the behavior of annealing of the iron at different temperatures. In this application of SA to the sustainability model, initial solution is set to be “do nothing” for each base. To initialize the algorithm, the final solution (the best evaluated solution so far) as well as the current solution are set to the initial solution (do nothing), the initial temperature (tinitial =1000) thus final temperature (tfinal =tinitial/m where m=100) , the parameter for how many times new solutions are explored at the specified temperature (n1=500), and the parameter to find a new temperature (v1=1 initially, where v1 is increased by 1 for each iteration) are specified. For a given temperature (represented by t in the code where t=tinitial/v1) the algorithm searches new solution by changing a single bit of a current solution similar to single-bit mutation employed in GA.

If the solution found has superior objective function, then the current solution as well as the final solutions are updated to this new superior solution. If the solution found is worse than the final solution found so far, there is still a certain probability depending on the current temperature to accept this inferior solution as a current solution (in order to move a different point in the solution place to explore). This probability is given by Equation **(3.8)**.

$$U(0,1) < e^{-\mathrm{deltasol/t}} \quad \textbf{(3.8)}$$

At the end of the exhaustive attempts (number of times that is equal to n1=500) for a desirable solution at the current temperature, the current solution is updated to the best evaluated solution so far; i.e. final solution, and new temperature for which the solution space is to be explored staring from the current solution is set (through findtemparature function in the code).

The loop of search continues until the current temperature falls below the final temperature. These steps of simulated annealing algorithm are displayed in Figure **(3.5)**. The essential codes of the SA are presented in Appendix 6.

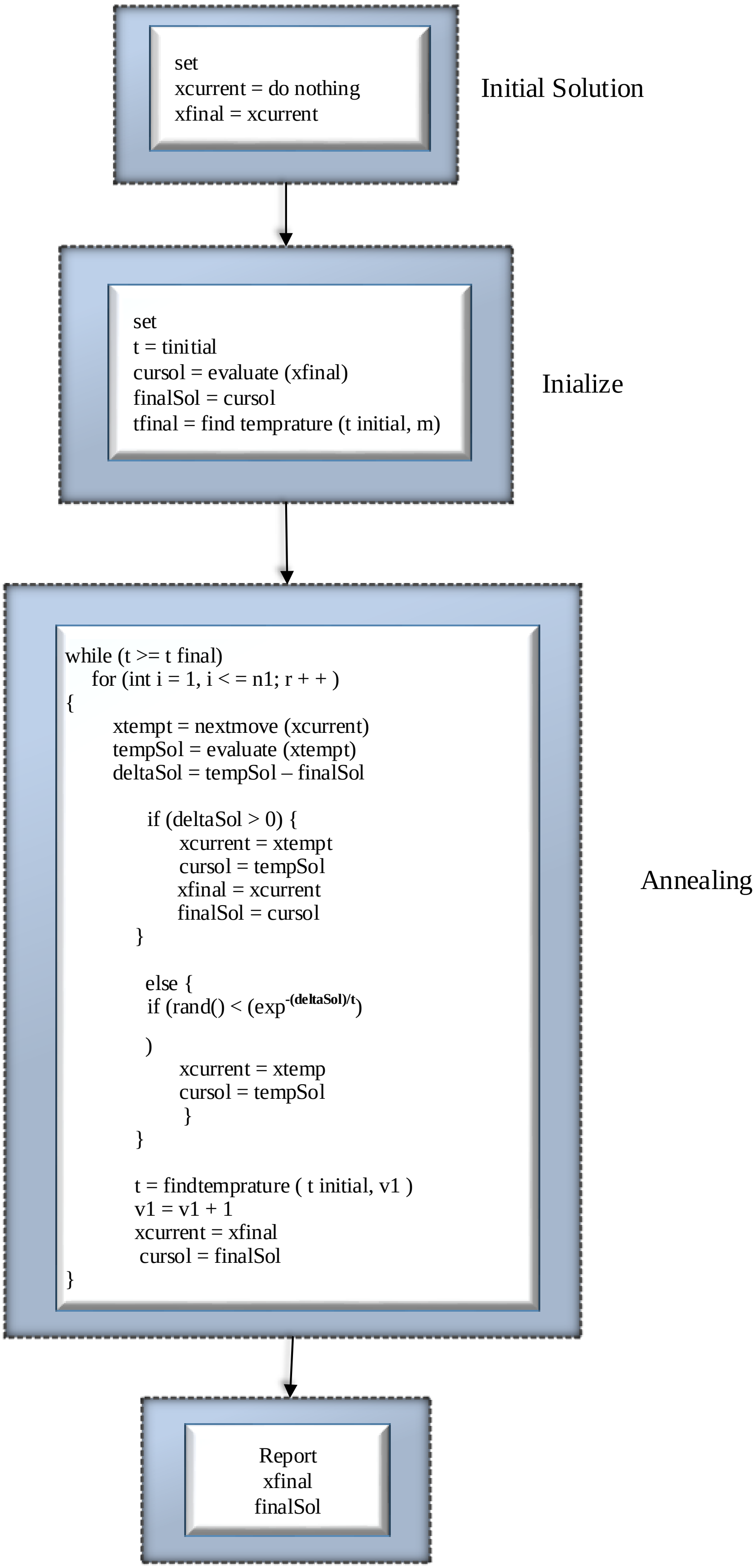


Figure 3.5. Steps of Simulated Annealing Algorithm Applied.

3.2.2. A Hybrid Heuristic: EDGASA (Genetic Algorithm with Simulating Annealing by Esra Dağ)

In literature GA is known to be a good technique to find a promising neighborhood for a good solution while, the SA is known to be a good technique to search exhaustively in a given neighborhood. This motivates to develop the hybrid heuristic, EDGASA, by combining the power of both of the heuristics. In this hybrid algorithm, the goal is to combine the ability of a GA to search a broad and diverse set of neighborhoods with the power of a SA that aggressively searches for better solutions in a specific neighborhood.

EDGASA combines the GA and SA into a hybrid algorithm for the non-separable, non-linear, non-convex, multi-dimensional knapsack problem discussed earlier. Inıtial population of EDGASA is obtained as follows: The specified number of solutions (indicated as “level” in Figure **(3.6)**, and recursive in java code as seen in Appendix 7) is obtained through annealing procedure while the rest of the solutions in the initial population is generated randomly.

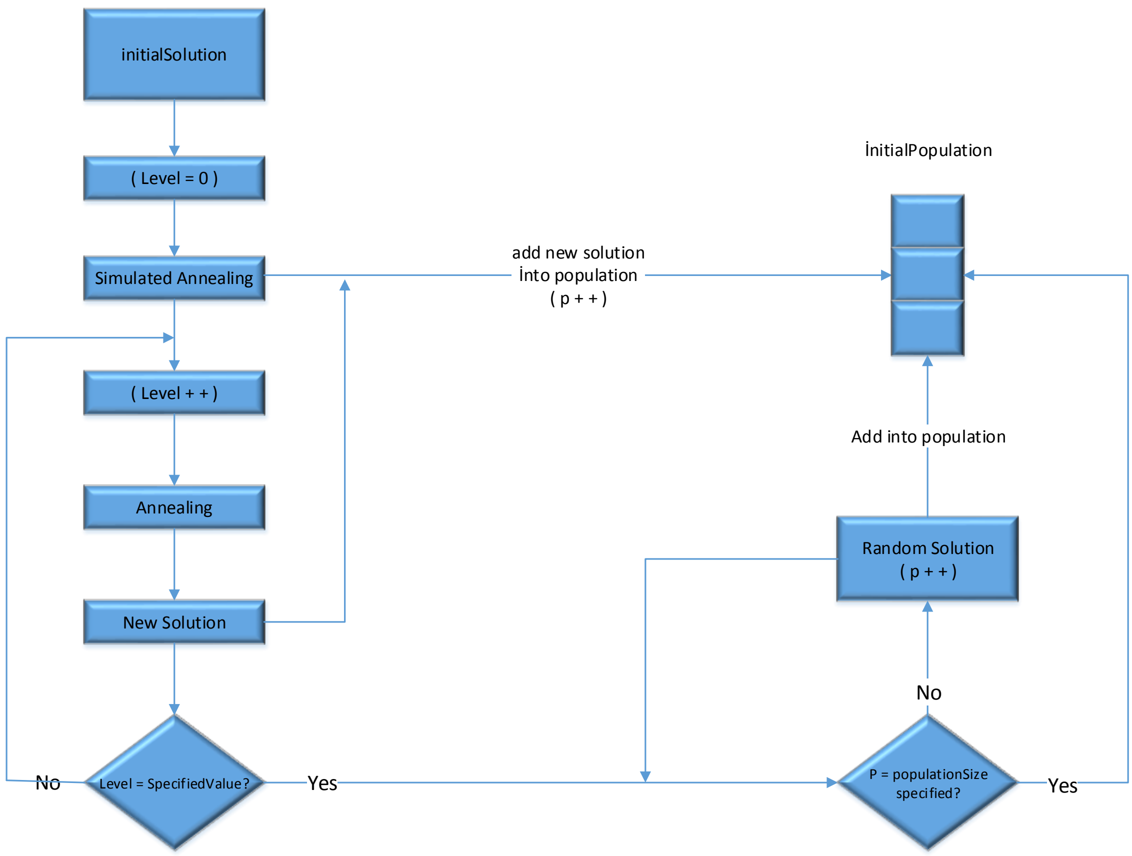


Figure3.6. Initial Population in EDGASA.

The next generation in EDGASA is obtained with the same mechanisms as those in GA with a difference in the eliciting mechanism as illustrated in Figure **(3.7)**. In eliciting in the hybrid heuristic developed, EDGASA, the best solution according to fitness are subject to annealing and the final solution obtained from annealing is placed into next population. This final solution is subject to annealing recursively number of times as specified by level (recursive=1 in this study). The crossing over and mutation mechanism are similar to those applied in GA.

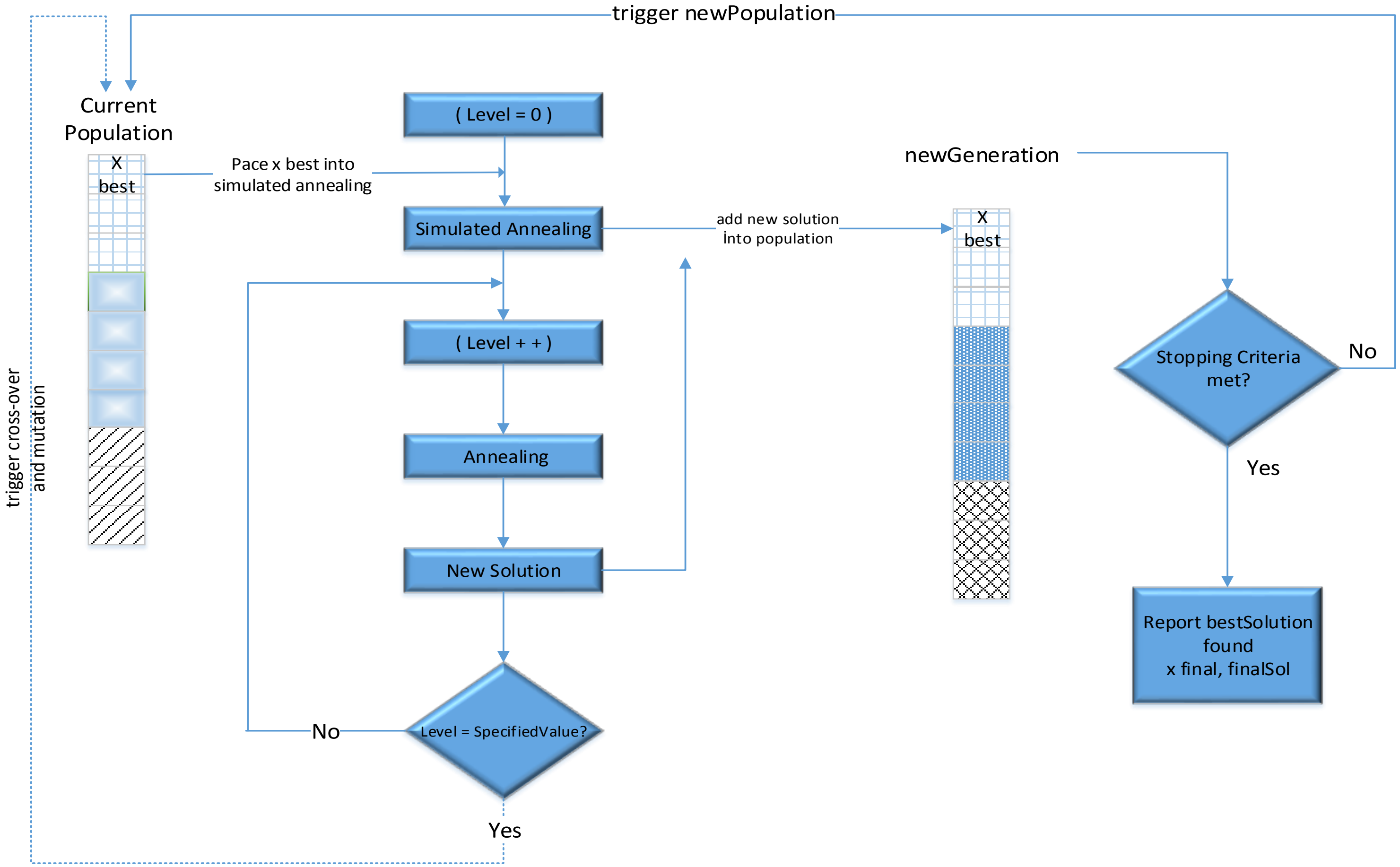


Figure 3.7. Generation Formation in EDGASA.

Table **(3.2)** compares these three algorithms investigated in this study in terms of initial solution/population, mechanism to obtain next solution/population and stopping criteria they use.

Table 3.2. Comparison of Algorithms based on their Mechanisms.

Algorithm Applied	**The way initial solution / initial population generated**	**The way next solution / next population generated**	**Stopping criteria**
SA	Single solution: do nothing	Single bit change in current solution	(t<tfinal)
GA	50 solutions: generated randomly	Eliciting of some percent of best solutions	Specified generation number (250)
		Single point crossing over depending on the fitness	
		Single bit mutation of some percent of worst solutions	
EDGASA	50 solutions: starting from do nothing solution the specified number of solutions are obtained through the application of annealing recursively. The rest of the solutions are generated randomly	Eliciting of the best solution along with application of annealing	Specified generation number (250 as it is in GA)
		Single point crossing over as it is in GA	
		Single bit mutation as it is in GA	

Table **(3.2)** indicates the mechanisms employed by each of the heuristics for which a set of experiments are designed in Section (3.3) to compare the efficiency and robustness of their performances.

3.3. Comparisons of Algorithms Investigated

This section provides the details about how algorithms specified in Section (3.2) are investigated according to their performance on solution quality and solution time. Specifically, it presents experimental design used for this comparison and the results of analysis.

3.3.1. Experimental Design

To compare the performances (quality of the solution found, and the time to find that solution) of each algorithms considered (namely GA, SA, EDGASA), a set of experiments are performed according to the design described below. This also compares the robustness and efficiency of each of the algorithms. The purpose of the experiment design is to illustrate the relative strengths of each of the algorithms based on random cases. To do this 100 experimental cases are generated randomly and each of these cases are solved 10 times by each of the algorithms to get their performances (average quality of the solution (Average-*R*), the best solution found out of 10 attempts (*R*-max), the worst solution found out of 10 attempts (*R*-min) and average solution times) of them for comparison.

In generating random experiment cases, the same approach used by Miman (2008) is employed. Specifically, in generating original and improved technologies' failure parameters, it is assumed that the improved links have a longer meantime to failure, i.e. $\eta_j^+ \Gamma(1+1/\beta_j^+) > \eta_j \Gamma(1+1/\beta_j)$. This can be modeled by setting the expected scale, η_j^+, and shape parameter β_j^+ for the improved links greater than those of original systems; η_j and β_j respectively. In terms of specifying cost parameters, the average imperfect maintenance cost, i.e., repair activity cost, cmr_j, is assumed to be less than that of average cost of perfect maintenance of replacements with original and improved links where average improved technology cost is greater than that of the original one.

Further, that replacement costs are supposed to be greater when the link is functioning (This might be because of different procedures to replace or due to the disposing of the functional link). Similarly, the average time to repair a link is assumed to be greater than the average time for replacements where replacement with the improved link is assumed to require more time than replacement with the original link due to potential setup and procedure changes.

