APPROXIMATION METHODS OF VECTOR OPTIMIZATION WITH APPLICATION TO THE DEMAND-RESPONSE MODEL OF SMART GRIDS

BY

NORAH ABDULLATIF MOHAMMED ALMURAYSIL

A Dissertation Presented to the

COLLEGE OF GRADUATE AND INTERDISCIPLINARY STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

MATHEMATICS

SEPTEMBER 2024

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This dissertation, written by **NORAH ABDULLATIF MOHAMMED ALMURAYSIL** under the direction of his dissertation adviser and approved by his dissertation committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN MATHEMATICS**.

Dissertation Committee

Dr. Mohammed Alshahrani (Adviser)

Dr. Slim Belhaiza (Co-Adviser)

Prof. Suliman Alhomaidan (Member)

Prof. Izhar Ahmed (Member)

Prof. Qamrul Hassan Ansari (Member)

Prof. Monther Rashed Saleh Alfuraidan
Department Chairman

Prof. Suliman Saleh Al- Homidan
Dean of Graduate Studies

Date

©Norah Abdullatif Almuraysil
2024

Dedication

This thesis is dedicated with love and appreciation to my dear parents, my supportive husband, my cherished siblings, and my lovely children.

ACKNOWLEDGMENTS

In the Name of Allah, the Beneficent, the Merciful.

First and foremost, praise be to Allah, the Almighty, for the blessings that have enabled me to complete this academic endeavor. I would like to express my deepest gratitude and appreciation to my advisor, Dr. Mohammed Alshahrani, for his patience, guidance, and and generous support throughout my research journey. His expertise was instrumental in establishing a robust foundation in my area of focus and in facilitating my acquisition of a new programming language. I would also like to express my appreciation to Dr. Slim Belhaiza for his valuable assistance, time, and knowledge, which were crucial in developing the new model and broadening my research horizons. My sincere thanks go to all committee members for their insightful feedback and suggestions, which significantly enhanced the quality of this work.

I am deeply indebted to King Fahd University of Petroleum and Minerals for providing me with the opportunity to undertake this research study. I extend my appreciation to Prof. Monther Alfuraidan, the esteemed Chairman of the Mathematics Department, for creating a supportive environment that has enabled us to continue our academic pursuits. Additionally, I acknowledge the significant contributions of Dr. Husain Al-Attas, the former Chairman of the Mathematics

Department, whose support was pivotal during the initial stages of my academic journey at KFUPM. I express my heartfelt gratitude to Imam Abdulrahman Bin Faisal University for granting me a scholarship and funding my educational expenses, which made this academic pursuit possible.

My heartfelt appreciation go to my father, Abdullatif Almuraysil, and my mother, Jammelah Alkudair, whose unwavering prayers have been a fundamental pillar of my achievements. I am deeply indebted to my husband, Mohammed Alwusayfir, for his constant encouragement and invaluable support throughout this challenging trip. Additionally, I would like to convey my gratefulness to my siblings, Mohammed, Marwah, Abdulrahman, Anas, Nawaf, and Modawi, for their constant affection and motivation. I also extend my sincere appreciation to my in-laws for their understanding and support, which have been invaluable throughout this journey. I am grateful to my children, my little man Abdullatif and my daughter princess Sarah, whose presence and joy have been a driving force for me.

Lastly, I would like to acknowledge my dear friend Hajar Alshaikh for her unwavering support during challenging periods of my academic journey. Additionally, I extend my thanks to my friends Sara AlKiswani who generously dedicated her time and my friends Amal Alhujaylan who was always supportive.

TABLE OF CONTENTS

ACKNOWLEDGMENTS **vi**

LIST OF TABLES **xi**

LIST OF FIGURES **xiii**

LIST OF ABBREVIATIONS **xiv**

ABSTRACT (ENGLISH) **xvi**

ABSTRACT (ARABIC) **xviii**

CHAPTER 1 INTRODUCTION **1**

1.1 Preliminaries 3

CHAPTER 2 LINEAR AND CONVEX VECTOR OPTIMIZATION **8**

2.1 Linear Vector Optimization 8

2.2 Approximation Algorithm of LVOP 9

2.3 Convex Vector Optimization 11

2.3.1 Outer Approximation Algorithms 13

2.4 Scalarization 15

2.4.1 The Weighted Sum Scalarization 16

2.4.2 Pascoletti-Serafini Scalarization 18

2.4.3 The Norm-Minimizing Scalarization 19

CHAPTER 3 OUTER APPROXIMATION ALGORITHMS FOR CVOP **22**

3.1 Direction 23

3.2 Vertex Selection 26

3.3 Enhanced Outer Approximation Algorithms 28

3.3.1 Improvement of Pascoletti-Serafini Scalarization. 28

3.3.2 Approximate (Adj) Direction Approach 31

CHAPTER 4 APPLICATION: SMART GRID DEMAND RESPONSE PROBLEM **40**

4.1 Smart Grid 41

4.2 Mathematical Formulation 44

CHAPTER 5 APPLICATION: DYNAMIC ECONOMIC EMISSIONS DISPATCH (DEED) **52**

5.1 DEED Problem 52

5.2 Dynamic Economic Emission Dispatch Of Electric Power Generation . 54

5.2.1 Mathematical Formulation 55

5.2.2 A Game-Theoretic DR-DEED formulation 59

5.2.3 Integrated DEED and Multi-Periodic Game-Theoretic Formulation 63

5.3 Applying Approximation Algorithm 64

CHAPTER 6 DYNAMIC ECONOMIC EMISSION DISPATCH IN MULTI-PERIOD SMART GRID DEMAND RESPONSE (NEW MODEL) **70**

6.1 Game Theoretic Models For Smart Grids With Shifted Demand 72

6.2 The Proposed Model 76

6.2.1 Energy Flow Conservation Constraints 78

6.3 Numerical Simulations and Results 84

6.3.1 Experiment 1 84

6.3.2 Experiment 2 92
6.4 Applying the Approximation Algorithm to SGSD-DEED 101

CHAPTER 7 CONCLUSION AND FUTURE WORKS 104

APPENDIX A REPRODUCING THE RESULTS 107
A.1 OAFCVX.jl 107
A.2 DRDeed.jl 109

APPENDIX B SUPPLEMENTARY MATERIAL 112

REFERENCES 123

VITAE 133

LIST OF TABLES

Table 3.1 Comparing The Applying of Fixed Direction with Random Vertex to The Outer Approximation Algorithms and The Enhanced Approximation Algorithms 38

Table 4.1 SG-DR Variables, Parameters, Functions 46

Table 4.2 Apply the Algorithm to the SG 50

Table 5.1 Comparing Results Scenario 1 65

Table 5.2 Comparing results Scenario 2 68

Table 6.1 Variable and Parameter of SG 75

Table 6.2 Various weighting factor values 85

Table 6.3 Comparison of Loss Improvement between DR-DEED and SGSD-DEED Models 87

Table 6.4 Comparison of Cost Improvement between DR-DEED Model and SGSD-DEED Model 88

Table 6.5 Comparison of Emissions Improvement between DR-DEED and SGSD-DEED Models 90

Table 6.6 Comparison of Power Generated Improvement between DR-DEED Model and SGSD-DEED Model 91

Table 6.7 Comparing results Scenario 1 102

Table B.1 Compares the Results of the Work by Rem Nur Keskin and the Enhanced Outer Approximation in Mathematics Example 113