The Table **(3.3)** provides the assumed expected values for cost and time requirements for each of the maintenance alternatives $k \in M$ where k = 1 refers do nothing k = 2 refers to repair, k = 3 refers to replacement of a failed link with an identical link k = 4 refers to the replacement of a functioning link with an identical link as described by Miman (2008). The plus indicates replacement with upgraded (superior) link.

Table 3.3. Expected Values for Decision Alternatives.

M_k	E[T_k] (hrs)	E[C_k] ($)
1	0	0
2	45	30
3	30	70
3^+	35	100
4	32	80
4^+	40	120

The random variables of maintenance cost and time are modeled through uniform distributions such that $T_k \sim U\left(LL_T^k, UL_T^k\right)$ and $C_k \sim C\left(LL_C^k, UL_C^k\right)$. The intervals are allowed to overlap in order to represent all possible conditions such as if a new technology is not readily available, the required time may become longer etc. Expected maintenance requirements (in terms of time and budget) for a risk neutral decision maker are taken into account in determination of available budget and cost which are modeled to be uniformly distributed as $C_o \sim C_o\left(LL_C^o, UL_C^o\right)$ and $T_o \sim T_o\left(LL_T^o, UL_T^o\right)$ respectively: For a single link, the expected maintenance cost can be given by Equation **(3.9)**. For a risk neutral decision maker, each of the three alternatives: repair, replacement with identical link, replacement with improved link are equally likely for a failed link while replacement with original link and replacement with improved link is also equally likely for a functioning link where a link at the beginning of maintenance decision is equally likely to be found failed or functioning as described by Miman (2008).

$$E[C]=\frac{1}{2}\left(\frac{E[C_2]+E[C_3]+E[C_3^+]}{3}\right)+\frac{1}{2}\left(\frac{E[C_4]+E[C_4^+]}{2}\right) \quad \textbf{(3.9)}$$

Similarly, for a single link the expected maintenance time can be given by Equation **(3.10)**.

$$E[T]=\frac{1}{2}\left(\frac{E[T_2]+E[T_3]+E[T_3^+]}{3}\right)+\frac{1}{2}\left(\frac{E[T_4]+E[T_4^+]}{2}\right) \quad \textbf{(3.10)}$$

According to experimental design, expected maintenance cost and time parameters specified in Table **(3.3)**, a single link's expected maintenance cost and time for a neutral decision-maker are (do nothing is a default action hence excluded from computations in terms of budget and time requirements) found to be approximately \$85.00 and 36.58 hrs respectively. Therefore for a 16-links system considered in this study whose reliability block diagram can be seen from Figure **(3.8)**, the expected cost and time required for maintenance alternatives are \$85.00×16=\$1360 and 36,58 hrs×16=585,28 hrs respectively. While budget and time availabilities are represented in the experimental design by uniformly distributed random variables, their expected values are modeled to be in accordance with approximately the half of the risk neutral CLN maintenance requirements, i.e. close to 700 and 290 respectively. To ensure at least one maintenance activity can be performed the lower limits of these distributions are set to be $LL_C^o \geq \underset{k}{Max}\{UL_C^k\}$ and $LL_T^o \geq \underset{k}{Max}\{UL_T^k\}$.

Further, the node reliabilities are generated through a U(0.84, 0.99) distribution in place of the interference model as current main interest here is the maintenance alternatives on the links. Table **(3.4)** displays the associated parameters for the experimental designs that are generated to test the effectiveness of the each of the algorithms:

Table 3.4. Experimental Design Parameters.

Parameter	Distribution
η_j	U(50,150)
β_j	U(1.1,2.1)
$v_j(k)$	U(0,50)
η_j^+	U(90,210)
β_j^+	U(1.5,3.0)
a_j	U(0.35,0.95)
$e_j(k)$	U(0.05,0.25)
$\tau_j(k)$	U(75,100)
cmr_j	U(20,40)
crl_j	U(50,90)
crl_j^+	U(70, 130)
$crfl_j$	U(60,100)
$crfl_j^+$	U(90,150)
tmr_j	U(30,60)
trl_j	U(20,40)
trl_j^+	U(25,45)
$trfl_j$	U(30,34)
$trfl_j^+$	U(35,45)
$R_j^n(k)$	U(0.84,0.99)
$C_0(k)$	U(550,750)
$T_0(k)$	U(200,380)

Note that, the failure parameters of the links ($\eta_j, \beta_j, v_j(k), \eta_j^+, \beta_j^+, a_j, e_j(k), \tau_j(k)$) are generated through exactly the same probability distributions used by Miman (2008) while the maintenance costs and time parameters are generated through the distributions specified in Table **(3.4)**.

In the experimental study, 100 random instances of the problem were generated and each instance was solved using the GA, SA, and EDGASA heuristics 10 times to compare the robustness and efficiency of each of the algorithms. The appropriateness of this approach is based on Miman (2008). Each of the heuristics has the do-nothing option as an initial solution and the remainder of the initial solution set is generated randomly. For each problem instance, 10 replications are performed for those heuristics that have stochastic components: GA, SA and EDGASA and the average performance across 10 replications as well as the best and worst solutions found were used to compare these heuristics. The goal of this experimentation is to be able to analyze the relative performance of each of the heuristics on this class of problem introduced by Miman (2008).

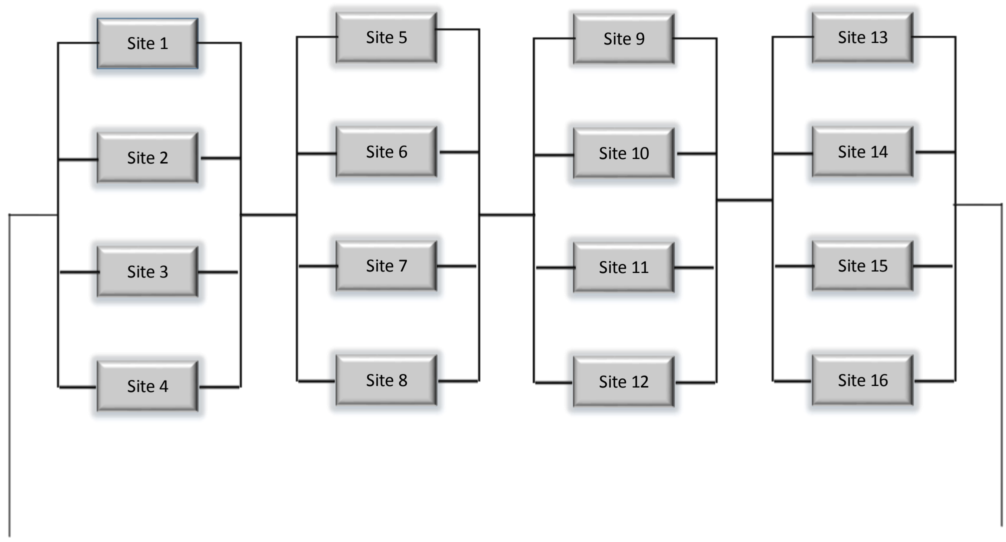


Figure 3.8. The Structure of the CLN Considered for the Experimentation.

In generating experimental designs, Eclipse is used as an Integrated Development Environment (IDE) for codes developed in Java Programming Language. It is also used to code the solution algorithms in consideration. Particular to the generation of an experimental design for each problem instance and collection of the statistics about the performance of each of the algorithms (average quality of the solution (Average-*R*) for 10 attempts to each problem instance, the best solution found out of 10 attempts (*R*-max) to each problem instance, the worst solution found out of 10 attempts (*R*-min) to each problem instance and average solution times) Java Simulation Library (JSL) version jsl-Beta-r4-v19 is used. Specifically, random package of JSL is used to generate each problem instance according to parameters specified in Table **(3.4)** and statistics package of JSL is used to collect the statistics about performances of each of the algorithm. The detailed information about JSL is provided by Rossetti (2008) who has developed the software kit, which is available at URL -2. The detailed results associated with each experimental setups solved through GA, SA and EDGASA are provided in Appendix 8, 9 and 10 respectively.

3.3.2 Analysis of Results

Using the contingency logistics models for selective maintenance alternatives' optimization previously formulated and the associated 100 randomly generated problem instances, the performance of each of the solution approaches: GA, SA, and EDGASA is investigated. This will give greater insights into solution approaches to adopt in response to the investigation of each algorithm's robustness to such problem structures as an extension of study conducted by Miman (2008) who analyzed the GA, TS, MULRR and GAFTS algorithms. The descriptive statistics for each algorithm are analyzed in Minitab and displayed in Table **(3.5)** below.

Table 1.5. Descriptive Statistics for Heuristic Solutions in terms of Average Reliability.

Variable	Method	N	N*	Mean	SE Mean	StDev	CoefVar	Minimum	Q1	Median
AverageR	EDGASA	100	0	0,76419	0,00628	0,06279	8,22	0,55003	0,72521	0,76941
	GA	100	0	0,5767	0,0148	0,1477	25,60	0,1091	0,4693	0,6135
	SA	100	0	0,68381	0,00831	0,08310	12,15	0,42879	0,63132	0,69394

Variable	Method	Q3	Maximum
AverageR	EDGASA	0,81137	0,89744
	GA	0,6822	0,8050
	SA	0,75049	0,82540

The results displayed in Table **(3.5)** indicated that EDGASA dominates all other algorithms (GA and SA) with regard to the performances across the problem domain for mean, median, minimum and maximum reliabilities for average of 10 attempts to each problem instances' solution of the network. It also has the lowest coefficient of variation which suggests that the algorithm is more robust than the others. Further, the results are analyzed statistically as illustrated in Table **(3.6)** according to the procedures employed by Rajagopalan and Cassady (2006) and later by Miman (2008) paired t-test on differences between mean performances on Average-R each algorithm finds based on 10 replications.

Table 3.6. Hypothesis Testing for the Average Reliability of a CLN by Heuristics.

Test	Hypothesis	T-Value	P-Value	95%CI on the mean difference	Conclusion
1	$H_0 : \mu_R^{GA} = \mu_R^{SA}$ $H_1 : \mu_R^{GA} \neq \mu_R^{SA}$	-12.44	0.000	(-12422 , -0.09004)	$\mu_R^{SA} > \mu_R^{GA}$
2	$H_0 : \mu_R^{EDGASA} = \mu_R^{SA}$ $H_1 : \mu_R^{EDGASA} \neq \mu_R^{SA}$	24,64	0.000	(0.07391 , 0.08686)	$\mu_R^{EDGASA} > \mu_R^{SA}$

A paired t-test, as performed by Rajagopalan and Cassady, (2006) and later by Miman (2008), is employed to compare the improvements through heuristics. First the evolutionary technique, GA was compared with simulated annealing technique, SA. The paired t-test indicates that $\mu_R^{SA} > \mu_R^{GA}$ for the CLN built by the parameters tabulated in Table **(3.4)**. In the second test, the hybrid heuristic where genetic algorithm is fed by the solutions obtained through simulated annealing, EDGASA, was compared against the pure classical SA. The analysis concludes that $\mu_R^{EDGASA} > \mu_R^{SA}$. When these two results are combined, the hybrid heuristics developed and proposed by this thesis, EDGASA, found to be superior to both of the traditional techniques GA and SA.

$$\mu_R^{EDGASA} > \mu_R^{SA} > \mu_R^{GA} \tag{3.11}$$

These results verify the research logic behind the hybrid algorithm, EDGASA, in combining the power of the SA in aggressively exploring the neighborhood with the power of GA in finding a good neighborhood. As a result, the performance of the pure SA is improved through the utilization of the neighborhood search of the GA in EDGASA.

Based on the initial analysis, EDGASA is found to be superior to all the other heuristics under consideration. Therefore, it can provide greater insights about the best solution approaches for the CLN design and can be further compared to GAFTS, which is proposed by Miman (2008) as the best solution approach for the problem structure for the sustainability of the CLN studied. To provide a measure of statistical confidence on the comparisons of all alternative heuristics, (k = 3) a multi-comparison approach as described by Goldsman and Nelson (1998) was performed:

Let $(1-\alpha)$% be the confidence over all of the comparisons of $\{\mu_i - \mu_0\}$

where $0 = EDGASA$ & $i \in \{GA, SA\}$ such that

$$\mu_i - \mu_0 \le d = \bar{Y}_i - \bar{Y}_0 + \sqrt{\frac{t^2_{1-\beta,n_0-1} S_0^2}{n_0} + \frac{t^2_{1-\beta,n_i-1} S_i^2}{n_i}} \qquad \textbf{(3.12)}$$

where $(1-\beta) = (1-\alpha)^{1/(k-1)}$

If $d \le 0$, then the adopted approach, EDGASA, should be kept in use as it does a better job at identifying superior solutions. Table **(3.7)** contains the results of the simultaneous multiple comparison test, which is also used by Miman (2008).

Table 3.7. Multiple Comparisons with EDGASA.

	I		
Performance	EDGASA	GA	SA
$\bar{Y}_i$	0.76419	0.57670	0.68381
S_i^2	0.00394	0.02182	0.00691
n_i	100	100	100
d	N/A	-0.15695	-0.06056
heuristic choice		EDGASA	EDGASA

Note that for the above multiple comparisons: $(1-\beta) = 0.95^{1/2} = 0.97467$ for an overall 95% confidence for the all possible comparisons with EDGASA. Now it can be said that the hybrid algorithm's (EDGASA's) performance is better than any other traditional metaheuristics examined on this problem with 95% confidence.

This results are verified by ANOVA analysis conducted by Minitab v17.0 with Tukey post hoc tests for simultaneous comparisons whose results are displayed in

Table **(3.8)** along with the Tukey's simultaneous 95% confidence intervals (CIs) plot provided in Figure **(3.9)**.

Table 3.8. Results for Simultaneous Comparisons of Mean Performances of Heuristics on Average-R in Minitab.

```
Means
Method    N     Mean    StDev        95% CI
EDGASA  100  0,76419  0,06279  (0,74366; 0,78472)
GA      100   0,5767   0,1477  ( 0,5561;  0,5972)
SA      100  0,68381  0,08310  (0,66328; 0,70434)
Pooled StDev = 0,104324

Tukey Pairwise Comparisons

Grouping Information Using the Tukey Method and 95% Confidence

Method    N     Mean  Grouping
EDGASA  100  0,76419  A
SA      100  0,68381    B
GA      100   0,5767      C

Means that do not share a letter are significantly different.


Tukey Simultaneous Tests for Differences of Means

Difference   Difference       SE of                                 Adjusted
of Levels      of Means  Difference        95% CI         T-Value    P-Value
GA - EDGASA     -0,1875      0,0148  (-0,2220; -0,1530)    -12,71      0,000
SA - EDGASA     -0,0804      0,0148  (-0,1149; -0,0459)     -5,45      0,000
SA - GA          0,1071      0,0148  ( 0,0726;  0,1417)      7,26      0,000

Individual confidence level = 98,01%
```

It is seen from Table **(3.8)** that all of the pairwise 95% Confidence Intervals constructed on mean performance of each heuristics do not overlap each other. This implies that the differences between mean performances of algorithms are statistically significant which is visually demonstrated in Figure **(3.9)** as none of the CIs about pairwise differences contains zero. The Tukey Pairwise Comparisons used for grouping by Minitab also indicates that each of the heuristics mean solution quality in terms of Average-R are different than the others (highlighted by different letter in Table **(3.8)**) with EDGASA, SA and GA in diminishing order.

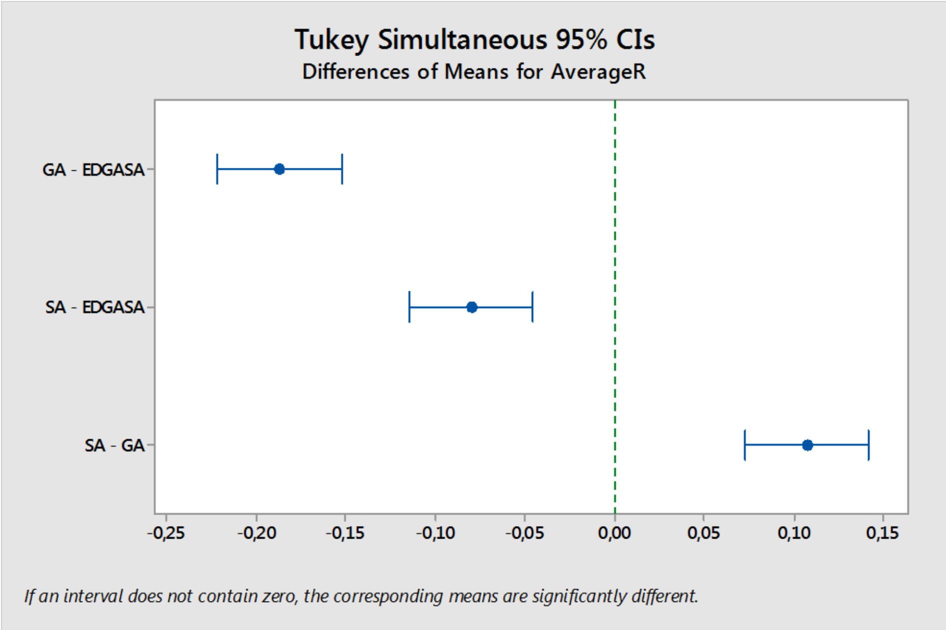


Figure 3.9. Tukey's Simultaneous 95% CIs Plot for Pairwise Differences.

Note that as it is seen from Figure **(3.9)**, none of the CIs about the pairwise difference of mean performance on average reliability provided by each of the algorithm contains zero, hence, the difference of mean performances are significant. Finally, reliability is a measure of probability and cannot exceed 1 or go below 0. Hence, natural bounds on the average performance of each of the algorithms are given through Equation **(3.13)**:

$$1 \geq \mu_R^{EDGASA} > \mu_R^{SA} > \mu_R^{GA} \geq 0 \qquad \textbf{(3.13)}$$

Further, being a coherent system, a maintenance action taken will not make the system to have inferior reliability than before, hence the do-nothing option provides a lower bound while perfect maintenance with upgraded technology on each link, although it may be infeasible in terms of cost and time, provides an absolute upper bound on the system. As exactly the same distributions are used for the failure parameters as those provided by Miman (2008); the lower and upper bounds proposed by him can also be adopted to this study such that:

$$0.056 \le \mu_R^{GA} < \mu_R^{SA} < \mu_R^{EDGASA} \le \; 0.939 \qquad \textbf{(3.14)}$$

When a measure of the closeness to the upper bound for GA, SA and EDGASA was computed, EDGASA solutions on average are found to be within 18.61% of the best possible solution without time and budget constraints while SA solutions on average were within 27.17% of the upper bound and GA solutions on average were within 38.58 % of the upper bound. This reflects that on the average EDGASA solutions are 8.56 % closer to the upper bound for the problem compared to second best solutions of algorithms considered in this study, specifically solutions provided by SA.

Average reliability performances of heuristics are related to the defined distributions for the parameters. In order to provide greater insights into the relative performance of each of the algorithms, the heuristics can be compared to each other in terms of the number of times their objective function is superior to the ones obtained through other algorithms over the 100 randomly generated problem instances. Such analysis were conducted for average reliability of 10 solution attempts to each problem instances of total 100 experimental designs; the best solution (maximum reliability) of 10 solution attempts to each problem instances of total these 100 instances and the worst solution (minimum reliability) of 10 solution attempts to each problem instances of total these 100 instances and the results are summarized in Table **(3.9)-(3.12)** respectively.

Table 3.9. Comparison of Algorithms in ability to find a Better Solution in Terms of Average Reliability.

# of Times	Average Performances (OUT OF 100)	Proportion of Time	95% CI Half-width
SA>GA (SA is better)	99	0.99	0.019
SA>=GA (SA is not worse than)	99	0.99	0.019
EDGASA>SA (EDGASA is better)	100	1.00	-
EDGASA>=SA (EDGASA is not worse than)	100	1.00	-
EDGASA>GA (EDGASA is better)	100	1.00	-
EDGASA>=GA (EDGASA is not worse than)	100	1.00	-

Table **(3.9)** provides consistent results with Table **(3.5)-(3.8)** reflecting that that EDGASA's performance is superior to SA, which is superior to GA in their ability to converge to a better solution. In detail, out of 100 problem instances, each of which are solved 10 times by each of the heuristics, on average SA provides 99% of times better solutions than those provided by GA and 99% of times its solutions are no worse than those of GA. Moreover, on average EDGASA provides always superior solutions compared to both SA and GA, i.e., 100% of times.

Table 3.10. Comparison of Algorithms in ability to find a Better Solution in terms of Minimum Reliability.

# of Times	Worse Performances (OUT OF 100)	Proportion of Time	95% CI Half-width
SA>GA (SA is better)	99	0.99	0.019
SA>=GA (SA is not worse than)	99	0.99	0.019
EDGASA>SA (EDGASA is better)	100	1.00	-
EDGASA>=SA (EDGASA is not worse than)	100	1.00	-
EDGASA>GA (EDGASA is better)	100	1.00	-
EDGASA>=GA (EDGASA is not worse than)	100	1.00	-

Each of the 100 problem instances generated randomly was solved 10 times due to inherent randomness of the solution procedures through each of the algorithms considered and when the worst solution (minimum reliability) of these 10 attempts to each problem instance were compared in Table **(3.10)**; the same results as those obtained for average reliability are obtained. Specifically, SA performs better than GA 99% of times and it also provides no solutions worse than SA %99 of times. The hybrid heuristics developed and proposed by this study, EDGASA, provides always, i.e. 100% of times, superior solutions in terms of worst solutions it provides compared to worst solutions SA and GA provide.

Table 3.11. Comparison of Algorithms in ability to find a Better Solution in terms of Maximum Reliability.

# of Times	Best Performances (OUT OF 100)	Proportion of Time	95% CI Half-width
SA>GA (SA is better)	88	0.88	0.06
SA>=GA (SA is not worse than)	88	0.88	0.06
EDGASA>SA (EDGASA is better)	93	0.93	0.05
EDGASA>=SA (EDGASA is not worse than)	100	1.00	-
EDGASA>GA (EDGASA is better)	100	1.00	-
EDGASA>=GA (EDGASA is not worse than)	100	1.00	-

As explained earlier, each of the 100 problem instances generated randomly was solved 10 times due to inherent randomness of the solution procedures through each of the algorithms considered and when the best solution (maximum reliability) of these 10 attempts to each problem instance were compared in Table **(3.11)**; SA performs better than GA 88% of times and it also provides no solutions worse than GA 88% of times. The hybrid heuristics developed and proposed by this study, EDGASA, provides 93% of times better results in terms of best solutions it provides compared to best solutions SA provides. Note that EDGSA never provides inferior solution compared to SA in terms of maximum reliability they found. When EDGASA compared to GA in terms of best solution they can provide; EDGASA always outperforms GA, i.e., %100 of times, hence, never provides no worse solutions than GA.

Another point of interest is the amount of time a heuristic takes to find a good solution, hence the average solution time, obtained through the Mac OS X with Intel core i5, 2.5 GHz processor, for each of the heuristic techniques were analyzed and analysis results were tabulated in Table **(3.12)**.

Table 3.12. Descriptive Statistics for Heuristics in terms of Average Solution Time.

Variable	Method	N	N*	Mean	SE Mean	StDev	CoefVar	Minimum	Q1	Median
AverageTime	EDGASA	100	0	101336	8371	83706	82,60	74447	92801	93092
	GA	100	0	49,316	0,201	2,006	4,07	46,800	47,900	48,900
	SA	100	0	190,46	0,754	7,54	3,96	183,30	185,90	187,85

Variable	Method	Q3	Maximum
AverageTime	EDGASA	93284	929786
	GA	49,975	56,300
	SA	191,85	224,20

Table **(3.12)** reveals that, without a need for further hypothesis testing, the ranking of average execution time for each heuristic is given by Equation **(3.15)**.

$$\mu_T^{GA} < \mu_T^{SA} < \mu_T^{EDGASA} \tag{3.15}$$

This implies that the solution quality of SA is improved significantly through the hybrid heuristic EDGASA through the feeding GA population with SA as well, yet, with a substantial increased solution time. However, there is a potential to decrease the solution time by setting the GA parameters as well as SA parameters that are being used in EDGASA carefully, which can be a potential future research's topic.

3.4. Multi-Objective Optimization Modeling for the Sustainability of a CLN based on Utopia Distance

This section proposes a multi-objective optimization of the sustainability of a CLN based on utopia distance paradigm. In the sustainability optimization of the model for whom metaheuristics are investigated is a multi-dimensional knapsack problem developed by Miman (2008). He analyzed this system as multi-objective optimization through physical programming. Here, as an alternative to his study, a multi-objective optimization model is to be developed based on utopia distance paradigm. In this model, the overall reliability of the CLN (sustainability) is to be maximized while the cost of selective maintenance actions for the sustainability, and time required by the selective maintenance actions are to be minimized. To develop the multi-objective optimization model; first, additional notations are introduced over the notation list presented in Chapter 3.1, the mathematical formulation based on utopia distance paradigm is introduced. The solution of the model presented in this section is not handled, but, is left for further studies.

3.4.1. Additional Notation List

x^{cur} : current solutions in terms of decision variables for selective maintenance actions for each base-*n* tubules

R^{cur} : reliability of the CLN based on x^{cur}

C^{cur} : cost of the selective maintenance actions performed based on x^{cur}

T^{cur} : time required for the selective maintenance actions performed based on x^{cur}

P^{cur} : current performance vector for the sustainability of the CLN-3 tubules

$P^{cur} = (R^{cur}, C^{cur}, T^{cur})$

U : utopia point, the ideal performance of the CLN, based on performances

$U = (1, 0, 0)$

d : The weighted distance between P^{cur} and U to be minimized

α : weight for R^{cur} in d

β : weight for C^{cur} in d

γ : weight for T^{cur} in d

3.4.2. Multi-Objective Model Developed

The multi-objective model to be developed has three dimensions; specifically the reliability of the CLN, the total cost of maintenance actions performed and the total time required to perform the maintenance actions. Each of these dimensions can be displaced in the performance space based on the current solution as illustrated in Figure **(3.10)**. Ideally, maximum reliability is desired to be achieved with a minimal cost and time. This implies that, in terms of utopia distance paradigm, the ideal performance based on the maintenance actions pursued is achieving reliability of 1 with zero cost and zero time. This constitutes to the utopic point to arrive. The model proposed by this study tries to minimize the weighted distance, as shown in Figure **(3.10)**, between current performances and utopic point based on the model displayed in Exhibit **(3.4)** with the decision variables as those in Exhibit **(3.3)**.

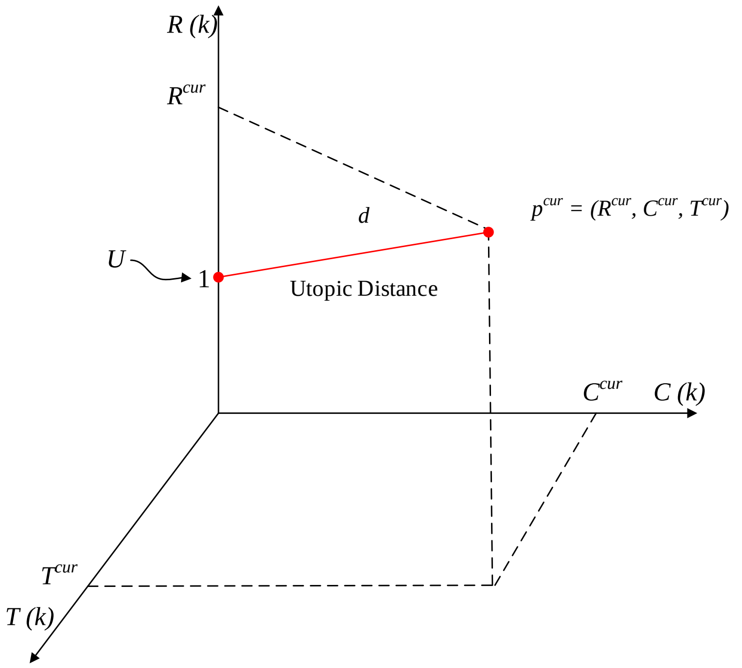


Figure 3.10. Performance Space for Utopia Distance Model.

$$
\begin{aligned}
&Min \quad d=\sqrt{\alpha(R^{cur}-1)^2+\beta(C^{cur}-0)^2+\gamma(T^{cur}-0)^2} \\
&s.t. \\
&\qquad \alpha,\beta,\gamma \geq 0 \\
&\qquad \alpha+\beta+\gamma=1
\end{aligned}
$$

Exhibit 3.4. Multi-Objective Optimization Model based on Utopia Distance, (*P7*).

The algorithm for the model presented in Exhibit **(3.4)** is displaced in Figure **(3.11)**.

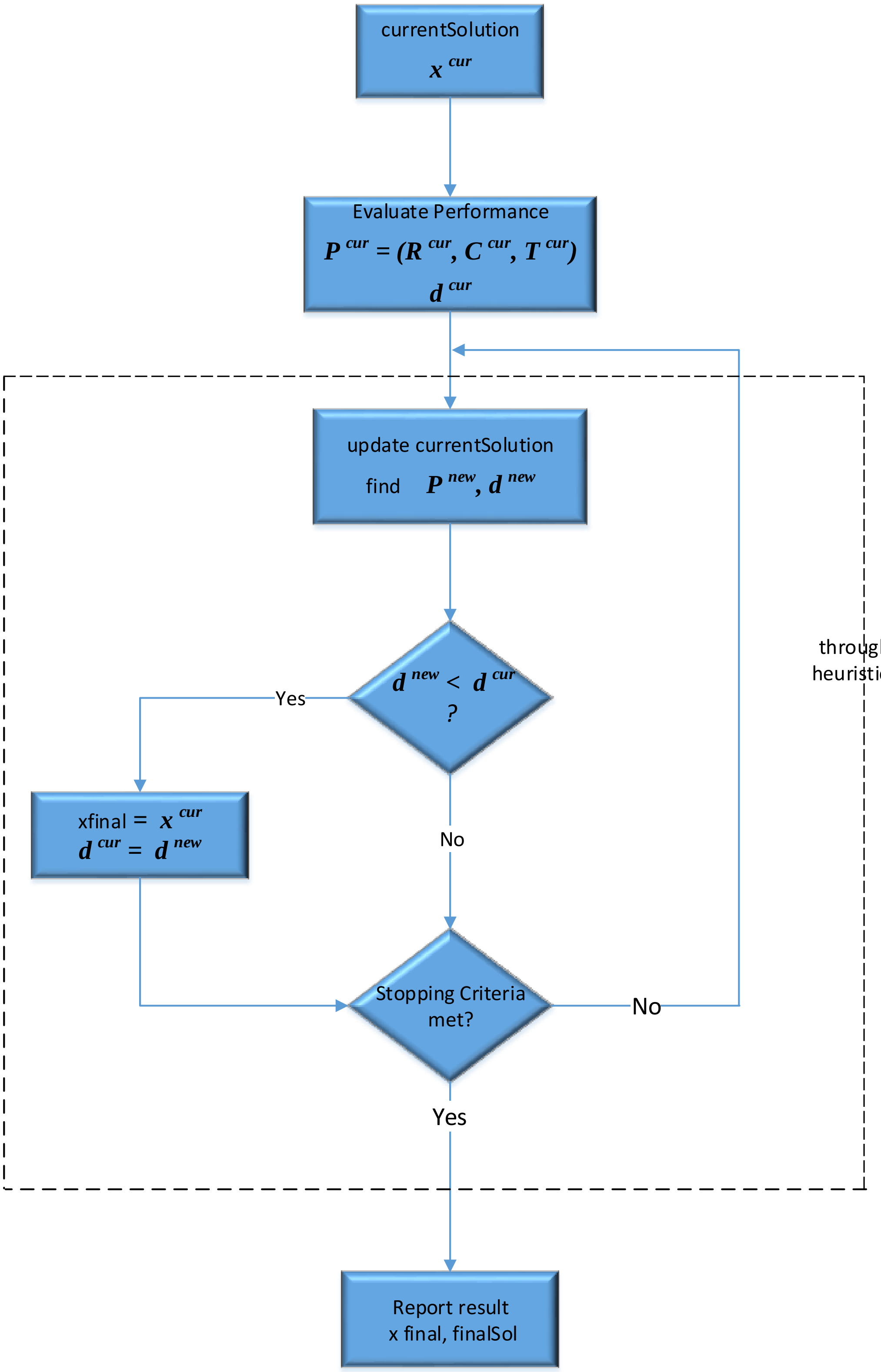


Figure 3.11. Solution Steps for Multi-Objective Optimization Model.