Table B.2 Results of Experiment 2. 114

Table B.3 Customer cost coefficient (Test System 1). 115
Table B.4 Customer cost coefficient (Test System 2). 115
Table B.5 Optimal results with various weighting factor values (Test System 1). 115
Table B.6 Optimal results with various weighting factor values (Test System 2). 115
Table B.7 Comparison of costs, emissions, power generated, and loss (Test System 2). 115
Table B.8 The Optimal Customer Power Curtailed (Test System 1) 116
Table B.9 Optimal Customer Incentive(Test System 1) 117
Table B.10 Optimal Power Generated by the Customer $p_{i,t}$ (Test System 1) . 118
Table B.11 Optimal Power Stored by the Customer $s_{i,t}$ (Test System 1) . . . 119
Table B.12 Optimal Demand to be Shifted by the Customer $h_{i,t}$ (Test System 1) 120
Table B.13 Optimal Demand Satisfied by the Customer $f_{i,t}$ (Test System 1) 121
Table B.14 Optimal Power Generated by Generators q_j, t (Test System 1) . . 122

LIST OF FIGURES

Figure 6.1 Optimal power curtailed for all customers (Scenario 1) 85
Figure 6.2 Total load profile before and after SGSD-DEED model (Scenario 1) 86
Figure 6.3 Comparison of Loss Improvement between DR-DEED and SGSD-DEED Models 87
Figure 6.4 Comparison of Cost Improvement between DR-DEED Model and SGSD-DEED Model 89
Figure 6.5 Comparison of Emissions Improvement between DR-DEED and SGSD-DEED Model 90
Figure 6.6 Comparison of Power Generated Improvement between DR-DEED Model and SGSD-DEED Model 91
Figure 6.7 Cost, $\mathcal{C}$, for different generators 94
Figure 6.8 Emission, $\mathcal{E}$, for different generators 95
Figure 6.9 Utility, $\mathcal{U}$, for different generators 96
Figure 6.10 Power Loss 97
Figure 6.11 Power Load Reduction for different generators 98

LIST OF ABBREVIATIONS

Adj	Adjacent vertices-based approach
Cl	clusters-vertex
CVOP	Convex vector optimization problem
DED	Dynamic Economic Dispatch
DEED	Dynamic Economic emission Dispatch
DR	Demand Response
DRM	Demand Response Management
DSM	the Demand Side Management
FD	Fixed-Direction
FP	Fixed-Point Direction
Id	Ideal-vertex
In	Inner-vertex
IP	Ideal-Point Direction
LVOP	Linear vector optimization problem
MOP	multi-objective optimization problem
PS	Pascoletti-Serafini scalarization

R	Random-vertex
SG	Smart Grid
SGSD-DEED	Smart-Grid Shifted Demand Deed Model
VOP	Vector optimization problem

DISSERTATION ABSTRACT

NAME:	NORAH ABDULLATIF MOHAMMED ALMURAYSIL
TITLE OF STUDY:	Approximation Methods of Vector Optimization with Application To the Demand-Response Model of Smart Grids
MAJOR FIELD:	MATHEMATICS
DATE OF DEGREE:	SEPTEMBER 2024

This thesis develops an outer approximation algorithm, which is one way to solve vector optimization problem(VOP) with a specific emphasis on convex vector optimization problems. The proposed improvements include enhanced scalarization techniques and direction selection methods which are demonstrated to be effective through mathematical examples. The enhanced outer approximation algorithm is then applied to various real-world applications, such as Multi-Periodic Smart Grid Demand Response Problem(SG), the Dynamic Economic Emission Dispatch (DEED), and Integrating DEED with Demand Response Programs(DR-DEED). The performance of the algorithm is evaluated by comparing the results with the existing result. The enhanced algorithm is expected to yield significant performance improvements, reaching more than 50% improvement in lowering the cost,power generated and the power generated

in some cases. Furthermore, this thesis introduces a new model for economic problems known as the Demand Response Dynamic Economic Emission Dispatch (DR-DEED) of Electric Power Generation problem. This model incorporates advanced constraint structures derived from Smart Grid Demand Response Problem. The enhanced outer approximation algorithm is also applied to this problem formulation. The proposed techniques and their successful application to real-world problems demonstrate the significant potential of these methods to address a wide range of critical decision-making challenges.

ملخص الرسالة

الاسم: نوره عبداللطيف محمد المريسل

عنوان الدراسة: طرق التقريب لتحسين المتجهات مع التطبيق على نموذج الاستجابة للطلب في الشبكات الذكية

التخصص: قسم الطلاب

تاريخ الدرجة العلمية: سبتمبر-2024

تطور هذه الأطروحة خوارزمية التقريب الخارجي، وهي إحدى الطرق لحل مشاكل تحسين المتجهات مع تركيز خاص على مشاكل تحسين المتجهات المحدبة. تشمل التحسينات المقترحة تركيزًا على مجالين رئيسيين: تقنيات التماثل والتي تقوم بتحويل مشكلة تحسين المتجه إلى مشكلة عددية؛ مما يسهل العثور على الحلول. أما بالنسبة للمجال الآخر، فهو عن طرق اختيار الإتجاه والتي تم إثبات فعاليتها من خلال أمثلة رياضية. ومن الأهداف الرئيسية لهذه الاطروحة، تطبيق خوارزمية التقريب الخارجي المحسنة على تطبيقات واقعية متعددة مثل: مشكلة استجابة الطلب الذكي مع برامج التوزيع الديناميكي للإقتصاد في انبعاثات توليد الطاقة الكهربائية و نموذج نظرية الألعاب لمشكلة استجابة الطلب في الشبكة الذكية متعددة الفترات. يتم تقييم أداء الخوارزمية من خلال مقارنة نتائجها مع النتائج المتاحة. علاوة على ذلك، تقدم هذه الأطروحة نموذجًا جديدًا للمشكلات الإقتصادية يعرف بمشكلة التوزيع الديناميكي للإقتصاد والإنبعاثات لإستجابة الطلب في الشبكات الذكية. كما يتم تطبيق خوارزمية التقريب الخارجي المحسنة على نمذجة هذه المشكلة. تُظهر التقنيات المقترحة وتطبيقاتها الناجحة على المشكلات الواقعية الإمكانيات الكبيرة لهذه الطرق في معالجة مجموعة واسعة من تحديات إتخاذ القرار.

CHAPTER 1

INTRODUCTION

An optimization problem where a multiple vector-valued conflicting objective function is minimized or maximized is known as Vector optimization problem (VOP). In VOP, we define the objective functions with respect to a partial order relation induced by an ordering cone C rather than comparing the objective function values component-wise as in the multi-objective optimization problem (MOP).

The ordering cone C plays a crucial role in VOP, as it determines the partial order used to compare vectors. If we consider the ordering cone to be the non-negative orthant, our problem will be considered as a MOP, which can be regarded as a specific case of VOP. This relationship highlights the generality of VOP and its ability to encompass a broader range of optimization scenarios.

Researchers have studied VOP for decades, leading to the development of various methods to solve or approximate them. These methods can be broadly categorized into two main approaches. The first approach focuses on finding solutions in the decision space that cannot be improved in any objective without degrading at least one other objective. These solutions are known as efficient solutions [1]. The second

approach involves mapping the efficient solutions to the objective space, resulting in what are known as non-dominated points. The set of all non-dominated points forms the non-dominated set or Pareto front [2]. This approach, generating non-dominated points, has several advantages: It provides a visual representation of the trade-offs among various objectives, which can be especially helpful for decision-makers in understanding the problem's structure. In many cases, the dimension of the objective space is lower than that of the decision space, simplifying the analysis and clarifying the results.

However, in real-life applications such as engineering design, finance, and economics, obtaining exact solutions directly can be quite challenging due to the complexity of many problems. The interaction of numerous variables, constraints, and objectives often leads to computationally intensive scenarios that make precise optimization impractical. As a consequence, researchers often focus on approximation algorithms that provide sufficiently good solutions while requiring less time and resources for implementation. Our focus will center on these approximation methods, exploring their various strategies and effectiveness in addressing complex VOP. Before examining the specifics of these approximation techniques, we will first provide essential definitions and concepts related to VOP to establish a solid foundation for understanding the challenges and strategies involved.