Note that in the step of updating current solution and finding d^{new}, metaheuristics such as EDGASA can be used as the model presented here is a also non-linear, non-separable and non-convex mathematical program. For the model proposed in Exhibit **(3.4)**, one can foresee that the selection of the weights for each performance dimension is one of the areas that needs further investigation. Some of the modeling issues developed and proposed in this section was presented by Dağ and Miman (2014).

4

CONCLUSION AND DISCUSSION

This research effort can be regarded as a building stone over the study conducted by Miman (2008) in terms of the investigation of heuristic solutions to the sustainability of contingency logistics networks (CLNs) and multi-objective optimization modeling for the sustainability of CLNs. The reliability of CLNs was explicitly modeled first time from the perspective of mission success by Thomas (2004) where he modeled a logistics network's ability to respond to a contingency in which that response is defined to be the proper set of operations for a mission or set of missions to recover from the contingency event. Miman (2008) extended preliminary study of Thomas (2004) in several ways. Among one of them, there is a sustainability model proposed which is based on a multi-action selective maintenance model for the links in the logistics network. His model enables contingency logistics network planners to identify the most proper maintenance actions for the links from a central distribution center to the operational bases (nodes) so that the logistic network's ability to respond to the contingency; i.e. the probability of success of contingency mission, is maximized subject to budget and time availability constraints for maintenance actions on the links. Miman (2008) investigates metaheuristics, particularly traditional metaheuristics genetic algorithm (GA) and tabu search (TS) and proposes multi-level ruin and rebuid heuristic (MULRR) and genetic algorithm fed by tabu search (GAFTS) he developed as solution procedures for the non-linear, non-separable and non-convex two-dimensional knapsack sustainability model. He concludes that memetic heuristic GAFTS he developed outperforms other heuristics in terms of solution quality at an expense of solution time required. In addition to the sustainability model as a two dimensional knapsack problem, he proposes the use of physical programming (PP) as a multi-objective optimization technique for the sustainability of the CLN.

This research effort continues the investigation of heuristics approaches for the sustainability model developed by Miman (2008). At the beginning, an extensive literature review is provided, including the recent literature on contingency logistics networks, heuristics solutions and multi-objective optimization techniques in Chapter 2. Next, traditional metaheuristics genetic algorithm (GA), simulated annealing (SA), a hybrid heuristic based on genetic algorithm, and simulated annealing, EDGASA, developed are analyzed and compared to each other in terms of solution quality (average reliability, best reliability, worst reliability they can find) and solution time they require in Chapter 3 of the study. The performance of the hybrid heuristic, EDGASA, is found to be superior to pure GA and pure SA in terms of solution quality at an expense of substantially increased solution time required to reach the solution.

This research also contributes to multi-objective optimization modeling paradigm for the sustainability of the CLN providing a multi-objective optimization model for the sustainability based on utopia distance. This provides greater insights to the network planners in terms of modeling.

This study can trigger a set of further studies. First of all, there is research potential on tuning the parameters on GA and SA that are used in EDGASA to improve further not only the solution quality but the solution time of EDGASA as well. Later, the performance of EDGASA developed here and GAFTS developed by Miman (2008) can be compared. Other heuristics such as ant colony optimization (ACO) and water cycle algorithm (WCA) can be investigated in search of the most robust, effective and efficient solution approach to the sustainability model developed by Miman (2008).

Further, the EDGASA can be applied to the general selective maintenance problems and other integer programs. Moreover, the sustainability model can be extended by including stock allocations for the nodes (bases) along with the selective maintenance for the links in the network. Hence a set of heuristics techniques can be investigated in search of the most robust, effective and efficient solution approach to the resulting model.

Finally, the multi-objective optimization model for the sustainability based on utopia distance developed in this study can be solved through the set of heuristics such as EDGASA, GAFTS, GA, TS, ACO, WCA and so on. It is also seen that there are available potential in the literature for the application of the multi-objective optimization modeling approach proposed in this study to other multi-criteria decision making environments.

REFERENCES

Analytis, P. P., Kothiyal, A. and Katsikopoulos, K. V. (2014). Multi-attribute utility models as cognitive search engines. *Judgment and Decision Making*, 9:403-419. 34.

Archibald, T. W., Sassen, S. A. and Thomas, L. C. (1997). An optimal policy for a two depot inventory problem with stock transfer. *Management Science*, 43:173–183.

Barbarosoglu, G., Ozdamar, L., and Cevik, A. (2002). An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. *European Journal of Operational Research*, 140, 118-133.

Balezentis, T. and Shouzhen, Z. (2013). Group multi-criteria decision making based upon interval-valued fuzzy numbers: An extension of the MULTIMOORA method. *Expert Systems with Applications*, 40, 543-550.

Barlow, R. E., and Proschan, F. (1975). Statistical Theory of Reliability and Life Testing Probability Models, New York: Holt, Rinehart, and Winston.

Belton, V. and Stewart, T. (2002). Multiple Criteria Decision Analysis - An integrated approach. Boston, Kluwer Academic Press.

Bersini H. and Varela F. J. (1990). Hints for adaptive problem solving gleaned from immune networks, Parallel Problem Solving from Nature, PPSW1, Dortmund, FRG.

Bevilacqua, M., Ciarapica, F. and Giacchetta, G. (2006). A fuzzy-QFD approach to supplier selection. *Journal of Purchasing and Supply Management*, 14-27.

Birnbaum, D. (2000). Birnbaum's Global Guide to Winning the Great Garment War, Third Horizon Press, Hong Kong.

Birnbaum, L. W. (1969). "On the importance of different components in a multi-component system" in Multivariate Analysis II, Krisnainah, Ed: Academic Press, 1969.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Comput. Surv.*, 35, 268-308.

Bowersox, D. J., Closs, D. J. and Stank, T. P. (1999). 21st Century Logistics: Making Supply Chain Integration a Reality, Council of Logistics Management, Oak Brook, IL.

Bowersox, D. J. and LaHowchic, N. J. (2008). Start Pulling Your Chain – Leading Responsive Supply Chain Transformation, OGI Enterprises, Pt. Saint Lucie, FL.

Brauers, W. K. M. (2004). Optimization methods for a stakeholder society. A revolution in economic thinking by multi-objective optimization: Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Boston, U.S.A.

Brauers, W. K. M., and Zavadskas, E. K. (2010). Project management by MULTIMOORA as an instrument for transition economies. *Technological and Economic Development of Economy*, 16 (1):5–24.

Bretthauer, K. M. and Shetty, B. (2002). The Nonlinear Knapsack Problem– Algorithms and Applications. *European Journal of Operational Research*, Vol. 138 (3):459-472.

Carmon, Z., Shanthikumar, J. G., and Carmon T. F. (1994). A psychological perspective on service segmentation: the significance of accounting for consumers perception of Waiting and Service. *Management Science*, 41:1806–1815.

Cassady, C. R. and Nachlas, J. A. (1998). Preventive Maintenance: The Next Frontier in Industrial Productivity Growth, *Industrial Engineering '98 Solutions Conference*, pp. 254-260.

Cassady, C. R., Pohl, E. A. and Murdock, W. P. (2001). Selective Maintenance Modeling for Industrial Systems. *Journal of Quality in Maintenance Engineering*, **7**(2):104-117.

Cassady, C. R. Pohl, E. A. and Murock, W. P. (2002). Optimizing Maintenance Resource Levels. *Proceedings of the Eleventh Industrial Engineering Research Conference*.

Cassady, C. R., Pohl, E. A. and Jin, S. (2004). Managing Availability Improvement Efforts with Importance Measures and Optimization. *IMA Journal of Management Mathematics*, Vol.15, No. 2, pp. 161-174.

Charnes, A., Cooper W. W. (1961). Management models and industrial applications of linear programming. Wiley:New York.

Charnes, A., Cooper W. W. and Ferguson, R. (1955). Optimal estimation of executive compensation by linear programming. *Management Science*, 1:138-151.

Chow, G., Heaver, T. D. and Henriksson, L. E. (1995). Strategy, structure and performance: a framework for logistics research. *Logistics and Transportation Review*, 31(4):285-308.

Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based Multi-objective optimization techniques. *Knowledge and Information Systems*, 1:269-308.

Coit, D. W. and Jin, T. (2001). Prioritizing System-reliability Prediction Improvements. *IEEE Transactions on Reliability*, Vol. 50, No.1, pp. 17-25.

Coit D. W. and Smith, A. E. (1996a). Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach. *Computers and Operations Research*, 23(6):515–526.

Coit D. W. and Smith, A. E. (1996b). Reliability optimization of series parallel systems using genetic algorithm. *IEEE Transactions on Reliability*, 45(2):254–260.

Coit D. W. (2003). Maximization of system reliability with a choice of redundancy strategies. *IIE Transactions*, 35:535-544.

Dağ, E. and Miman, M. (2014). Beklenmedik Durumlar Lojistiğinin Optimizasyonunda Ütopya Uzaklık Metodu. *III. Ulusal Lojistik ve Tedarik Zinciri Kongresi*, pp680-688, May 2014, Trabzon.

Dağ, E. and Miman, M. (2015). Multi-objective Optimization of Contingency Logistics Networks with Distorted Risks. *International Conference on Value Chain Sustainability*, (*ICOVACS*), pp487-492, March 2015, İstanbul.

Dantzig, G. B. (1963). Linear Programming and Extensions, Princeton University Press.

Das, C. (1975). Supply and redistribution rules for two-location inventory systems: one period analysis. *Management Science*, 21:765–776.

Deb, K. (2001). Multi-objective optimization. *Multi-objective optimization using evolutionary algorithms*, 13-46.

DeCroix, G. and Arreola-Risa, A. (1998). Optimal Production and Inventory Policy for Multiple Products under Resource Constraints. *Management Science*, 44(7):95-96.

Desrosiers, J., Dumas, Y., Solomon, M. M. and Soumis, F. (1990). Time constrained routing and scheduling. In: M.E. Ball, T.L. Magnanti, C. Monna and G.L. Nemhauser, Editors,Handbooks in Operations Research and Management Science: Networks, North-Holland, Amsterdam, Ch. 11.

Dorigo, M. (1992). Optimization, Learning and Natural Algorithms (in Italian). *Ph.D. thesis,* Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dzyabura, D. (2014). Preference discovery in product search. New York University working paper.

Eskandar, H., Sadollah, A., Bahreininejad, A. and Hamdi, M. (2012). Water cycle algorithm-A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures*, 110-111:151–166.

Farzamnia, E. and Babolghani, M. D. (2014). Group Decision-Making Process for Supplier Selection Using Multimoora Technique under Fuzzy Environment. *Kuwait Chaper pf Arabian Journal of Business and Management Review*, 3(11):203-218.

Farmer J. D., Packard N. and Perelson A. (1986). The immune system, adaptation and machine learning. *Physica D*, 2:187-204.

Fisher, M. L. (1997). What is the right supply chain for your product. *Harvard Business Review*, 75(2):105-116.

Geem, Z. W., Kim, J. H. and Loganathan, G. V. (2001). A new heuristic optimization: Harmony search, *Simulation*, 76:60–68.

Gigerenzer, G., Hertwig, R., van der Broek, E., Fasolo, B., and Katsikopoulos, K. V. (2005). A 30% chance of rain tomorrow: How does the public understand weather forecasts. *Risk Analysis*, 25:623-629.

Gilmore, P. C. and Gomory, R. E. (1961). A linear programming approach to the cutting stock problem. *Operations Research*, 9 (6):849–859.

Glover, F. (1986). Future paths for integer programming and links to artficial Intelligence. *Computers and Operations Research*,13:533-549.

Glover F. and Laguna M. (1997). Tabu Search, Kluwer, Boston.

Glover F. and Kochenberger G. A. (2003). Handbook of Metaheuristics, Springer.

Gross, D. (1963). Centralized inventory control in multi-location systems. In: Scarf HE, Gilford DM and Shelly MW (eds). Multistage Inventory Models and Techniques. Stanford University Press, Stanford.

Goldsman. D. and Nelson, B. L. (1998). Comparing Systems via Simulation invited chapter in *Handbook of Simulation*, edited by Jerry Banks, John Willey & Sons, Inc.

Gupta, M., Bhattacharjee, D. (2012). Two Weighted Fuzzy Goal Programming Methods to Solve Multi objective Goal Programming Problem. *Journal of Applied Mathematics*, Volume 2012, Article ID 796028, pp. 20-40.

Ha, C. and Kuo, W. (2006). Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems. *European Journal of Operational Research*, 171(1):24-38.

Haddad O. B., Afshar A., Marino M. A. (2006). Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization. *Water Resources Management*, 20:661-680.

Hannan, E. L. (1981). On Fuzzy Goal Programming. *Decision Sciences*, 12:522–531

Herer, Y. T., Tzur, M. and Yücesan, E. (2006). The multi-location transshipment problem. *IIE Transactions*, 38:185-200.

Ho, Y. C., M. A. Eyler, and T. T. Chien. (1979). A gradient technique for general buffer storage design in a serial production line. *International Journal of Production Research*, 17:557-580.

Holland, J. (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.

Hughes, E. J. (2003). Multiple single objective pareto sampling. Congress on Evolutionary Computation, pages 2678–2684, Canberra, Australia, 8-12 December 2003, *IEEE*.

Karaboga D. (2005). An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Erciyes University, Turkey.

Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization, in: Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942–1948.

Kevin Y. K. NG and Sancho, N. G. F. (2001). A hybrid 'dynamic programming/depth-first search' algorithm, with an application to redundancy allocation. *IIE Transactions*, 33 (12):1047-1058.

Kickert, W. J. (1978). Fuzzy Theories on Decision-Making: Frontiers in systems Research. Leiden, The Netherlands: Martinus Nijhoff.

Kim, I.Y., De Weck O. (2012). Adaptive Weighted Sum Method for Multiobjective Optimization, Submitted to Structural and Multidisciplinary Optimization.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220 (4598), 671–680.

Kiwus, C. H. (1990). Contingency Construction Planning in the U.S. Armed Services, Including the Extent of Modular Construction. Master's Thesis for Georgia Institute of Technology Atlanta School of Civil Engineering.

Kodialam, M. S. and Luss, H. (1998). Algorithms for separable nonlinear resource allocation problems. *Operations Research*, 46 (2):272–284.

Krishnan, K. S. and Rao, V. R. K. (1965). Inventory control in N warehouses. *Journal of Industrial Engineering*, 16:212–215.

Kukreja, A. and Schmidt, C. P. (2005). A model for lumpy demand parts in a multi-location inventory system with transshipments. *Computers and Operations Research*, 32:2059–2075.

Lambert, D. M., James R. Stock, and Ellram, L. M. (1998). Fundamentals of Logistics Management, Boston, MA: Irwin/McGraw-Hill, Chapter 14.

Lee, Y. H., Jung, J. W. and Jeon, Y.S. (2007). An effective lateral transshipment policy to improve service level in the supply chain. *International Journal of Production Economics*, 106:115–126.

Lee. S. J. and Siau, K. (2001). A review of data mining techniques. *Industrial Management and Data System*, 101(1):41-46.

Liao Z. and Rittscher J. (2007). A multi-objective supplier selection model under stochastic demand conditions. *International Journal of Production Economics*, 105(1)150–159.

Luan, S., Schooler, L. J. and Gigerenzer, G. (2014). From perception to preference and on to inference: An approach-avoidance analysis of thresholds. *Psychological Review*, 121:501-525.

Mathur, K., Salkin, H. M., Mohanty, B. B. (1986). A note on a general non-linear knapsack problem. *Operations Research Letters*, 5(2):79–81.

McLeish, D. L., and Reesor, R. M. (2003). Risk, entropy, and the transformation of distributions. *North American Actuarial Journal*, 7:128-144.

Messac, A., (1996). Physical Programming: Effective Optimization for Computational Design, *AIAA Journal*, 34(1):149-158.

Messac, A., Batayneh, W. M., and Ismail-Yahaya, A. (2002). Production Planning Optimization with Physical Programming, *Engineering Optimization*, 34(4):323-340.

Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, 63:81-97.

Miman, M. (2008). Modeling and Analysis of the Reliability of Contingency Logistic Networks: A Multi-dimensional Knapsack Approach, Dissertation for PhD in Industrial Engineering, University of Arkansas

Miman M. and Pohl A. E. (2006). Uncertainty Assessment for Availability: Importance Measure. The Proceedings of the 2006 Reliability and Maintainability Symposium.

Miman, M. and Pohl, E. A. (2008). Modelling and analysis of risk and reliability for a contingency logistics supply chain. *Journal of Risk and Reliability*, 222(4):477-494.

Miman, M. and Pohl, E. A. (2012). Multi-objective optimisation of a contingency logistics network through physical programming. *International Journal of Collaborative Enterprise*, 3(1):1-17.

Miman, M. and Pohl, E. A. (2013). Sustainability Optimization for a Contingency Logistic Network: A Multi-dimentional Knapsack Approach, (Submitted to *European Journal of Operational Research*).

Minoux, M. (1987). A class of combinatorial optimization problems with polynomially solvable large scale set-covering/partitioning relaxations, RAIRO. *Operations Research*, 21(2):105-136.

Moritz, R. F. and Southwick, E. E. (1992). Bees as superorganisms, Springer.
Moscato, P. (1989). On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program (report 826).
Nakrani, S. and Tovey, C. (2004). On honey bees and dynamic server allocation in Internet hosting centers. *Adaptive Behavior*, 12:223-240.
Needham, P. M. and Evers, P. T. (1998). The influence of individual cost factors on the use of emergency transshipments. *Transportation Research E,* 34:149–160.
Offut, M. E., Kharoufeh, J. P. and Deckro, R. F. (2006). Distorted Risk Measures with Application to Military Capability Shortfalls, *Military Operation Research*, 11(4):25-39.
Ohtagaki, H., Nakagawa, Y. and Narisha, H. (2000). Smart Greedy Procedure for Solving a Multidimensional Nonlinear Knapsack Class of Reliability Optimization Problems. *Mathematical and Computer Modeling*, 31(10-12):283-288.
Parlar, M. (1997). Continuous-review inventory problem with random supply interruptions. *European Journal of Operational Research*, 99:366–385.
Parlar, M., and D. Berkin. (1991). Future supply uncertainty in EOQ models. *Naval Research Logistics*, 38:107–121.
Passino K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control System Magazine*, pp. 52-67.
Pham D. T., Ghanbarzadeh A., Koc E., Otri S., Rahim S. and Zaidi M. (2005). The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK.
Pohl, E. A., Cassady, C. R. and Kwinn, M. J. (2005). A Selective Maintenance Model for Serial Manufacturing Systems Involving Multiple Maintenance Actions. *Proceedings of Proceedings of Industrial Engineering Research Conference*.
Pohl, E. A., Miman, M., Xiaoping, L., Unlu, Y. (2007). Multi Action Selective Maintenance Problem: Exact and Meta Heuristic Solution Approaches. (Working paper to submit Journal of Quality in Maintenance Engineering).
Price K., Storn R. and Lampinen J., (2005). Differential Evolution: A Practical Approach to Global Optimization, Springer.
Purshouse R. C. (2003). Evolutionary many-objective optimisation: An exploratory analysis. Congress on Evolutionary Computation (CEC 2003), volume 3, pages 2066–2073, Canberra, Australia, 8–12 December 2003, *IEEE*.
Rajagopalan, R. and Cassady, C. R. (2006). An improved selective maintenance solution approach, *Journal of Quality in Maintenance Engineering*, Vol. 12, No. 2, pp. 172-185.
Rice, W. F., Cassady, C. R. and Nachlas, J. A. (1998). Optimal Maintenance Plans under Limited Maintenance Time. *Proceedings of the Seventh Industrial Engineering Research Conference*.
Robertson, D. and Ulrich, K. (1998). Planning for Product Platforms. *Sloan Management Review* (Summer), 19–31.

Rossetti, M. D. (2008). JSL: An Open-Source Object-Oriented Framework for Discrete-Event Simulation in Java. *International Journal of Simulation and Process Modeling*, vol. 4., no. 1, pp. 69-87,

Rubinstein R. Y. (1997). Optimization of computer simulation models with rare Events. *European Journal of Operations Research*, 99:89-112.

Saaty, T. L. (1980). The analytic hierarchy process: planning, priority setting, resource allocation. Mcgraw-Hill.

Saaty, T. L. (1990). How to make a decision: The analytical hierarchy process. *European Journal of Operational Research*, 48:9-26.

Shannon, C. E. and Weaver, W. (1947). The mathematical theory of communication . Urbana: University of Illinois Press.

Snyder, L. V. and Shen, Z. J. M. (2006). Disruptions in Multi-Echelon Supply Chains: A Simulation Study.

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

Talbi, E. G. (2009). Metaheuristics: From Design to Implementation, John Wiley & Sons.

Tan, K. C. (2001). A framework of supply chain management literature. *European Journal of Purchasing and Supply Management*, 7:39–48.

Thomas, M. U. (2004). Assessing Reliability of Contingency Logistic network. *Military Operation Research*, Vol. 9, No. 1, pp. 33-41.

Thomas, M. U. (2008). Inventory models for contingency operations in Handbook of Military Industrial Engineering, A.B. Badiru and M.U. Thomas(Eds), 21-1-21-10. Taylor & Francis, Baco Raton, FL.

Thomas, M. U., Deshpande, V., and Iyer, A. V. (2002). Contingency Logistics Systems-A Taxonomy and Application Modeling Framework for Contingency Logistics Planning. Tech. Memo. No. 02-03, School of Industrial Engineering, Purdue University, West Lafayette, Indiana.

Tian, Z., Zuo, M. J. and Huang, H. (2008). Reliability-redundancy allocation for multi-state series-parallel systems. *IEEE Transactions on Reliability*, Vol. 57, No.2, pp. 303-310.

Triantaphyllou, E. and Mann, S. H. (1995). Using the analytic hierarchy process for decision making in engineering applications: some challenges. *International Journal of Industrial Engineering: Applications and Practice*, *2*(1):35-44.

Vance, P. H., Barnhart, C., Johnson, E. L. and Nemhauser, G. L. (1994). Solving binary cutting stock problems by column generation and branch-and-bound. *Computational Optimization and Applications*, Vol. 3, No. 2, pp. 111-130.

Vanderbeck, F. and Wolsey, L. A. (1996). An exact algorithm for IP column generation. *Operations Research Letters*, Vol. 19, No. 4, pp. 151-159.

Vollmann, T. E. and Cordon, C. (1998). Building successful customer-supplier Alliances. *Long Range Planning*, 31(5):684-94.

Voss, S. (2001). Meta-heuristics: the state of the art, in: Local Search for Planning and Scheduling (Ed. A. Nareyek), LNAI 2148, pp. 1-23 (2001).

Weitzman M. L. (1979). Optimal Search for the Best Alternative. *Econometrica*, Vol. 47, No:3, pp. 641-654.

Wong, H, VanHoutum, G. J., Cattrysse, D. and Van Oudheusden, D. (2006). Multi-item spare parts systems with lateral transshipments and waiting time constraints. *European Journal of Operational Research*, 171:1071–1093.

Wren, D. A. (1994). The Evolution of Management Thought. 4th Edition, Wiley & Sons, New York.

Xie, M. and Shen, K. (1989). On ranking of system components with respect to different improvement actions. *Microelectrics and Reliability*, 29(2):159-164.

Yang, X. S. and Deb, S. (2009) Cuckoo search via L´evy flights, Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC), December 2009, India. IEEE Publications, USA, pp. 210-214.

Yang, X. S. (2005). Engineering optimization via nature-inspired virtual bee algorithms, IWINAC. *Lecture Notes in Computer Science*, 3562, 317-323.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization, 5th Symposium on Stochastic Algorithms, Foundation and Applications (SAGA 2009) (Eds Watanabe O. and Zeugmann T.), LNCS, 5792, pp. 169–178.

Yang, X. S. (2010). Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley & Sons.

Yang, X. S., and Deb, S. (2010). Engineering optimization by cuckoo search. *Int. J. Math. Modelling Num. Optimisation*, 1(4):330–343.

Yang, X. S. (2011) Metaheuristic Optimization. *Scholarpedia*, 6(8):11472.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8:338-353.

Zeleny, M. (1982). Multiple criteria decision making. New York: McGraw-Hill.

Zimmermann, H. J. (1987). Fuzzy Sets, Decision Making, and Experts Systems. Boston: Kluwer.

Zimmermann, H. J. (1991). Fuzzy Set Theory-and Its Applications, (2nd rev.ed). Boston: Kluwer.

Zia, L. and Coit, D. W. (2006). A General Column Generation Algorithm Applied to System Reliability Optimization Problems, *Proceedings of the Industrial Engineering Research Conference* (IERC), Orlando, FL, May 2006.

Url-1 <*https://en.wikipedia.org/wiki/Goal_programming*>, date retrieved 14.04.2015.

Url-2 <*http://cavern.uark.edu/~rossetti/research/research_interests/simulation/java_simulation_library_jsl/*>, date retrieved 09.08.2014.

APPENDICES

This section provides supplementary materials refereed inside the text to make this study easy to be understood.

APPENDIX 1. Packages Used for Coding Algorithms

```
package ThesisCodes;
/*This code is generated by
 @Esra DAG for her thesis
 under the supervision of
 Assistant Professor Dr. Mehmet Miman
 The Last Updated on 25.07.2015
 */

import java.io.*;
import java.util.*;
import java.lang.Math;
//  packages used from JSL
import jsl.utilities.random.DUniform;
import jsl.utilities.random.RandomIfc;
import jsl.utilities.random.Uniform;
import jsl.utilities.statistic.Statistic;
```

APPENDIX 2. Initial Variables and Parameter Used in Codes

```
static int n=16; // number of sites
// variables for experimental design parameters
static double [] Aijm= new double [n+1];
static double [] nuijm_new= new double [n+1];
static double [] betaijm_new= new double [n+1];
static double [] Imijm_new= new double [n+1];
static double [] Xijm= new double [n+1];
static double [] Wijm= new double [n+1];
static double [] Wtrial= {0.0,0.0,1.0,0.0,0.0,1.0,0.0};
static double [] Vtrial= {0.0,1.0,0.0,0.0,1.0,0.0,0.0};
static double [] Ztrial= {0.0,0.0,0.0,1.0,0.0,0.0,1.0};
static double [] Vijm= new double [n+1];
static double [] Zijm= new double [n+1];
static double [] Rijm= new double [n+1];
static double [] Rnodes= new double [n+1];
static double [] Ri= new double [5];
static double [] Ritemp= new double [5];
static double [] CMR= new double [n+1];
static double [] CRC= new double [n+1];
static double [] CRN= new double [n+1];
static double [] CRFC= new double [n+1];
static double [] CRFN= new double [n+1];
static double [] TMR= new double [n+1];
static double [] TRC= new double [n+1];
static double [] TRN= new double [n+1];
static double [] TRFC= new double [n+1];
static double [] TRFN= new double [n+1];
```

```
// distributions to generate random experimental setups
static RandomIfc myYs= new DUniform(0,1);
static RandomIfc myBs= new Uniform(0,50);
static RandomIfc myNus= new Uniform(50,150);
static RandomIfc myBetas= new Uniform(1.1,2.1);
static RandomIfc myNumus= new Uniform(90,210);
static RandomIfc myBetamus= new Uniform(1.5,3.0);
static RandomIfc myIms= new Uniform(0.35,0.95);
static RandomIfc myRFs= new Uniform(0.05,0.20);
static RandomIfc myLs=new Uniform(75,100);
static RandomIfc mycmrs= new Uniform(20,40);
static RandomIfc mycrcs= new Uniform(50,90);
static RandomIfc mycrns= new Uniform(70,130);
static RandomIfc mycrfcs= new Uniform(60,100);
static RandomIfc mycrfns= new Uniform(90,150);
static RandomIfc mytmrs= new Uniform(30,60);
static RandomIfc mytrcs= new Uniform(20,40);
static RandomIfc mytrns= new Uniform(25,45);
static RandomIfc mytrfcs= new Uniform(30,34);
static RandomIfc mytrfns= new Uniform(35,45);
static RandomIfc myRnodes= new Uniform(0.84,0.99);
static RandomIfc myCAvailable= new Uniform(550,750);
static RandomIfc myTAvailable= new Uniform(200,380);
static int num=100; // number of experiments
static int numRep=10; //number of replications used for each setup
// Statistics Collectors for experiments
static Statistic myEDGASAs = new Statistic("EDGASAReliabilityStatistic");
static Statistic myEDGASATs = new Statistic("EDGASATimeStatistic");
```

```
// parameters for GA
static int populationsize=50;  //population
static ArrayList<int[]> mypopulation = new ArrayList<int[]>(); //for current population
static ArrayList<int[]> mynextpopulation = new ArrayList<int[]>(); //for next generation
static int [] newchromosome =new int[2*n+1]; //represent a solution
static int [] populationIndex;
static double [] populationFitness;
static int [] tempchromosome =new int[2*n+1];
static int [] parent1 =new int[2*n+1];
static int [] parent2 =new int[2*n+1];
static int [] child1 =new int[2*n+1];
static int [] child2 =new int[2*n+1];
static int [] child =new int[2*n+1];
static int [] mutant =new int[2*n+1];
static double [] FR;
static double [] CFR;
static int recursive=1;
static int [] mask =new int[2*n+1];
static int nElicit;
static int p1Index;
static int p2Index;
static int nCrossover;
static int nMutation;
static double pElicit=0.04;//eliciting rate
static double pCrossover=0.56;//crossing over rate
static double pMutation=0.40;//mutation rate
static int endElicitIndex;
static int beginCrossoverIndex;
static int endCrossoverIndex;
static int beginMutationIndex;
static int endMutationIndex;
static int m1Index;
static int m2Index;
static int m1;
static int m2;
static int generationnumber=250; //maximum generation number
```

```
// parameters for SA
static double deltasol=0;// improvement in solution
static double t=0; // for current temperature
static double tinitial=1000; //initial temperature
static double tfinal=0; //for final temperature
static int n1=500; //number of times solutions tried at each t
static int m=100; // to find the final temperature
static int v1=0; // to compute next t
static double alpha=0.8;//for another way of setting next t
```

```java
// to compute the solution and fitness

// do nothing solution
static int [] myinitsol ={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

static int [] mytempsol= new int [1+2*n] ;
static int [] mycursol= new int [1+2*n] ;
static int [] myfinalsol= new int [1+2*n] ;
static int [] zero={0,0,0};
static int [] w={0,0,1};
static int [] v={0,1,0};
static int [] z={0,1,1};
static int [] temp=new int [3];
static double []left =new double[3]; // left [1] holds total cost, left [2] holds total time
static double []right =new double[3]; // right [1] holds available budget, right[2] holds available time
static boolean feasible [] = new boolean [3];


static double penalty=10000;
static double totalpenalty=0;
static double R=0;
static double C=0;
static double T=0;

static double finalsol=0;
static double cursol=0;
static double tempsol=0;
```

APPENDIX 3. makeExperimentalSetup Function

```
public static void makeExperiementSetup(){
    //reset statistics collectors
    myEDGASAs.reset();
    myEDGASATs.reset();

    //reset experimental parameters' generators
    myYs.resetNextSubstream();
    myBs.resetNextSubstream();
    myNus.resetNextSubstream();
    myBetas.resetNextSubstream();
    myNumus.resetNextSubstream();
    myBetamus.resetNextSubstream();
    myIms.resetNextSubstream();
    myRFs.resetNextSubstream();
    myLs.resetNextSubstream();
    mycmrs.resetNextSubstream();
    mycrcs.resetNextSubstream();
    mycrns.resetNextSubstream();
    mycrfcs.resetNextSubstream();
    mycrfns.resetNextSubstream();
    mytmrs.resetNextSubstream();
    mytrcs.resetNextSubstream();
    mytrns.resetNextSubstream();
    mytrfcs.resetNextSubstream();
    mytrfns.resetNextSubstream();
    myRnodes.resetNextSubstream();
    myCAvailable.resetNextSubstream();
    myTAvailable.resetNextSubstream();
```

```
// set parameters of each base for a given experimental design
        for(int l=1;l<=n;l++){
             Yijm[l] =myYs.getValue();
             Bijm[l] =myBs.getValue();
             nuijm[l] =myNus.getValue();
             betaijm[l] =myBetas.getValue();
             nuijmu[l] =myNumus.getValue();
             betaijmu[l] =myBetamus.getValue();
             Imijm[l] =myIms.getValue();
             RFijm[l] =myRFs.getValue();
             Lijm[l] =myLs.getValue();
             cmrij[l] =mycmrs.getValue();
             crcij[l] =mycrcs.getValue();
             crnij[l] =mycrns.getValue();
             crfcij[l] =mycrfcs.getValue();
             crfnij[l] =mycrfns.getValue();
             tmrij[l] =mytmrs.getValue();
             trcij[l] =mytrcs.getValue();;
             trnij[l] =mytrns.getValue();
             trfcij[l] =mytrfcs.getValue();
             trfnij[l] =mytrfns.getValue();
             Rnodes[l]=myRnodes.getValue();
             CAvailable=myCAvailable.getValue();
             TAvailable=myTAvailable.getValue();
        }
```