1.1 Preliminaries

In this section, we outline the notations and definitions that will be used throughout the work. Let $\mathbb{R}^q$ be the q-dimensional Euclidean space, where q is a positive integer. $\|.\|$ denotes the Euclidean norm on $\mathbb{R}^q$ and the dual of this norm denotes by $\|.\|_*$. We denote the non-negative orthant of $\mathbb{R}^q$ by $\mathbb{R}^q_+$. Let K be a nonempty convex subset of $\mathbb{R}^q$, and denote by $\text{int}(K)$, $\text{cl}(K)$, $\text{bd}(K)$, $\text{conv}(K)$, $\text{cone}(K)$, the *interior*, *closure*, *boundary*, *convex hull*, *conic hull* of K, respectively.

A *recession direction* of K is a nonzero vector $r \in \mathbb{R}^q$ satisfying $K + [0, \infty)r \subset K$. The set of all recession directions of K, $\text{rece}K = \{r \in \mathbb{R}^q : K + \lambda r \subset K, \text{ for all } \lambda \geq 0\}$, is a cone called the *recession cone* of K. For a nonzero vector $r \in \mathbb{R}^q$ and a scalar s, the set $\mathcal{H}_{r,s} = \{y \in \mathbb{R}^q : r^T y = s\}$ is called a *hyperplane*. $\mathcal{H}_{r,s}$ is called a *supporting hyperplane* of K if $K \cap \mathcal{H}_{r,s} \neq \emptyset$ and $K \subseteq \mathcal{H}^+_{r,s} = \{y \in \mathbb{R}^q : r^T y \geq s\}$. The set K is a *polyhedral* convex set if it can be written as $K = \bigcap_{i=1}^k \mathcal{H}^+_{r_i,s_i}$ for some $0 \neq r_i \in \mathbb{R}^q, s_i \in \mathbb{R}$ and for $i \in \{1, \cdots, k\}$. This representation of K is called the *halfspace representation* ($H-$ representation).

The polyhedral convex set K can also be represented as $K = \text{conv}\{v_1, v_2, \cdots, v_m\} + \text{cone}\{r_1, r_2, \cdots, r_n\}$, where m is a positive integer, n is a non-negative integer, each v_i is a nonzero vector in $\mathbb{R}^q$ and each $r_j \in \text{rece}K$. This representation is called the *vertex representation* ($V-$ representation) of K.

A convex subset $S \subset K$ is called a *face* of K if for $x, y \in S$ whenever $\lambda x + (1 - \lambda)y \in S$ with $x, y \in K$ and $\lambda \in (0, 1)$. A *vertex* or (an *extreme point*) is a zero-dimensional face. A one-dimensional face is however called an *edge* of K. In

polyhedral convex set K, two vertices are said to be *adjacent* if the line segment between them is an edge of K.

The vertex enumeration problem is the problem of computing the vertices and extreme directions of a polyhedral convex set K. To accomplish this, we utilize the Bensolve tools [3], [4]. Bensolve tools also yield the set of all adjacent vertices of K.

Let $S \subseteq \mathbb{R}^q$ be another subset of $\mathbb{R}^q$. The Minkowski sum of K and S is $K + S := \{k + s \in \mathbb{R}^q \mid k \in K, s \in S\}$. Additionally, for $\mu \in \mathbb{R}$, we have $\mu K := \{\mu k \in \mathbb{R}^q \mid k \in K\}$.

Definition 1.1 *Let $K, S \subseteq \mathbb{R}^q$. The Hausdorff distance between K and S is defined as*

$$d_H(K, S) := \max \left\{ \sup_{k \in K} \inf_{s \in S} \|k - s\|, \sup_{s \in S} \inf_{k \in K} \|k - s\| \right\},$$

where $\|.\|$ is the euclidean norm in $\mathbb{R}^q$.

Let $C \subset \mathbb{R}^q$ be a nonempty closed convex cone. The cone C is non-trivial if $\{0\} \subsetneq C \subsetneq \mathbb{R}^q$, pointed if $C \cap (-C) = \{0\}$ (i.e it does not contain any lines) and it is said to be solid if $\text{int}C \neq \phi$. The dual cone of C is defined as $C^+ := \{r \in \mathbb{R}^q | \forall c \in C : r^T c \geq 0\}$ and it is a closed convex cone. The following definitions are well-known concepts in the VOP based on the nonempty convex pointed cone $C \subset \mathbb{R}^q$ with the partial order relation $\leq_C$ on it. If the convex cone C is nontrivial and pointed, we can define a partial order relation over C which is defined as follows: for all $y^1, y^2 \in \mathbb{R}^q, y^1 \leq_C y^2$ holds if and only if $y^2 - y^1 \in C$.

Definition 1.2 ([5]) *A function $\Omega : \mathbb{R}^n \to \mathbb{R}^q$ is said to be C-convex if $\Omega(\lambda x + (1 - \lambda)y) \leq_C \lambda\Omega(x) + (1 - \lambda)\Omega(y)$ for all $x, y \in \mathbb{R}^n, \lambda \in [0, 1]$.*

Definition 1.3 ([5]) *For a set $K \subset \mathbb{R}^q$, a point $k \in K$ is a C-minimal element of K if $(\{k\} - C \setminus \{0\}) \cap K = \phi$. If C has a nonempty interior and $(\{k\} - intC) \cap K = \phi$, then $k \in K$ is called a weakly C-minimal element of K. The set of all C-minimal (weakly C-minimal) elements of K is denoted by $Min_C K (wMin_C K)$.*

The VOP can be defined mathematically as :

$$\begin{aligned} &\min\ \Omega(x) \text{with respect to} \leq_C \\ &\text{subject to } x \in \Psi, \end{aligned} \tag{P}$$

where $\Omega : \mathbb{R}^n \to \mathbb{R}^q$ is a continuous C-function that composed of q real-valued objective functions, given by $\Omega := (\zeta_1(x), \ldots, \zeta_q(x))^T$, $\zeta_i(x) : \mathbb{R}^n \to \overline{\mathbb{R}}, i = 1, 2, \ldots, q$. The set C is a solid pointed polyhedral cone $\subseteq \mathbb{R}^q$. The set $\Psi \subset \mathbb{R}^n$ is a nonempty closed convex set representing the feasible region. The image of the feasible region $\Omega(\Psi) := \{\Omega(x) \in \mathbb{R}^q : x \in \Psi\}$.

The solutions to VOP are known as Pareto-optimal solutions, also referred to as efficient or non-dominated solutions. These form a fundamental concept in multi-objective optimization, including VOPs. In a VOP context, where multiple conflicting objectives need simultaneous optimization, a Pareto-optimal solution is defined as follows:

Definition 1.4 *A feasible solution $\psi^* \in \Psi$ is efficient (weakly efficient) if there does not exist any other feasible solution $\psi \in \Psi$ such that $\Omega(\psi) \leq \Omega(\psi^*)$ $((\Omega(\psi) < \Omega(\psi^*)))$. If ψ^* is efficient, then $\Omega(\psi^*)$ is a n non-dominated point (weakly non-dominated point). If $\psi, \psi' \in \Psi$ are such that $\Omega(\psi) \leq \Omega(\psi')$ we say that ψ dominates ψ' and $\Omega(\psi)$*

dominates $\Omega(\psi')$. *Feasible solutions* $\psi, \psi' \in \Psi$ *are equivalent if* $\Omega(\psi) = \Omega(\psi')$.