APPENDIX 4. Common Functions used in all of the Algorithms

```
public static double encomputeobj(int x[]){
    // evaluate the fitness of the solution "x"
    R=0;
    tempsol=0;
    // convert the 2-bit solution encoding
    //into decision variable values
    encode(x);
    updateparameters(); // update parameters according to decision variables
    computeleft();  // compute cost and time required
    computeobj(); // compute reliability
    totalpenalty=0;
    for(int k=1;k<=2;k++){
        if(left[k]>right[k]){
            feasible[k]=false; //no need
            // if constraints are violated
            // compute the penalty
            totalpenalty=totalpenalty+penalty*(left[k]-right[k]);
        }
    }
    tempsol = R-totalpenalty;
    return tempsol;
}
```

```
public static void updateparameters() {
    for(int i=1;i<=n;i++){ // for each node
        // update parameters required to compute the reliability
        // based on decision variables
        Imijm_new[i]=(Imijm[i]+RFijm[i])*Wijm[i]*(1-Vijm[i])*(1-Zijm[i]);
        Aijm[i]=Bijm[i]-Bijm[i]*Vijm[i]-Bijm[i]*Zijm[i]-(1-Imijm_new[i])*Bijm[i]*Wijm[i];
        Xijm[i]=Yijm[i]+Wijm[i]*(1-Yijm[i])+Vijm[i]*(1-Yijm[i])+Zijm[i]*(1-Yijm[i]);
        betaijm_new[i]=betaijm[i]*(1-Zijm[i])+betaijmu[i]*Zijm[i];
        nuijm_new[i]=nuijm[i]*(1-Zijm[i])+nuijmu[i]*Zijm[i];
    }
}
```

```
public static void computeleft(){
    // initialize cost and time required to be zero
    left= new double [3];
    // foe each node
    for(int h=1;h<=n;h++){
        // compute cost of selected maintenance alternative
        CMR[h]=cmrij[h]*Wijm[h];
        left[1]=left[1]+CMR[h];
        CRC[h]=crcij[h]*Vijm[h]*(1-Yijm[h]);
        left[1]=left[1]+CRC[h];
        CRN[h]=crnij[h]*Zijm[h]*(1-Yijm[h]);
        left[1]=left[1]+CRN[h];
        CRFC[h]=crfcij[h]*Vijm[h]*Yijm[h];
        left[1]=left[1]+CRFC[h];
        CRFN[h]=crfnij[h]*Zijm[h]*Yijm[h];
        left[1]=left[1]+CRFN[h]; // compute total cost

        // compute time of selected maintenance alternative
        TMR[h]=tmrij[h]*Wijm[h];
        left[2]=left[2]+TMR[h];
        TRC[h]=trcij[h]*Vijm[h]*(1-Yijm[h]);
        left[2]=left[2]+TRC[h];
        TRN[h]=trnij[h]*Zijm[h]*(1-Yijm[h]);
        left[2]=left[2]+TRN[h];
        TRFC[h]=trfcij[h]*Vijm[h]*Yijm[h];
        left[2]=left[2]+TRFC[h];
        TRFN[h]=trfnij[h]*Zijm[h]*Yijm[h];
        left[2]=left[2]+TRFN[h]; // compute total time
    }
}
```

```
public static void computeobj() {
    double Rtemp=1.0;
    // compute the reliability of each base
    for(int j=1;j<=n;j++){
        Rijm[j]=Rnodes[j]*Math.exp(-Math.pow((Lijm[j]+Aijm[j])/nuijm_new[j],
        betaijm_new[j])+Math.pow(Aijm[j]/nuijm_new[j],betaijm_new[j]))*Xijm[j];
     }

     Ri=new double [5];
     Ritemp[1]=1;
     Ritemp[2]=1;
     Ritemp[3]=1;
     Ritemp[4]=1;
     // compute the reliability of each subsystem
     for(int z=1;z<=4;z++){
         Ritemp[1]=Ritemp[1]*(1-Rijm[z]);
     }
     Ri[1]=1-Ritemp[1];

     for(int k=5;k<=8;k++){
         Ritemp[2]=Ritemp[2]*(1-Rijm[k]);
     }
     Ri[2]=1-Ritemp[2];
     for(int l=9;l<=12;l++){
         Ritemp[3]=Ritemp[3]*(1-Rijm[l]);
     }
     Ri[3]=1-Ritemp[3];

     for(int m=13;m<=16;m++){
         Ritemp[4]=Ritemp[4]*(1-Rijm[m]);
     }
     Ri[4]=1-Ritemp[4];
     // compute the reliability of the CLN
     R=Ri[1]*Ri[2]*Ri[3]*Ri[4];
}
```

APPENDIX 5. Codes of Functions for GA Algorithm

```
public static void main(String[] args) throws IOException  {
    // make "num" times experimental designs
    for (int iter=1;iter<=num;iter++){
        makeExperiementSetup(); // set experimental design parameters
        // run "numRep" times replication for a given experimental design
        for(int s=1; s<=numRep;s++){
            /*
            set initial values for start and end time
            to find a solution for a given replication
            */
            double timeStart=0;
            double timeEnd=0;
            // mark start time for the solution
            timeStart=System.currentTimeMillis();
            initialize(); // initialize GA algorithm
            // generate initial solution
            getinitialpopulation();
            // sort solution according to their fitness
            insertionSort( populationFitness,populationsize);
            gaonly(); // start GA mechanism
            // mark end time for the solution
            timeEnd=System.currentTimeMillis();
            /*
             collect the best evaluated solution found
             as the result of the current replication
             */
            myEDGASAs.collect(populationFitness[0]);
            /*
             collect the solution time it takes to find
             best evaluated solution found for the current replication
            */
            myEDGASATs.collect((timeEnd-timeStart));
       } // end of each replication
      /*
       Report statistics for each replication:
       Average-R,R-min,R-max, standard error for reliability
       Report statistics for each replication:
       Average-T,T-min,T-max, standard error for solution time
      */
      System.out.println(iter +"  "+myEDGASAs.getAverage()+"  "+myEDGASAs.getMin()+
      "  "+myEDGASAs.getMax()+"  "+myEDGASAs.getStandardError()+"  "+"***"+"  "+
      myEDGASATs.getAverage()+"  "+myEDGASATs.getMin()+"  "+myEDGASATs.getMax()+
      "  "+myEDGASATs.getStandardError());
      }// end of each experimental design
}// end of main
```

```
public static void initialize(){

    myinitsol=initialsolution();
    right[1]=CAvailable;
    right[2]=TAvailable;
    myfinalsol=myinitsol.clone();
    mycursol=myinitsol.clone();
    cursol=encomputeobj(mycursol);
    finalsol=cursol;

    // set parameters for generating populations

    populationIndex =new int[populationsize];
    populationFitness=new double[populationsize];
    FR=new double[populationsize];
    CFR=new double[populationsize];
    nElicit=(int) (populationsize*pElicit); // number solution to be elicited
    nMutation=(int) (populationsize*pMutation); // number of solutions to be mutated
    nCrossover=(int) (populationsize*pCrossover); // number of solution for crossing over
    endElicitIndex=nElicit-1;
    beginCrossoverIndex=nElicit;
    endCrossoverIndex=nElicit+nCrossover-1;
    beginMutationIndex=endCrossoverIndex+1;
    endMutationIndex=populationsize-1;
    mypopulation.clear();
    mynextpopulation.clear();

}
```

```
public static void getinitialpopulation(){
    double asol=0;

        for(int i=0;i<populationsize;i++) {
            // generate populations randomly
            newchromosome = randominitialsolution().clone();
            populationIndex[i]=i;
            asol=encomputeobj(newchromosome);
            if(asol<0)
            populationFitness[i]=1/Math.abs(asol);
            else
            populationFitness[i]=asol;
            mypopulation.add(newchromosome .clone());

        }

    }
```

```
public static void insertionSort(double numbers[], int array_size) {
  int i, j, index2;
  double index;

  for (i=1; i < array_size; i++)
  {
    index = numbers[i];
    index2 = populationIndex[i];
    j = i;

    while ((j > 0) && (numbers[j-1] < index)) {
      numbers[j] = numbers[j-1];
      populationIndex[j] = populationIndex[j-1];
      j = j - 1;
    }
    numbers[j] = index;
    populationIndex[j] = index2;
  }
}
```

```
public static void gaonly() {
        for(int i=1;i<=generationnumber;i++){
            // generate next population
            getnextpopulationonly();
            // update index
            updateIndex();
            // sort population according to fitness
            insertionSort( populationFitness,populationsize);
        }
}
```

```
public static void getnextpopulationonly(){
    //initialize population
    mynextpopulation.clear();
    elicitga(); // eliciting
    s_r_crossover(); //crossing over
    // rangecrossover2(); (An alternative crossing-over mechanism)
     mutation_bit(); //mutation
    //mutation_bit2(); (An alternative mutation mechanism)
    mypopulation=(ArrayList<int[]>) mynextpopulation.clone();
}
```

```
private static void elicitga() {

    for(int e=0;e<=endElicitIndex;e++){
        // get the solution with best solution
        tempchromosome=mypopulation.get(populationIndex[e]);
        // insert in into next generation
        mynextpopulation.add(tempchromosome.clone());
    }
}
```

```
private static void s_r_crossover() {
        boolean cdone=false;
        int ncdone =0;
        double R1;
        double R2;
        int cp;
        int range=endCrossoverIndex+1;
        while(!cdone){
            boolean []crossed=new boolean [16];
            boolean []locations=new boolean [16];
            //determine the likelihood of each chromosome in population
            //to be selected for crossing over
            computeFR();
            R1=Math.random();
            R2=Math.random();
            boolean nextfound1=false;
            for(int j=0;j<populationsize;j++) {
                    if(R1<=CFR[j]){
                        p1Index=j; //first parent index
                        break;  }
            }
            for(int j=0;j<populationsize;j++) {
                if(R2<=CFR[j]){
                    p2Index=j; //second parent index
                    break;  }
            }
            parent1=mypopulation.get(populationIndex[p1Index]); // get first parent
            parent2=mypopulation.get(populationIndex[p2Index]); // get second parent
            child1=parent1.clone();
            child2=parent2.clone();
            // determine the single crossing over point
            cp=(int)Math.floor(2*n*Math.random())+1;
            // exchange the parts to generate children
            int tempc1[]=new int [cp+1];
            for(int i=1;i<=cp;i++){
                tempc1[i]=child1[i];
                child1[i]=child2[i];
                child2[i]=tempc1[i];
            }
            mynextpopulation.add(child1.clone());// insert child1 to the next population
            mynextpopulation.add(child2.clone());// insert child2 to the next population
            ncdone=ncdone+2;
            if(ncdone==nCrossover){
                cdone=true; }
        }
}
```

```
private static void mutation_bit() {

        for(int k=beginMutationIndex;k<=endMutationIndex;k++){
                // determine the solution with
                // with worst fitness to mutate by "k"
                mutant=mypopulation.get(populationIndex[k]);
                // determine the bit randomly to change
                int mb = (int)Math.floor(2*n*Math.random())+1;
                // update the solution at that point
                mutant[mb]=1-mutant[mb];
                // insert the  mutated solution to the next population
                mynextpopulation.add(mutant.clone());
         }
}
```

APPENDIX 6. Codes of Functions for SA Algorithm

```
public static void main(String[] args) throws IOException  {
    // make "num" times experimental designs
    for (int iter=1;iter<=num;iter++){
    makeExperiementSetup(); // set experimental design parameters
    // run "numRep" times replication for a given experimental design
        for(int s=1; s<=numRep;s++){ /*
            set initial values for start and end time
            to find a solution for a given replication
            */
            double timeStart=0;
            double timeEnd=0;
            // mark start time for the solution
            timeStart=System.currentTimeMillis();
            initializesa(); // initialize SA algorithm
            annealing(); // start SA mechanism
            // mark end time for the solution
            timeEnd=System.currentTimeMillis();
            /*
            collect the best evaluated solution found
            as the result of the current replication
             */
            myEDGASAs.collect(finalsol);
            /*
            collect the solution time it takes to find
            best evaluated solution found for the current replication
            */
            myEDGASATs.collect((timeEnd-timeStart));
        } // end of each replication
        /*
        Report statistics for each replication:
        Average-R,R-min,R-max, standard error for reliability
        Report statistics for each replication:
        Average-T,T-min,T-max, standard error for solution time
        */
        System.out.println(iter +"  "+myEDGASAs.getAverage()+"  "+myEDGASAs.getMin()+
        "   "+myEDGASAs.getMax()+"  "+myEDGASAs.getStandardError()+"  "+"***"+"  "+
        myEDGASATs.getAverage()+"  "+myEDGASATs.getMin()+"  "+myEDGASATs.getMax()+
        "   "+myEDGASATs.getStandardError());
        } // end of each experimental design
    } // end of main
```

```
    public static void initializesa(){

        // initialize SA parameters
        v1=1;
        right[1]=CAvailable;
        right[2]=TAvailable;
        t=tinitial;
        // set initial solution as do nothing
        myfinalsol=myinitsol.clone();
        mycursol=myinitsol.clone();
        cursol=encomputeobj(mycursol);
        finalsol=cursol;
        // set final temperature for annealing
        tfinal=findtemperature(tinitial,m);
    }
```

```java
public static void annealing(){

    while(t>=tfinal){ //as long as current temperature is not less than final temperature

        for(int h=1;h<=n1;h++){ // for a given temperature
            mytempsol=nextmovesa(mycursol); // find new solution
            tempsol=encomputeobj(mytempsol); // evaluate new solution
            deltasol=tempsol-finalsol;   // compute the improvement by new solution
            if(deltasol>0){ // if new solution improves
                mycursol=mytempsol.clone(); // update current solution to the new solution
                cursol=tempsol;
                myfinalsol=mycursol.clone(); // update final solution to the current solution
                finalsol=cursol;
            }
            else { // if new solution does not improve
                if(Math.random()<Math.exp(-(double)(deltasol)/t)){ // by a certain probability
                    mycursol=mytempsol.clone(); // accept this inferior solution as a current solution
                    cursol=tempsol;
                }
            }
        } // end of attempts for a given temperature
        //t=t*alpha; /(another way of setting the next temperature)
        t=findtemperature(tinitial, v1); // set next temperature
        v1=v1+1; // update parameter used for setting the next temperature
        /*
         set current solution to the final solution
         obtained at previous temperature
         */
        mycursol=myfinalsol.clone();
        cursol=finalsol;
    } // end of annealing
}
```

APPENDIX 7. Codes of Functions for EDGASA Hybrid Algorithm

```
public static void main(String[] args) throws IOException  {
    // make "num" times experimental designs
    for (int iter=1;iter<=num;iter++){
        makeExperiementSetup(); / set experimental design parameters
        // run "numRep" times replication for a given experimental design
        for(int s=1; s<=numRep;s++){
            /*
            set initial values for start and end time
            to find a solution for a given replication
            */
            double timeStart=0;
            double timeEnd=0;
            // mark start time for the solution
            timeStart=System.currentTimeMillis();
            initialize(); // initialize EDGASA algorithm
            // generate initial solution
            getinitialpopulation();
            // sort solution according to their fitness
            insertionSort( populationFitness,populationsize);
            ga(); // start EDGASA mechanism
            // mark end time for the solution
            timeEnd=System.currentTimeMillis();
            /*
            collect the best evaluated solution found
            as the result of the current replication
             */
            myEDGASAs.collect(populationFitness[0]);
            /*
             collect the solution time it takes to find
             best evaluated solution found for the current replication
            */
            myEDGASATs.collect((timeEnd-timeStart));
        } // end of each replication
        /*
        Report statistics for each replication:
        Average-R,R-min,R-max, standard error for reliability
        Report statistics for each replication:
        Average-T,T-min,T-max, standard error for solution time
        */
        System.out.println(iter +"  "+myEDGASAs.getAverage()+"  "+myEDGASAs.getMin()+
        "  "+myEDGASAs.getMax()+"  "+myEDGASAs.getStandardError()+"  "+"***"+"  "+
        myEDGASATs.getAverage()+"  "+myGAFTSTs.getMin()+"  "+myGAFTSTs.getMax()+
        "  "+myGAFTSTs.getStandardError());
    } // end of each experimental design
} // end of main
```

```java
public static void initialize(){
    myinitsol=initialsolution();
    mycursol=myinitsol.clone();
    cursol=encomputeobj(mycursol);
    finalsol=cursol;

    // initialize parameters that are used for SA in EDGASA
    t=tinitial;
    tfinal=findtemperature(tinitial,m);
    v1=0;
    right[1]=CAvailable;
    right[2]=TAvailable;

    // initialize parameters that are used for GA in EDGASA
    populationIndex =new int[populationsize];
    populationFitness=new double[populationsize];
    FR=new double[populationsize];
    CFR=new double[populationsize];
    nElicit=(int) (populationsize*pElicit);
    nMutation=(int) (populationsize*pMutation);
    nCrossover=(int) (populationsize*pCrossover);
    endElicitIndex=nElicit-1;
    beginCrossoverIndex=nElicit;
    endCrossoverIndex=nElicit+nCrossover-1;
    beginMutationIndex=endCrossoverIndex+1;
    endMutationIndex=populationsize-1;
    mypopulation.clear();
    mynextpopulation.clear();
}
```

```
public static void getinitialpopulation(){
    double asol=0;
    // start with a do nothing initial solution for SA used in EDGASA
    mycursol= initialsolution().clone();
    myfinalsol=mycursol.clone();
    cursol=encomputeobj(mycursol);
    finalsol=cursol;
    t=tinitial;
    myfinalsol=myinitsol.clone();
    mycursol=myinitsol.clone();
    cursol=encomputeobj(mycursol);
    tfinal=findtemperature(tinitial,m);
    right[1]=CAvailable;        right[2]=TAvailable;
    annealing(); // apply SA
    if(finalsol<0){
        // assign the fitness of solution when it is infeasible
        populationFitness[0]=1/Math.abs(finalsol);  }
    else{
        // assign the fitness of solution when it is feasible
        populationFitness[0]=finalsol;  }
        // add the finalsolution obtained from SA into initital population
    mypopulation.add(myfinalsol.clone());
    // continue SA recursivelt number of times equal to recursive
    for(int o=1;o<=recursive;o++){
        // initialize annealing parameters
        t=tinitial;
        tfinal=findtemperature(tinitial,m);
        annealing(); // apply SA similar to above
        if(finalsol<0){
            populationFitness[0]=1/Math.abs(finalsol);  }
        else{
            populationFitness[0]=finalsol;  }
        mypopulation.add(myfinalsol.clone());
    }
    for(int i=recursive+1;i<populationsize;i++) {
        // generate randomly the rest of the solutions in the initial population
        newchromosome = randominitialsolution().clone();
        populationIndex[i]=i;
        asol=encomputeobj(newchromosome);
        if(asol<0){
            populationFitness[i]=1/Math.abs(asol);}
        else{
            populationFitness[i]=asol;}
        mypopulation.add(newchromosome.clone());
    }
} // end of getinitializepopulation
```

```
public static void ga() {
    // generate and evaulate population up to specified number
    for(int i=1;i<=generationnumber;i++){
        getnextpopulation();
        updateIndex();
        insertionSort( populationFitness,populationsize);
    }
}
```

```
public static void getnextpopulation(){
    // getnext population mechanism for EDGASA
    // similar to that for GA except the difference in eliciting
    mynextpopulation.clear();
    elicit();
    s_r_crossover();
    mutation_bit();
    mypopulation=(ArrayList) mynextpopulation.clone();
}
```

```
private static void elicit() {
    // get the best solution in the population
    tempchromosome= (int[]) mypopulation.get(populationIndex[0]);
    // set annealing parameters strating from this solution
    mycursol=tempchromosome.clone();
    myfinalsol=mycursol.clone();
    cursol=encomputeobj(mycursol);
    t=tinitial;
    tfinal=findtemperature(tinitial,m);
    v1=1;
    finalsol=cursol;
    annealing (); // apply annealing
    // add the final solution obtained from the SA into nextpopulation
    mynextpopulation.add(myfinalsol.clone());
    // starting from the final solution obtained above
    // apply SA recursively specified number of times
    for(int o=1;o<=recursive;o++){
        // initialize parameters for SA
        t=tinitial;
        tfinal=findtemperature(tinitial,m);
        v1=1;

        annealing (); // apply annealing
        // add the final solution obtained from annealing
        // into nextpopulation
        mynextpopulation.add(myfinalsol.clone());
    }

}
```

APPENDIX 8. Results Obtained from GA

Experiment No	Average R	R-min	R-max	Standard Error		Average Time	Min Time	Max Time
1	0,687440	0,613922	0,742850	0,011549	***	56,3	46	129
2	0,678824	0,540079	0,765085	0,021950	***	48,3	47	50
3	0,751765	0,681254	0,826583	0,013189	***	49,1	47	55
4	0,674582	0,603715	0,751341	0,016576	***	48,6	47	51
5	0,454754	0,370622	0,591898	0,020826	***	48,2	47	51
6	0,622029	0,539559	0,688785	0,016329	***	47,9	46	51
7	0,547025	0,436013	0,653130	0,024485	***	47,1	44	53
8	0,793690	0,677980	0,827405	0,013202	***	47,8	47	50
9	0,737874	0,641151	0,787757	0,014071	***	48,8	45	53
10	0,775584	0,746260	0,812039	0,006461	***	55,6	52	61
11	0,688360	0,592826	0,768652	0,016804	***	50,0	47	55
12	0,632743	0,575544	0,714534	0,012923	***	49,8	46	54
13	0,452717	0,375142	0,549149	0,018071	***	51,6	48	55
14	0,573178	0,405668	0,686715	0,028640	***	48,8	47	51
15	0,410613	0,256494	0,572826	0,030570	***	55,0	47	61
16	0,661439	0,628390	0,710713	0,009005	***	49,5	47	51
17	0,699478	0,654115	0,749920	0,010411	***	47,0	45	49
18	0,620941	0,539739	0,714134	0,016757	***	53,7	46	58
19	0,536633	0,494548	0,592186	0,009106	***	53,3	46	58
20	0,731437	0,648317	0,773403	0,011832	***	51,5	47	57
21	0,668108	0,620738	0,753983	0,013274	***	49,6	48	52
22	0,644410	0,554411	0,781035	0,026949	***	54,1	48	58
23	0,709667	0,616252	0,766081	0,015474	***	49,3	46	55
24	0,353597	0,138915	0,603743	0,042491	***	49,7	48	52
25	0,699178	0,658135	0,729908	0,007488	***	48,9	47	53
26	0,436448	0,241816	0,667427	0,040052	***	50,0	49	52
27	0,282473	0,063601	0,574676	0,058837	***	49,6	47	55
28	0,675367	0,570691	0,766268	0,021016	***	47,6	45	51
29	0,722478	0,641396	0,763817	0,013313	***	49,7	48	52
30	0,630214	0,544824	0,717323	0,018198	***	49,9	47	53
31	0,746787	0,655184	0,795197	0,013638	***	47,5	45	50
32	0,588274	0,513214	0,662180	0,015073	***	48,9	46	53
33	0,681593	0,646392	0,744250	0,009655	***	46,8	45	49
34	0,467769	0,355084	0,546209	0,017747	***	48,1	46	51
35	0,615271	0,498016	0,704611	0,018565	***	48,3	46	53
36	0,227501	0,096752	0,347713	0,027715	***	49,2	46	55
37	0,693593	0,633122	0,767353	0,014114	***	47,3	45	52
38	0,325243	0,061969	0,514790	0,051525	***	47,5	46	50
39	0,694398	0,578912	0,833625	0,025197	***	47,1	44	52
40	0,491900	0,279183	0,569303	0,029920	***	48,7	46	52
41	0,366544	0,311577	0,401614	0,007858	***	48,6	46	52
42	0,722917	0,669441	0,752516	0,008844	***	46,8	45	50
43	0,605055	0,547966	0,693466	0,017077	***	47,8	46	55
44	0,428617	0,243844	0,525715	0,025248	***	49,6	47	55
45	0,682299	0,618098	0,724277	0,008935	***	49,0	46	55
46	0,664019	0,608181	0,712651	0,012341	***	49,8	47	55
47	0,515142	0,402044	0,644951	0,028204	***	47,3	45	50
48	0,596402	0,513122	0,681261	0,016120	***	48,0	46	51
49	0,668572	0,613327	0,710287	0,010476	***	48,9	46	53
50	0,576414	0,485395	0,704180	0,021996	***	49,6	47	52

Experiment No	Average R	R-min	R-max	Standard Error		Average Time	Min Time	Max Time
51	0,390572	0,199076	0,568919	0,038970	***	47,9	46	50
52	0,745675	0,672119	0,809281	0,013846	***	47,9	46	50
53	0,583844	0,462175	0,674813	0,022291	***	48,4	46	51
54	0,614176	0,504510	0,689639	0,019781	***	49,9	47	54
55	0,722985	0,636818	0,811410	0,017462	***	50,1	47	57
56	0,489499	0,489499	0,489499	0,000000	***	48,9	47	53
57	0,290675	0,119650	0,471676	0,034467	***	52,1	50	56
58	0,109080	0,011435	0,450635	0,048074	***	50,5	48	54
59	0,779928	0,716073	0,839974	0,014212	***	47,4	45	51
60	0,492253	0,383527	0,590022	0,024127	***	51,0	47	58
61	0,668171	0,502848	0,753398	0,022309	***	48,1	46	50
62	0,271680	0,115734	0,437242	0,030236	***	47,9	47	50
63	0,289243	0,160211	0,408029	0,029132	***	49,4	47	52
64	0,596327	0,490392	0,703313	0,020668	***	48,0	46	50
65	0,473731	0,294945	0,667630	0,032240	***	47,9	46	50
66	0,454638	0,262155	0,577349	0,025889	***	49,0	46	52
67	0,411657	0,157367	0,548763	0,033603	***	48,3	46	51
68	0,441123	0,378941	0,551280	0,018531	***	47,8	46	51
69	0,454159	0,310672	0,625316	0,032782	***	48,1	47	51
70	0,705571	0,604935	0,781427	0,016847	***	48,0	46	51
71	0,567937	0,523598	0,607478	0,008722	***	48,4	46	52
72	0,316962	0,187059	0,433055	0,024172	***	48,6	47	52
73	0,415697	0,282375	0,583305	0,027999	***	48,8	46	54
74	0,614628	0,451175	0,719321	0,025411	***	49,1	46	53
75	0,750755	0,711396	0,800243	0,009792	***	47,2	46	49
76	0,591392	0,537935	0,667416	0,012976	***	48,9	47	54
77	0,612809	0,484129	0,721619	0,027152	***	47,4	46	50
78	0,487545	0,350257	0,651900	0,033949	***	48,1	46	50
79	0,608866	0,573948	0,677275	0,009482	***	47,2	46	49
80	0,564205	0,389889	0,645292	0,025505	***	50,1	46	58
81	0,422127	0,248734	0,573875	0,044390	***	51,3	47	57
82	0,681371	0,613627	0,734742	0,013577	***	51,1	47	57
83	0,669198	0,624401	0,715047	0,009454	***	52,4	45	57
84	0,610274	0,572796	0,674949	0,010178	***	50,6	47	60
85	0,750458	0,697379	0,784392	0,008322	***	53,7	46	57
86	0,805027	0,773945	0,830047	0,007057	***	52,0	46	60
87	0,429157	0,298061	0,527248	0,022768	***	50,3	47	53
88	0,241340	0,040309	0,463919	0,049825	***	50,0	47	57
89	0,554915	0,456995	0,682209	0,023083	***	48,5	46	51
90	0,533150	0,371452	0,673398	0,029864	***	49,2	47	52
91	0,569100	0,433842	0,667828	0,020497	***	48,4	47	52
92	0,665127	0,611422	0,746672	0,012595	***	47,7	45	52
93	0,545560	0,432526	0,664188	0,023901	***	47,2	44	50
94	0,628912	0,540589	0,690734	0,013037	***	48,1	46	52
95	0,751985	0,708385	0,800789	0,010381	***	47,6	46	50
96	0,618007	0,539661	0,698949	0,013788	***	49,6	47	55
97	0,681726	0,631263	0,726614	0,010002	***	49,5	47	53
98	0,667292	0,563740	0,786763	0,019699	***	55,0	48	76
99	0,735036	0,642503	0,813471	0,017919	***	50,0	47	54
100	0,680713	0,643342	0,725525	0,007958	***	47,9	47	50

APPENDIX 9. Results Obtained From SA

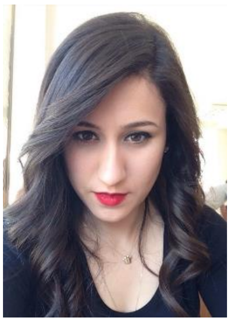
Experiment No	Average R	R-min	R-max	Standard Error		Average Time	Min Time	Max Time
1	0,722815	0,676502	0,754335	0,007804	***	197,2	180	294
2	0,749214	0,706796	0,793760	0,008240	***	187,1	179	201
3	0,784764	0,744860	0,816533	0,006639	***	188,6	179	195
4	0,741770	0,642659	0,790495	0,013797	***	200,1	184	219
5	0,558556	0,483425	0,623699	0,013702	***	192,2	180	208
6	0,695783	0,666677	0,746435	0,008975	***	199,7	185	215
7	0,660733	0,596096	0,705692	0,009888	***	188,1	179	206
8	0,789648	0,745584	0,812504	0,006656	***	184,7	182	190
9	0,765473	0,726321	0,795440	0,007861	***	186,3	182	190
10	0,800147	0,788798	0,819709	0,003714	***	183,7	180	193
11	0,782505	0,741363	0,820067	0,007437	***	184,3	180	190
12	0,708913	0,651738	0,743915	0,010595	***	183,9	179	193
13	0,599841	0,517847	0,662676	0,015518	***	187,3	179	203
14	0,759640	0,714397	0,803751	0,011586	***	204,3	191	225
15	0,598053	0,498816	0,676476	0,016614	***	192,9	183	222
16	0,710033	0,677862	0,742026	0,007414	***	186,7	179	192
17	0,724902	0,693255	0,747015	0,005762	***	191,9	187	197
18	0,724211	0,656812	0,763358	0,009467	***	186,8	181	197
19	0,599184	0,541148	0,679524	0,011831	***	187,1	182	193
20	0,757631	0,723721	0,784051	0,006624	***	187,6	182	194
21	0,708488	0,687089	0,742878	0,006444	***	190,1	182	207
22	0,789391	0,748234	0,828304	0,007515	***	188,6	184	195
23	0,788957	0,735033	0,824312	0,007334	***	193,0	185	215
24	0,685054	0,625925	0,753691	0,012444	***	187,6	181	192
25	0,708505	0,663416	0,744932	0,008217	***	186,9	182	198
26	0,667949	0,606400	0,732706	0,013660	***	187,6	179	191
27	0,632224	0,550785	0,706158	0,016357	***	184,9	180	191
28	0,738221	0,676693	0,782266	0,009909	***	185,2	178	195
29	0,779976	0,750048	0,822114	0,007811	***	184,2	179	188
30	0,697329	0,668391	0,742438	0,007671	***	184,9	182	189
31	0,776083	0,765108	0,792491	0,002815	***	193,2	187	219
32	0,725151	0,677189	0,802534	0,011990	***	194,6	186	230
33	0,758320	0,719550	0,796993	0,008881	***	188,3	180	197
34	0,565021	0,496435	0,597701	0,010193	***	191,2	186	203
35	0,636540	0,609080	0,657541	0,004368	***	189,7	183	195
36	0,461383	0,347476	0,645205	0,030312	***	186,7	184	194
37	0,751804	0,711138	0,776616	0,006767	***	188,1	180	196
38	0,550350	0,463330	0,634278	0,017582	***	189,5	182	213
39	0,784155	0,753532	0,815132	0,006001	***	184,6	178	194
40	0,678603	0,617862	0,718926	0,009979	***	189,2	183	199
41	0,428785	0,385762	0,494252	0,012407	***	190,0	181	205
42	0,745941	0,716503	0,781268	0,007616	***	186,7	180	196
43	0,659314	0,577693	0,739717	0,015352	***	192,6	186	209
44	0,630176	0,551891	0,700045	0,014443	***	190,5	185	197
45	0,707047	0,683428	0,741238	0,005086	***	187,3	180	192
46	0,671788	0,594921	0,726799	0,011983	***	192,3	188	205
47	0,614562	0,509748	0,674968	0,017231	***	190,5	184	207
48	0,678908	0,658953	0,709420	0,005510	***	185,8	180	189
49	0,701817	0,672968	0,729096	0,005195	***	185,6	181	194
50	0,680005	0,610969	0,703086	0,009124	***	197,3	183	210