The projection of such point ψ^* in the objective space, (i.e.$\Omega(\psi^*)$, is called a Pareto Point, and the projection of the Pareto-optimal solutions set is called the Pareto Front[6].

Definition 1.5 ([7], [8]) *A point* $\bar{\psi} \in \Psi$ *is said to be a (weak) minimizer for* (P)*if* $\Omega(\bar{\psi})$ *is a (weakly) C-minimal element of* $\Omega(\Psi)$.

Approximation methods have become powerful techniques for addressing the challenges associated with solving VOP. While the exact computation of the complete set of Pareto-optimal solutions is desirable, it can be computationally intractable for many real-world VOP, especially those with complex objective functions and constraints. These methods aim to balance between solution quality and computational efficiency.

In the upcoming chapter, **chapter (2)**, we will explore and analyze two significant classes of VOP: Linear vector optimization problem (LVOP) and Convex vector optimization problem (CVOP). We will provide a comprehensive literature review of the algorithms that have been employed to solve these problems. Our primary focus will be on the approximation methods for solving the problems particularly the CVOP class of VOP. Additionally, in **chapter (3)**, we will improve the outer approximation algorithms for CVOP. This improvement will include enhancements to the (PS) scalarization and one of the direction selection methods. Moreover, mathematical examples will be examined for the new algorithm. Following that, in **chapter (4-5)**, we will apply the improved algorithm to various real-life applications and compare the

results. Finally, in **chapter(6)**, we will present a new model for economic problems known as the DR-DEED (Demand Response Dynamic Economic Emission Dispatch of Electric Power Generation) problem. This new model will incorporate the constraint structures derived from a smart grid demand response game theory. The integration of these advanced concepts is expected to yield significant performance improvements of more than 50%.

CHAPTER 2

LINEAR AND CONVEX VECTOR OPTIMIZATION

2.1 Linear Vector Optimization

LVOP is an extension of linear optimization where the linear real-valued objective function and the decision variables are vectors rather than scalars, and the feasible region is a polyhedron [9].

Researchers have been investigating vector objective linear programming since the 1960s, and multiple methods exist to address LVOP. Some methods operate in the decision variable space to find the efficient set, such as simplex methods and interior point methods. In the simplex method, researchers use an initial feasible solution as a starting point and iteratively improve it by moving along the edges of a polyhedral feasible region until an optimal solution is reached. This method has been developed by several researchers[10], [11], [12], [13]. In 2007, a primal-dual simplex algorithm was developed for multicriteria linear programming by Ehrgott, Puerto, and Rodriguez-

Chía [14]. However, solving multiple objective linear programming problems MOP in the decision space has presented several challenges. As a consequence, researchers have decided to concentrate on the objective space. For example, Dauer and Liu in [15], introduced a method for identifying the non-dominated extreme points and edges within the objective space.

2.2 Approximation Algorithm of LVOP

Approximation methods have become powerful techniques to address the challenges associated with solving VOP. Due to the challenging nature of precisely determining the complete non-dominated set, which is often difficult or in some cases mathematically impossible or computationally expensive, researchers have directed their attention toward the development of approximation algorithms. One of the key advantages of approximation methods is their ability to handle a wide range of problem complexities, from linear and convex problems to more challenging nonlinear and discrete problems. When the solution set is numerically computable, such as in linear or convex VOP, these methods can fully characterize the Pareto-optimal solutions, according to Ruzika and Wiecek [16]. In cases where some, but not all, of the efficient or Pareto-optimal points are numerically computable, such as in nonlinear VOP, the approximation approaches are employed to approximate the solution set. Furthermore, when the optimal solutions are not easily computable, such as in discrete VOP, approximation techniques help to estimate the solution set. In this way, decision-makers can gain valuable insights into the problem and make informed decisions without

requiring the whole solution set to be characterized.

It is widely acknowledged that the work of H.P. Benson [17], [18] has significantly advanced the development of approximation algorithms for linear multi-objective optimization problems. Furthermore, Benson's approximation-based approaches have provided researchers with practical solution methodologies to more effectively characterize the Pareto-optimal frontier for LVOP [19]. In 1998, he focused on generating the set of all efficient extreme points in the outcome set rather than in the decision set using the outer-approximation algorithm [17], [18]. After that, Ehrgott *et al.*[20] represented a dual variant of Benson's outer approximation algorithm based on geometric duality theory. Many modifications and adaptations of Benson's algorithm have been developed to achieve the intended outcome including bounded and unbounded problems. In 2013, researchers presented an enhanced version of Benson's algorithm, that significantly improved efficiency. Rather than solving two or three linear programs in each iteration, the enhanced algorithm only requires solving a single scalar linear program [21], [22].

The fundamental concept behind Benson's algorithm is to iteratively generate solutions that approximate the Pareto frontier. Following that, the objective function values of each solution are evaluated, and dominance is determined. In the current approximation of the Pareto frontier, dominated solutions are removed and non-dominated solutions are updated. During each iteration, solutions are selected from the non-dominated set, and new solutions are generated through specific algorithmic operations. This main algorithm has undergone several improvements over time to enhance several aspects such as efficiency, scalability, constraint handling ca-

pabilities, and solution quality. In 2007, Shao, Lizhen, and Ehrgott [23], proposed an enhanced version of the Benson approximation algorithm that incorporates inner and outer approximation techniques. This variant seeks to improve the accuracy and quality of the approximation to the Pareto frontier in LVOP.

2.3 Convex Vector Optimization

A particular subclass of MOP problems is the convex vector optimization problem CVOP, which is characterized by the convexity of the objective functions and the feasible region. Solving a convex vector optimization problem CVOP is known to be more challenging than solving LVOP. Consequently, approximation methods are actively being developed by researchers that aim to find inner and/or outer approximations of the efficient frontier rather than seeking the exact solution. In [24], Ruzika and Wiecek wrote a survey paper that classifies the approximation methods for a CVOP. For bounded convex vector optimization problems, several outer approximation algorithms have been developed. In 2011, Ehrgott et al. [25] extended the LVOP algorithm to solve the CVOP, motivated by Benson's arguments. Then, in 2014, Löhne *et al.*[26] worked to improve it by introducing a new algorithm that solves fewer optimization problems in each iteration.

Moreover, researchers have been working to improve the methods used to select the reference point for scalarization, which are the vertex selection and the direction, using different types of scalarization methods. Some researchers have utilized a Pascoletti-Serafini scalarization (PS) which requires a reference point and a direction vector as

its parameters [25], [26]. Recently, in 2021, Dörfler *et al.*[27] proposed an algorithm for CVOP includes a vertex selection procedure with PS scalarization. Moreover, in 2023, Irem Nur Keskin and Firdevs Ulus [28] proposed new methods for outer approximation algorithms to select the parameters used in the PS approach. Others, such as Klamroth *et al.*[29], used a Gauge-based scalarization model. Also, in 2022, Ararat et al.[9] used norm-minimizing scalarizations with outer approximation algorithm for CVOP which requires only a reference point and does not need a direction parameter. The reference point is selected among the vertices of the current outer approximation. Notice that, the methods in [29], [27] require solving additional models to select a vertex at each iteration. With this feature, they can display the current approximation error at each iteration of the algorithm without needing to solve multiple models.