Experiment No	Average R	R-min	R-max	Standard Error		Average Time	Min Time	Max Time
51	0,620470	0,507255	0,686736	0,020256	***	210,2	204	219
52	0,799105	0,777481	0,824150	0,004496	***	202,6	193	216
53	0,618832	0,567003	0,676998	0,013040	***	208,2	204	213
54	0,712080	0,651465	0,764710	0,011733	***	216,7	209	227
55	0,771752	0,742250	0,810719	0,009692	***	224,2	209	239
56	0,695206	0,576924	0,759750	0,020437	***	213,2	202	218
57	0,522647	0,352505	0,593360	0,022732	***	207,1	180	215
58	0,486837	0,415332	0,555415	0,017362	***	208,7	186	229
59	0,817483	0,781860	0,854971	0,007066	***	189,5	182	211
60	0,599500	0,516862	0,655052	0,013632	***	189,3	184	200
61	0,708505	0,631922	0,761220	0,012885	***	189,0	180	199
62	0,501576	0,418920	0,614757	0,017228	***	188,2	183	198
63	0,590752	0,416470	0,676368	0,023113	***	193,8	182	216
64	0,742168	0,658449	0,790266	0,011742	***	187,2	180	197
65	0,665983	0,575851	0,741613	0,016872	***	187,4	182	195
66	0,555130	0,481674	0,616509	0,013723	***	187,1	181	191
67	0,594863	0,468601	0,723028	0,020849	***	185,2	180	193
68	0,600080	0,521719	0,645376	0,012178	***	187,7	183	198
69	0,577180	0,539153	0,626827	0,007314	***	185,2	180	190
70	0,762770	0,743068	0,794418	0,005564	***	185,9	181	193
71	0,660943	0,613925	0,745326	0,012705	***	187,7	183	199
72	0,597716	0,504635	0,647062	0,014214	***	195,1	183	209
73	0,618988	0,549677	0,704046	0,013916	***	185,8	179	196
74	0,686834	0,610091	0,766412	0,016064	***	183,7	179	192
75	0,792478	0,747026	0,818858	0,006450	***	183,8	178	191
76	0,649229	0,618715	0,672772	0,006386	***	184,5	180	189
77	0,670804	0,612935	0,719307	0,010331	***	190,6	181	216
78	0,660192	0,608627	0,740755	0,014816	***	186,1	181	194
79	0,635537	0,605690	0,677040	0,007388	***	187,5	180	195
80	0,667314	0,641602	0,699540	0,005693	***	184,8	178	194
81	0,670100	0,494116	0,746077	0,024887	***	185,4	181	195
82	0,716197	0,661289	0,755221	0,010311	***	200,1	183	211
83	0,750911	0,727697	0,779053	0,006119	***	195,8	183	212
84	0,673412	0,643785	0,705511	0,006792	***	187,1	182	198
85	0,765007	0,736925	0,795761	0,007054	***	188,4	182	192
86	0,825397	0,782265	0,857721	0,007691	***	191,7	182	218
87	0,567619	0,507921	0,630162	0,014575	***	189,7	184	196
88	0,631016	0,500279	0,776754	0,028208	***	188,1	181	201
89	0,660698	0,565987	0,757940	0,019739	***	186,7	182	193
90	0,644705	0,600887	0,685440	0,009834	***	185,1	180	188
91	0,672850	0,598894	0,718816	0,011497	***	186,7	180	189
92	0,739634	0,709121	0,777039	0,006638	***	185,9	181	193
93	0,641573	0,597554	0,697771	0,010780	***	186,5	179	195
94	0,692673	0,648729	0,745526	0,009374	***	185,8	181	192
95	0,774476	0,753928	0,799735	0,005838	***	188,0	181	197
96	0,705221	0,621627	0,776746	0,017461	***	188,6	185	199
97	0,708666	0,656194	0,749361	0,010038	***	188,3	181	195
98	0,783601	0,718366	0,822372	0,010032	***	185,7	179	196
99	0,818694	0,785041	0,851743	0,006441	***	183,3	178	188
100	0,719784	0,688040	0,743481	0,006009	***	185,9	181	196

APPENDIX 10. Results Obtained From EDGASA

Experiment No	Average R	R-min	R-max	Standard Error		Average Time	Min Time	Max Time
1	0,775605	0,770169	0,776209	0,006405	***	92123,5	91882	92348
2	0,793760	0,793760	0,793760	0,000000	***	92652,4	92423	92935
3	0,844172	0,836307	0,845895	0,001158	***	92296,6	92175	92453
4	0,820449	0,820449	0,820449	0,000000	***	93254,3	93041	93584
5	0,672472	0,672472	0,672472	0,000000	***	74447,1	92749	6451482
6	0,754873	0,753658	0,755177	0,001860	***	92485,6	87149	93272
7	0,742086	0,726134	0,745896	0,002542	***	93663,5	91956	97532
8	0,842098	0,840567	0,845503	0,002717	***	96447,8	93681	100360
9	0,829287	0,823656	0,832811	0,001205	***	97793,8	95205	101226
10	0,822553	0,822125	0,824265	0,0063047	***	94759,4	92385	100447
11	0,832515	0,827098	0,835169	0,001097	***	92826,2	92732	92896
12	0,768326	0,760102	0,770382	0,001371	***	92475,0	92213	92632
13	0,681849	0,669922	0,689500	0,002665	***	93005,2	92840	93223
14	0,818048	0,818048	0,818048	0,000000	***	93015,9	92879	93268
15	0,713447	0,703852	0,714514	0,001066	***	93067,7	92994	93287
16	0,764518	0,760193	0,766605	0,008017	***	92560,3	92346	92817
17	0,793579	0,772627	0,796068	0,002329	***	93251,3	93143	93384
18	0,772895	0,764966	0,775346	0,001099	***	93159,6	93042	93445
19	0,669551	0,658367	0,681987	0,003125	***	92650,9	92455	92769
20	0,797678	0,796372	0,798237	0,006626	***	92594,5	92224	93461
21	0,767582	0,761731	0,769044	0,008080	***	93219,3	92884	94703
22	0,828304	0,828304	0,828304	0,000000	***	93249,5	93091	93356
23	0,822045	0,815074	0,824312	0,001174	***	92387,3	92258	92513
24	0,812849	0,806013	0,817406	0,001861	***	93399,3	93300	93622
25	0,774321	0,755059	0,776461	0,002140	***	92707,6	92553	92814
26	0,781047	0,779510	0,781217	0,003989	***	93230,2	93099	93439
27	0,768654	0,760757	0,776551	0,002632	***	93288,4	93153	93706
28	0,797525	0,785909	0,801535	0,001850	***	93119,2	92982	93303
29	0,840547	0,827428	0,842005	0,001458	***	93187,7	93117	93258
30	0,798178	0,798178	0,798178	0,000000	***	93065,3	92953	93147
31	0,846308	0,843732	0,847807	0,007673	***	93175,5	93093	93271
32	0,802534	0,802534	0,802534	0,000000	***	93094,8	92978	93175
33	0,814666	0,810163	0,815792	0,007291	***	92868,4	92767	93032
34	0,656548	0,651620	0,659834	0,001341	***	93831,2	93683	93918
35	0,717289	0,707123	0,718419	0,001130	***	93199,7	93089	93326
36	0,671424	0,669659	0,671620	0,000684	***	93269,2	93100	93409
37	0,804405	0,798315	0,805297	0,001345	***	92955,9	92837	93151
38	0,685301	0,661592	0,693366	0,003583	***	93139,0	92917	93345
39	0,875043	0,873029	0,876386	0,006692	***	92874,0	92768	92951
40	0,769090	0,769090	0,769090	0,000000	***	93443,9	93345	93567
41	0,550033	0,508184	0,554683	0,004650	***	93657,8	93585	93708
42	0,797461	0,791873	0,800502	0,004293	***	93353,9	93207	93542
43	0,771511	0,762146	0,776584	0,001591	***	93811,5	93686	93934
44	0,762579	0,759794	0,762889	0,009898	***	93831,5	93736	93939
45	0,753247	0,745855	0,759190	0,001665	***	93167,5	93049	93229
46	0,754215	0,754215	0,754215	0,000000	***	93075,2	92938	93217
47	0,702194	0,676883	0,708950	0,004220	***	92976,1	92857	93113
48	0,738921	0,728682	0,744600	0,002372	***	92869,6	92748	93017
49	0,757057	0,757057	0,757057	0,000000	***	93431,5	93245	93529
50	0,789080	0,788105	0,797570	0,007925	***	93381,7	93113	93627

Experiment No	Average R	R-min	R-max	Standard Error		Average Time	Min Time	Max Time
51	0,724471	0,720911	0,725996	0,004946	***	929786,0	92823	93091
52	0,837522	0,837341	0,837568	0,002207	***	92827,2	92713	93144
53	0,739379	0,724876	0,740991	0,001611	***	93234,8	93080	93375
54	0,768615	0,766764	0,770855	0,008474	***	93717,3	93636	93843
55	0,820340	0,818470	0,823144	0,000920	***	93233,9	93098	93352
56	0,811810	0,800678	0,819231	0,003030	***	93029,4	92881	93194
57	0,681741	0,666820	0,683399	0,001658	***	93312,4	93177	93414
58	0,594955	0,582283	0,598891	0,002054	***	93684,5	93489	93943
59	0,881445	0,881445	0,881445	0,000000	***	93349,2	92581	94730
60	0,665812	0,662724	0,667870	0,005129	***	93445,4	93293	93611
61	0,788568	0,788568	0,788568	0,000000	***	92929,1	92757	93054
62	0,651196	0,631423	0,655074	0,002391	***	93066,0	92637	93658
63	0,697987	0,697488	0,699152	0,000371	***	93218,5	92860	93979
64	0,832614	0,828797	0,835803	0,001103	***	93104,9	92996	93256
65	0,756678	0,742524	0,764290	0,002471	***	93103,4	92981	93322
66	0,635484	0,629851	0,647092	0,001383	***	93170,0	92611	93470
67	0,746310	0,719359	0,755094	0,004773	***	92595,1	92452	92830
68	0,683977	0,666259	0,697102	0,003699	***	92758,6	92632	92923
69	0,679791	0,662165	0,690132	0,003409	***	93722,6	93551	93867
70	0,810057	0,810057	0,810057	0,000000	***	92842,2	92637	93008
71	0,769725	0,751944	0,779983	0,004198	***	93170,6	93027	93379
72	0,683942	0,659742	0,688721	0,003275	***	92747,2	92595	92958
73	0,726186	0,717163	0,736528	0,002930	***	92617,9	92416	92780
74	0,780639	0,779819	0,781185	0,009132	***	92861,8	92431	93169
75	0,858029	0,852996	0,859698	0,006005	***	92516,0	92279	92656
76	0,708398	0,704158	0,716828	0,001841	***	92856,9	92579	93073
77	0,748231	0,719315	0,761216	0,003394	***	92814,3	92611	92998
78	0,770961	0,754372	0,780461	0,003275	***	94667,4	92900	101658
79	0,715724	0,714496	0,715860	0,009517	***	92916,5	92771	93083
80	0,724879	0,706408	0,731939	0,003021	***	92718,1	92420	92856
81	0,783635	0,780577	0,785675	0,002001	***	93041,6	92840	93207
82	0,765483	0,757352	0,767406	0,004292	***	93266,2	93128	93360
83	0,816756	0,814911	0,819524	0,006969	***	93454,0	93272	93538
84	0,722837	0,714570	0,723755	0,007518	***	93266,5	93039	93481
85	0,808259	0,797088	0,816691	0,002387	***	93039,9	92888	93320
86	0,897441	0,897441	0,897441	0,000000	***	93289,5	93162	93436
87	0,678681	0,659114	0,690226	0,004065	***	92558,8	92464	92666
88	0,773500	0,760484	0,776754	0,002169	***	93089,0	92880	93618
89	0,757940	0,757940	0,757940	0,000000	***	93140,3	92656	93436
90	0,728235	0,706866	0,733647	0,002653	***	92336,2	91988	92599
91	0,751144	0,749099	0,753124	0,008430	***	93062,9	92860	93258
92	0,807854	0,804161	0,808460	0,009336	***	92797,2	92308	93049
93	0,732752	0,729446	0,733565	0,009816	***	92944,8	92882	93000
94	0,743951	0,739224	0,746158	0,001035	***	93253,5	93127	93421
95	0,815023	0,812192	0,815731	0,002777	***	94018,4	93232	95451
96	0,788506	0,782479	0,789482	0,002627	***	92569,1	92466	92688
97	0,765613	0,765613	0,765613	0,000000	***	92442,5	92356	92575
98	0,842928	0,842928	0,842928	0,000000	***	92577,4	92413	92776
99	0,851743	0,851743	0,851743	0,000000	***	92172,5	91992	92334
100	0,771779	0,765686	0,776055	0,001662	***	92505,0	92347	92649



CURRICULUM VITAE

Contact Information

@mail **:** essradag@gmail.com

Personal Information

Place of Birth **:** Gaziantep
Date of Birth **:** 25 December, 1990
Marital Status **:** Single

Education

- **Toros Unıversıty,** Department of Industrial Engineering (M.Sc**.** 2013-2015),
  CGPA : 4.00 / 4.00 **ALES QTY:** 70.06

  Thesis Title: Optimization of the Sustainability of Contingency Logistics Networks: Application of a Hybrid Heuristic & a Multi-Objective Optimization Approaches.

- **Anadolu University,** Management (B.Sc., 2012 -), Mersin

- **Yaşar University, Computer Engineering** (B.Sc., 2008 - 2012), İzmir

Computer Skills

- **Programming Languages:** C++, C#, Java, HTML, SharePoint, Expression Web, JavaScript, MySQL, MS Access

- **Applications:** MS Office Applications, AMPL, Photoshop

- **Operating Systems:** Windows, Mac OS x, Linux

Experiences

- **Atako Uluslararası Taşımacılık, Denizcilik ve Ticaret Ltd. Şti.,** Information Systems (Intern), Mersin

- **Edc Drink Nakliye ve Lojistik Gıda Sanayi Ltd. Şti.,** Information Systems

- **Toros University,** Research Assistant, Mersin

  <u>Courses Assisted:</u>

  - PSY107 :Statistics in Behavioral Science-I
  - MAN213 :Statistics
  - MAT311 :Engineering Statistics
  - INE403 :Supply Chain Management

References

- Prof. Dr. D. Tayyar Şen,
  Toros Universıty, Department of Industrial Engineering,
  Phone: 0324-325-33-00 @mail: tayyar.sen@toros.edu.tr

- Assist. Prof. Dr. Mehmet Miman,
  Toros Universıty, Department of Industrial Engineering,
  Phone: 0324-325-33-00 @mail: mehmet.miman@toros.edu.tr

- Funda Uluer, General Director,
  Edc Drink Nakliye ve Lojistik Gıda Sanayi Ltd. Şti,
  Phone: 0535-660-56-22 @mail: edc@edc-e.com

Publications

- Dağ, E. ve Miman, M. (2014) “Beklenmedik Durumlar Lojistiğinin Optimizasyonunda Ütopya Uzaklık Metodu”, III. Ulusal Lojistik ve Tedarik Zinciri Kongresi, s. 680-688, 15-17 May 2014, KTÜ University, Trabzon.

- Dağ, E. ve Miman, M. (2015) “Multi-Objective Optimization of Contingency Logistics Networks with Distorted Risks”, ICOVACS, pp: 487-499, 14-15 March 2015, Marmara University, İstanbul.

- Miman, M., Dağ, E. and Güloğlu, U. (2015, Working Paper) “A rearch on Corporate Social Responsibility Perceptions: ATAKO Example”, XIII International Logistics and Supply Chain Congress, 22-23 October, 2015, İzmir University, İzmir.