The primary concept of the approximation methods involves evaluating the upper image of the problem (P) which is defined by adding the ordering cone C to the image of the feasible set $\Omega(\Psi)$. It contains all non-dominated points in its boundary and is defined as $\mathcal{U} := cl(\Omega(\Psi) + C)$. The Pareto front of the original optimization problem (P) corresponds exactly to the set of minimal elements of the upper image $\mathcal{U}$ when considering the partial order induced by the ordering cone C. The connection between the Pareto front and the upper image is a powerful theoretical result that forms the foundation for many approximation methods through the use of different algorithms that generate sequences of polyhedra in VOP. Some algorithms used inner approximations to approximate the upper image, while other algorithms used outer approximations. Moreover, some algorithms combine both inner and outer polyhedral approximations. These methods work by finding better approximations of the upper

image and stopping when the approximation is sufficiently fine. In the algorithms that use inner polyhedral approximations, lower bounds to the feasible region are provided, as well as outer polyhedral approximations enclose the feasible region to give upper bounds on the optimal solution. On the other hand, the use of both inner and outer polyhedral approximations allows the optimization algorithm to systematically narrow the gap between the lower and upper bounds. The choice of approximation method often depends on the specific problem structure, the number of objectives, the nature of the feasible set, and the desired properties of the approximation like accuracy, computational efficiency, etc...

2.3.1 Outer Approximation Algorithms

In this subsection, we will focus on the outer approximation algorithm that works by constructing a series of polyhedral sets that progressively get closer to the true upper image $\mathcal{U}$. This approach was recently described as a general framework in [28]. First, we will outline some assumptions that will be considered in our work.

Assumption 2.1 *The following assumptions about (P) hold.*

1. *The vector objective function $\Omega : \mathbb{R}^n \to \mathbb{R}^q$ is continuous and C-convex.*

2. *The feasible region Ψ is a compact subset of $\mathbb{R}^n$ with non-empty interior.*

3. *The ordering cone C is a polyhedral closed convex cone that is also solid, pointed, and nontrivial*

This assumption implies that problem (P) is bounded and a solution to (P) exists.

Indeed, since the set Ψ is compact and the function Ω is continuous, the image $\Omega(\Psi)$ is also compact and, therefore, bounded. This means there exists a point $b \in \mathbb{R}^q$ and a positive radius r such that the open ball $B(b,r)$ contains $\Omega(\Psi)$. Moreover, for the vector $c \in intC$, there exists a positive radius s such that the open ball $B(b,s) \subset b - c + C$. By setting $y = b - (\frac{r}{s}) * c$, we can show that the set $y + C$ contains the open ball $B(b,r)$. This is because the vector y is obtained by subtracting a scaled version ofc from the point b, and then the set $y + C$ covers the entire ball $B(b,r)$. Since the ball $B(b,r)$ contains the set $\Omega(\Psi)$, we have $\Omega(\Psi) \subset B(b,r) \subset y + C$. This, we have shown that there exists a point y such that the set $y+C$ includes the entire image $\Omega(\Psi)$. Moreover, the assumption guarantees the existence of solutions and this is by Proposition 4.2 in [26].

For bounded convex vector optimization problems, two solution concepts for weak minimizers are derived from a set-optimization perspective. [19].

Definition 2.1 ([26], Definition 3.3) *: Let $c \in int\, C$ be fixed. For $\epsilon > 0$, a nonempty finite set $\overline{\Psi} \subseteq \Psi$ of (weak) minimizers is a finite (weak) ϵ-solution with respect to c if conv $\Omega(\overline{\Psi}) + C - \epsilon\{c\} \supseteq \mathcal{U}$.*

Definition 2.2 ([9], [27]) *: For $\epsilon > 0$, a nonempty finite seta nonempty finite set $\overline{\Psi} \subseteq \Psi$ of (weak) minimizers is a finite (weak) ϵ-solution if conv $\Omega(\overline{\Psi}) + C + B(0,\epsilon) \supseteq \mathcal{U}$.*

Both of these solution concepts provide an outer approximation to the upper image.

The general steps of the algorithm begin with an initial polyhedral outer approximation that encompasses the upper image. The main idea underlying most of these

algorithmic approaches is to iteratively generate a sequence of polyhedral approximations that successively improve the representation of the upper image [9]. The first step in these approximation methods is to generate a set of (weak) minimizers for the original problem (P). One common technique for producing these minimizers is to solve a series of scalarization models, which will be defined in more detail in the following. This process yields an initial set of weak minimizers, denoted as Ψ^o. To refine the outer polyhedral approximation, the algorithm identifies supporting hyperplanes using the solutions obtained from the scalarization sub-problems. Incorporating these new supporting hyperplanes into the outer polyhedral approximation improves the representation of the upper image. To check for convergence, we can evaluate the upper bound provided by the current outer approximation. If the change in the upper bound from the previous iteration is smaller than a specified tolerance, stop, otherwise, return to the scalarization subproblems steps and repeat the process. Once the change in the upper bound is sufficiently small, the algorithm terminates and outputs the approximate optimal solution provided by the final outer approximation.

2.4 Scalarization

Scalarization methods, which involve transforming MOP problem into a single-objective problem, were introduced in the 1970s as a solution process and have since become one of the most commonly used methods for solving these problems, either precisely or approximately [30]. Each Pareto optimal solution is computed by categorizing scalarization methods that modify either the constraints or the objective

function of the original MOP in each sub-problem. Additional parameters, auxiliary points, or variables may also be used as part of the process. The scalarized optimization problems arising from scalarization techniques are subsequently solved using methods from single-objective optimization, which are specifically designed to optimize a single objective function.

Researchers have used different types of Scalarization methods to find the minimizers in their algorithms.

2.4.1 The Weighted Sum Scalarization

As a first step, most papers employ the well-known method known as weighted sum scalarization. The process involves assigning weights to each objective and forms an aggregated objective function by summing the weighted objectives. The relative importance or priority of each objective is represented by its weight. the scalarized problem for a given weight vector w is written as follows:

$$\min_{x \in \Psi} \sum_{i=1}^{q} w_i \zeta_i(x) \tag{WS(w)}$$

In general, the weight vectors are the generating vectors of the dual cone C^+. By considering different weight vectors w, a set of weak minimizers can be generated. These weak minimizers from the scalarization sub-problems are then used to identify supporting hyperplanes that refine the outer polyhedral approximation of the upper image, as described earlier. The iterative process of solving scalarization models and updating the outer approximation continues until a satisfactory approximation is

achieved.

Proposition 2.1 ([31], Corollary 5.29) *Let $w \in C^+ \setminus \{0\}$. An optimal solution x^w of $(WS(w))$ is a weak minimizer of $\mathcal{U}$. Conversely, for any weak minimizer $x \in \Psi$, there exists $w \in C^+ \setminus \{0\}$ such that x is an optimal solution of $(WS(w))$.*

The dual program of $(WS(w))$ is define using the Lagrangian

$$L_w : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, \qquad L_w(x, u) := w^T \Omega(x) + u^T \rho(x)$$

The dual program is defined as

$$\max\{\varphi_w(u), u \in \mathbb{R}^m_+\}, \quad \text{where} \quad \varphi_w(u) = \inf_{x \in \mathbb{R}^n} L_w(x, u).$$

The primary advantage of this scalarization method is its simplicity and its ability to generate a diverse set of weak minimizer points. By varying the weight values assigned to the different objective functions, this approach can produce a wide range of candidate minimizer points. This allows for a more comprehensive exploration of the Pareto optimal set, as different weight combinations can lead to various points along the Pareto front. The flexibility to adjust the weight values enables the algorithm to systematically identify a diverse set of potential minimizer solutions, which is particularly useful when dealing with complex MOP problems.

2.4.2 Pascoletti-Serafini Scalarization

In each iteration, to find a new set of weak minimizers, we can use different types of scalarization. For example, we can use the scalarization known as Pascoletti-Serafini scalarization (PS) which is defined as

$$\begin{aligned} &\min z \\ &\text{subject to} \quad \Omega(x) \leq v + zd, \\ &\qquad\qquad\quad x \in \Psi, z \in \mathbb{R} \end{aligned} \tag{PS(v,d)}$$

The parameter $v \in \mathbb{R}^q$ is the reference point and $d \in \mathbb{R}^q$ is the direction.

The Lagrange dual of the PS problem can be written as

$$\begin{aligned} &\max \qquad \left[\inf_{x\in\Psi} w^T\Omega(x)\right] - w^T v \\ &\text{subject to} \\ &\qquad\qquad w^T d = 1, w \in C^+ \end{aligned} \tag{D-PS(v,d)}$$

This scalarization method is capable of finding all minimizers of a given MOP even if the problem is not convex. The inclusion of the reference point v in this scalarization allows for greater flexibility in shaping the approximation of the Pareto front. On the other hand, The direction vector d provides additional control over the regions of the Pareto front that are explored. By varying these points, the $PS(v,d)$ can generate a diverse set of weak minimizers that cover different parts of the Pareto front. In the next section, we will discuss in more detail the direction and the reference point.

Proposition 2.2 ([26],Proposition 4.5) *If $(x^*, z^*) \in \mathbb{R}^{n+1}$ is an optimal solution*

of problem(PS(v,d))*, then* x^* *is a weak minimizer. Moreover,* $v + z^*d \in \mathrm{bd}\,\mathcal{U}$*.*

Remark 2.1 $\Omega(x) \leq_C v + zd$ *holds for some* $x \in \Psi$ *if and only if* $v + zd \in \mathcal{U}$*.*

Notice that, the notation $\leq_C$ indicates that the difference $v + zd - \Omega(x)$ belongs to the convex set C. So, if we assume this direction holds for some $x \in \Psi$, then $v+zd-\Omega(x) \in C$. Thus, $v+zd \in \Omega(x)+C \subseteq \Omega(\Psi)+C = \mathcal{U}$. Conversely, if $v+zd \in \mathcal{U}$, there exists a point $\Omega(x)$ such that $v+zd \in \Omega(x)+C$. Then, $v+zd-\Omega(x) \in C$ which leading to $\Omega(x) \leq_C v + zd$ holds for some $x \in \Psi$.

2.4.3 The Norm-Minimizing Scalarization

We can use the norm-minimizing scalarization at each iteration, which defines a fixed arbitrary norm $\|.\|$ on $\mathbb{R}^q$ and a point $v \in \mathbb{R}^q$ as scalarization of (P) given by

$$\begin{aligned} \min \quad & \|\kappa\| \\ \text{subject to} \quad & \Omega(x) - \kappa - v \leq 0, \\ & x \in \Psi, \kappa \in \mathbb{R} \end{aligned} \tag{NS(v)}$$

For the Lagrange dual of $NS(v)$ problem, we need first to define an equivalent formulation by defining a scalar function $S : \mathbb{R}^n \times \mathbb{R}^q \to \overline{\mathbb{R}}$ and a set-valued function $B : \mathbb{R}^n \times \mathbb{R}^q \rightrightarrows \mathbb{R}^q$

$$S(x,\kappa) := \|\kappa\| + I_\Psi(x); \qquad B(x,\kappa) := \{\Omega(x) - \kappa - v\}; \quad x \in \mathbb{R}^n; \kappa \in \mathbb{R}^q.$$

Then $NS(v)$ is equivalent to :

$$\min S(x,\kappa) \;\text{ subject to }\; B(x,\kappa)\cap -C \neq \phi, (x,\kappa)\in\mathbb{R}^q\times\mathbb{R}^q. \qquad \text{(NŚ)}$$

For convex programming with set-valued constraints and by the results from ([19], Section 3.3.1) and ([32] , Section 8.3.2), we define the Lagrangian $L:\mathbb{R}^n\times\mathbb{R}^q\to\overline{\mathbb{R}}$ for NS' by

$$L(x,\kappa,w) := S(x,\kappa) + \inf_{u\in B(x,\kappa)+C} \quad w^{Tu}; (x,\kappa,w)\in\mathbb{R}^n\times\mathbb{R}^q\times\mathbb{R}^q.$$

Then, the dual objective function $\varphi:\mathbb{R}^q\to\overline{\mathbb{R}}$ is defined by

$$\varphi(w) = \inf_{x\in\mathbb{R}^n,\kappa\in\mathbb{R}^q} \quad L_w(x,\kappa,w), w\in\mathbb{R}^q.$$

By the definitions of S, B and using the fact that $\inf_{c\in C}\; w^Tc = -I_C + (w)$ for every $w\in\mathbb{R}^q$ we obtain

$$\varphi(w) = \begin{cases} \inf_{x\in\Psi,\kappa\in\mathbb{R}^q} \quad \Big(\|\kappa\| w^T(\Omega(x)-\kappa-v)\Big) & \text{if} \qquad w\in C^+, \\ -\infty & \text{otherwise.} \end{cases}$$

As a result, the dual of NS' is

$$\max\varphi(w) \;\text{ subject to } w\in\mathbb{R}^q. \qquad \text{(D-NS}(v))$$

In the PS and the norm scalarization methods, the reference point (or vertex) v is obtained from the outer polyhedral approximation. However, in [26], [27], they

consider another type of scalarization that uses the reference point v that does typically not belong to $\mathcal{U}$,

$$\begin{aligned} \min \quad & \kappa \\ \text{subject to} \quad & g(x) \leq 0 \\ & \Gamma^T(\Omega(x) - v - \kappa d) \leq 0, \end{aligned} \tag{Pr(v,c)}$$

where Γ is the matrix, whose columns $\kappa_1, \ldots, \kappa_j$ are the generating vectors of the dual cone C^+ of the ordering cone C.

The Lagrangian dual problem is given as:

$$\begin{aligned} \max \inf_{x \in \Psi} \{u^T g(x) + w^T \Omega(x)\} - w^T v \\ s.t. u \geq 0, \\ w^T c = 1, \\ w \in C^+. \end{aligned} \tag{D-Pr(v,c)}$$

Proposition 2.3 ([27], Proposition 3.7) *Let the assumptions hold and let $p \in$ int $\mathcal{U}$. Then for every $v \in \mathbb{R}^q \setminus \mathcal{U}$ and $c := p - v$, solutions $(x^*, \kappa^*)$and (u^*, w^*) to Pr(v,c) and D-Pr(v,c) ,respectively, exist and their optimal values coincide.*

Proposition 2.4 ([27], Proposition 3.8) *Let (x^*, κ^*)be a solution to Pr(v,c). Then x^* is a weak minimizer of $\mathcal{U}$ and $y := v + \kappa^* c$ is a weakly C-minimal element of $\mathcal{U}$.*

CHAPTER 3

OUTER APPROXIMATION ALGORITHMS FOR CVOP

As mentioned in Chapter(1), the outer approximation algorithms have emerged as a powerful class of techniques for solving CVOP problems. The algorithms work by constructing a sequence of outer approximations to the feasible set and the Pareto front until an acceptable solution is obtained. At each iteration, they solve a scalarized subproblem to generate new efficient points and refine the approximation. The process continues until a desired level of accuracy is achieved. The scalarization that builds upon outer approximation algorithms is PS method. The solutions to these scalarized sub-problems generate new efficient points. With each new efficient point, the approximation of the Pareto front is refined, gradually improving the accuracy of the representation.

By improving this underlying scalarization approach, we aim to achieve more efficient and accurate solutions. Existing research literature has introduced various

methods for direction and vertex selection to choose the reference point for the PS scalarization. In the following, we will provide a brief overview of these existing methods and introduce a new direction selection method. Additionally, we will propose an improved PS formulation that reduces the number of scalarized subproblems required at each iteration of the outer approximation algorithm. The effectiveness of these advancements will be demonstrated through comprehensive computational studies.

3.1 Direction

There are many ways to choose a direction based on the scalarization methods used at each iteration, the kinds of cones, and other factors. The first way to choose it is **Fixed-Direction (FD)** method. In this approach, the direction is chosen as a fixed vector from the interior of the ordering cone, and this FD is used throughout the algorithm. For example, in [26], researchers selected the direction as an interior point of the cone such that $\|d\| = 1$ with the PS scalarization. FD provides a consistent directional reference for generating new efficient points.

Some researchers used another direction approach known as the **Fixed-Point Direction (FP)**. In this FP method, the direction can be a fixed point from the upper image, as in [25], [29]. Alternatively, others utilize the point from the inner approximation that yields the minimum distance to the selected vertex, as described in [27]. Additionally, the direction can be chosen as the difference between a point outside the upper image in $\mathbb{R}^q$ and the interior point of the upper image[25]. Thus, for a fixed point $\hat{u}$ in the interior of the upper image and for every $v \in \mathbb{R}^q$, $d = \hat{u} - v$

Taking inspiration from these results, İrem Nur Keskin and Firdevs Ulus, in [28], proposed two new selection rules for determining the direction parameter. The first one is **Ideal-Point Direction (IP)**. This approach is based on the ideal point, denoted by $\mathbf{y}^I$, which is the point that minimizes each objective function individually, defined as follows:

$$\mathbf{y}^I := \inf\{\Omega_i(x) | x \in \Psi\} \ for \ i = 1, \ldots, q.$$

It serves as a reference point that indicates the perfect outcome for the optimization problem. Assume that the ordering cone is $\mathbb{R}^q_+$, hence the ideal point $\mathbf{y}^I$ is well defined. As an initial step, We have a vertex v of the current outer approximation U^k. Then, for this vector, we compute the vector $v - \mathbf{y}^I$. This vector points from the current vertex v towards the ideal point $\mathbf{y}^I$, providing information about the relative position of the current solution. Notice that, in the initial iteration, we find that $(v - \mathbf{y}^I = 0)$. In the subsequent iterations, we have $(v - \mathbf{y}^I \geq 0)$ since U^k is always a subset of U^0 at any stage of the algorithm. For each component $i \in 1, \ldots, q$, we compute $d_i = \frac{1}{\mathbf{v_i} - \mathbf{y_i}^I + \bar{\epsilon}}$ for some sufficiently small $\bar{\epsilon} > 0$, which is added for computational convenience (avoiding division by zero). Then d is normalized such that $\|d\| = 1$. Geometrically, the normalized direction vector d can be interpreted as the normal vector of the hyperplane passing through the points $(v_i - \mathbf{y_i}^I)\mathbf{e}_i$ for each $i \in 1, 2, ..., q$.

The second direction proposed is **Adjacent vertices-based approach (Adj)**. This approach begins with the use of the double description method to solve the vertex enumeration problem. The double description method a is a straightforward and

effective algorithm for enumerating all extreme rays [33]. This process not only computes the vertices of a convex polytope cone but also yields a set of adjacent vertices for each vertex involved. For each vertex v, the normal direction of a hyperplane passing through its adjacent vertices is computed. This normal vector is critical for understanding how the geometric properties of the convex hull influence the optimization process. The primary motivation for using adjacent vertices stems from their relationship to the curvature of the upper image of the problem. The positions of these adjacent vertices arise from supporting hyperplanes, which closely relate to the current vertex. Adjacent vertices can sometimes include extreme directions. If an adjacent extreme direction $\mathbf{z}$ is identified for a vertex v, it implies that moving along this direction yields a line segment $\{v + r\mathbf{z} \in \mathbb{R}^q \mid r \geq 0\}$. In scenarios where extreme directions are detected, an artificial vertex is created by modifying the current vertex$(v+\mathbf{z})$. Notice that, for each vertex, we might have more adjacent vertices than required. Therefore, to find an appropriate direction we will choose q of the vertices where$q$ is the dimension of the space $\mathbb{R}^q$. The goal is to compute a direction vector $d \in \mathbb{R}^q$ such that:

$$d^\top v_{\text{adj}}^{i_1} = d^\top v_{\text{adj}}^{i_2} = \ldots = d^\top v_{\text{adj}}^{i_q},$$

Notably, solving this system may yield d or $-d$ as potential solutions. In this step, we need to check if $d \in \operatorname{int} C$ or $-d \in \operatorname{int} C$. By proposition 3.6 in [28], if $d \in \operatorname{int} C$, we obtain a weak minimizer solving PS. If either d or $-d$ qualifies, a predetermined direction is utilized as $d = \frac{\sum_{i=1}^{r} c^i}{\|\sum_{i=1}^{r} c^i\|}$, where $c^1, \ldots, c^r$ correspond to the extreme direction of C.

3.2 Vertex Selection

As the performance of the algorithm is also affected by vertex selection, researchers have introduced several vertex selection rules and methods. For some algorithms, they do not solve additional models to select a vertex at each iteration and that depends on the **Random-vertex (R) selection**. For example, in [26] for a vertex v of U^k, a point y on the boundary of the upper image, which is in 'minimum distance' to v, is determined. Note that $y = \Omega(x) + c$ for some $x \in \Psi$, and $c \in C$. If the minimum distance is less than or equal to an error level $\epsilon > 0$, which is determined by the user, the algorithm proceeds to check another vertex of U^k similarly. In [9], they define the set V_k as the set of vertices of the current outer approximation U^k. Then, for each $v \in V^k$, if not done before, the norm-minimizing scalarization and its dual are solved using this vertex v as the reference point.

From another point of view, other researchers believe that solving additional models to select a vertex can yield better results in the least possible time. In [27], they select the vertex according to the distance to the inner approximation. We can call this method **Inner-vertex (In) selection**.In the algorithm, they computed a decreasing sequence of polyhedral outer approximations and growing a sequence of polyhedral inner approximations of the upper image $\mathcal{U}$ which is achieved by iteratively cutting off vertices v of outer approximation while introducing new half-spaces. For the iteration k, for every v in the set of vertices of the outer, the Euclidean distance to the inner approximation is computed. Then, a bi-level optimization problem. A solution to it is a vertex of the outer approximation that yields the shortest distance to the

current inner approximation. Another way to select a vertex that is Closest to the Ideal Point is Known as **Ideal-vertex (Id) selection**, which is used if the ordering cone is $\mathbb{R}^p_+$. The vertex that we will use need to satisfy that $\|v - \mathbf{y}^I\|$ is the minimum, where $v \in V^k$, and $\mathbf{y}^I$ is the ideal point. Moreover, we can use the method known as Farthest to an Inner Point (In). For the fixed point $p^* \in \mathcal{U}$ that define as $p_i^* := 2\max\{\zeta_i(x_1), \ldots, \zeta_i(x_n)\} - v_i^0$, we choose $v \in V^k \setminus V_{used}$ which yields the greatest distance $\|p^* - v\|$.

Recently, in 2023, Keskin and Ulus [28] proposed different vertex selection rules that can be used with their algorithm that uses (PS) scalarization for each iteration k. Specifically, their goal is to propose vertex selection rules that are less computationally complex and potentially improve the algorithm's efficiency. One of these methods is the **clusters-vertex (Cl) selection**. This technique organizes vertices of the current outer approximation into groups, or clusters, to ensure a comprehensive and balanced exploration. The process begins with establishing the centers for the clusters through an iterative procedure. After solving $(PS(v, d))$ for each vertex of the initial polyhedron U^0, which helps define the supporting halfspaces, we create a new polyhedron U^1 by intersecting these halfspaces with U^0. This procedure is repeated for U^1, leading to the identification of vertices from U^2 as the centers of the clusters. It is important to note that one could choose to use vertices from U^1 to create fewer clusters, or continue using vertices from higher iterations U^k (where $k > 2$) for a greater number of clusters. Notice that, before computing the set of vertices V^{k+1} of U^k for $k \in \{0, 1\}$, the pair (V^0, U^0) must be called to initialize the cluster centers. In the subsequent iterations, each vertex of the current outer approximation is assigned

to the cluster whose center is closest, based on Euclidean distance. The selection process then proceeds as follows: First of all, the clusters are visited sequentially. When a vertex needs to be selected for exploration, an arbitrary vertex is chosen from the current cluster. If all vertices in that cluster have already been explored, the algorithm will then move on to the next nonempty cluster.

3.3 Enhanced Outer Approximation Algorithms

In this section, we will discuss two main improvements that have been made to the approximation algorithm introduced in [28]. The first improvement focuses on the scalarization methods used within the algorithm. As previously mentioned, scalarization is a crucial step in converting the MOP problem into a single-objective form that can be efficiently solved. This work enhances the Pascoletti-Serafini scalarization technique, which is known for its effectiveness in managing the balance between competing objectives. In addition to the improvements in scalarization, we will introduce novel direction selection methods inspired by the adjacent vertices approach, aiming to enhance exploration and improve the optimization process.

3.3.1 Improvement of Pascoletti-Serafini Scalarization.

The Lagrange dual of the Pascoletti-Serafini scalarization leads to a max-min optimization problem, which can be computationally challenging to solve directly. This is because the max-min formulation involves an inner minimization problem that must be solved for each iteration of the outer maximization problem. To address this prob-

lem, we employ Relaxation Methods. We have introduced a new variable $t \in \mathbb{R}$ as a lower bound for $w^T\Omega(x)$, where w are the scalarization weights and $\Omega(x)$ are the objective functions. This transformation allows us to reformulate the max-min optimization problem as a maximization problem with additional inequality constraints involving t. This reformulation makes the optimization problem more computationally tractable. These enhancements to the Pascoletti-Serafini scalarization are expected to lead to a more accurate and robust approximation of the Pareto-optimal solutions, ultimately improving the overall performance of the algorithm.

The Lagrange dual of the PS problem can be written as

$$\begin{aligned} \max \quad & t - w^T v, \\ \text{subject to} \quad & w^T\Omega(x) \geq t, \\ & w^T d = 1, w \in C^+. \end{aligned} \tag{ND-PS(v,d)}$$

Under the assumptions of the problem (P) and some additional conditions on the problem parameters, the following proposition demonstrates the existence of an optimal solution to the problem (PS(v,d)).

Proposition 3.1 *Let $v \in \mathbb{R}^q$. If $d \in \operatorname{int} C$, then there exist optimal solutions to PS(v, d)) and (ND-PS(v, d)). Moreover, the optimal values of the two problems coincide.*

Proof. For the existence of a feasible solution for (PS(v,d)), we start with the assumption that there exists $\bar{x} \in \operatorname{int}\Psi$($\Psi$ is a compact set with nonempty interior). Let define $\bar{z} := \max_{i\in 1,\dots,l} \frac{(w^\nu)^\top(\Omega(\bar{x})-v)}{(w^i)^\top d} + \delta$ for some $\delta > 0$, where $C^+ = \operatorname{co}\operatorname{conv}\{w^1,\dots,w^l\}$.

By the condition in d ($d \in \operatorname{int} C$), $\bar{z}$ is well defined. Moreover, it follows that

$(w^i)^\top d > 0$ for all i. This ensures that the denominator in the expression is positive. Now, The tuple $(\bar{x}, \bar{z})$ is confirmed to be a feasible solution for (PS(v, d)).

Indeed, Since $\bar{z}$ is defined strictly greater than a certain lower bound, it is a Slater point(strictly feasible point), thus satisfying the necessary conditions for optimality.

For the existence of an optimal solution for (ND-PS(v, d)), from the Slater condition, it follows that there exists an optimal solution w^* for (ND-PS(v, d)). Moreover, the optimal values of (PS(v, d)) and (D-PS(v, d)) coincide.

For the existence of an optimal solution for (PS(v, d)): Let (x, z) be any feasible solution in $\mathbb{R}^{n+1}$. Our goal is to show that z is bounded below. The compactness of Ψ implies (P) is bounded. Therefore, there exists $a \in \mathbb{R}^p$ such that $\mathcal{U} \subseteq a + C$. This relationship demonstrates that $v + zd \in a + C$ holds for any feasible (x, z) by remark 2.1. Therefore, $w^\top(v + zd - a) \geq 0$ (ie. $w^\top(v - a) + zw^\top d \geq 0$). Since $w^\top d > 0$, we can divide both sides by it and get $z \geq -\frac{w^\top(v-a)}{w^\top d}$. To satisfy this inequality for all $w \in C^+ \setminus \{0\}$, we must ensure that z is at least as large as the maximum value that the right side can take:

$$z \geq \sup_{w \in C^+ \setminus \{0\}} -\frac{w^\top(v-a)}{w^\top d}.$$

$$z \geq \sup_{w \in C^+ \setminus \{0\}} \frac{w^\top(a-v)}{w^\top d} = \sup \left\{ \frac{w^\top(a-v)}{w^\top d} \middle| \, w \in C^+, \|w\| = 1 \right\} =: \tilde{z} \in \mathbb{R}.$$

Then, without changing the problem, we may add the constraint: $\tilde{z} \leq z \leq \bar{z}$ to (PS(v, d)). This makes the feasible region of (PS(v, d)) compact. In conclusion: as

the feasible region of the modified (PS(v, d)) is compact, it has an optimal solution. ∎

3.3.2 Approximate (Adj) Direction Approach

Inspired by the adjacent vertices-based approach introduced in [28], we propose additional direction selection methods. Given the challenges associated with calculating adjacent vertices, these methods aim to enhance the exploration of selection rules for the direction parameter used in PS scalarization. By building on the concepts of adjacent vertices, we strive to identify more effective strategies to improve the optimization process.

When the algorithm selects a vertex v from the set of vertices V_k that define the current outer approximation U^k, it must determine a direction for solving the Pascoletti-Serafini scalarization. One potential approach is to use the adjacent vertices of v as the basis for this direction. However, in our proposed method, instead of relying on the adjacent vertices, we select the set of vertices that are closest to v to define the direction. This alternative approach offers several advantages. It allows the algorithm to better capture the local curvature and shape of the Pareto front near v. Furthermore, the adjacent vertices may not always provide the most appropriate information for local optimization, as they can be too distant from v to accurately reflect the local characteristics of the Pareto front. Our proposed method is also not constrained by the specific structure of vertex adjacency, making it more applicable to a broader range of PS problems.

Approximate (Adj) direction
1: Compute the set of vertices V^k of U^k;
2: Pick $v_r \in V^k \backslash V_{used}$;
3: Compute $\|v_i - v_r\|$, where $v_i \in V^k \backslash V_{\text{used}}$;
4: Pick q vertices $v_r^{i_1}, \ldots, v_r^{i_q}$ where they are the closest to v_r and let $A = \left[v_r^{i_1}, \ldots, v_r^{i_q}\right] \in \mathbb{R}^{q \times q}$;
5: Compute $d \in B(0,1)$ satisfying $A^\top d = e \left(d = \frac{A^{\top -1} e}{\|A^{\top -1} e\|} \text{ if } A^{-1} \text{ exists}\right)$;
6: if $d \in \operatorname{int} C$ then return d
else
if $-d \in \operatorname{int} C$ then return $-d$
else
return $d = \frac{\sum_{i=1}^r c^i}{\|\sum_{i=1}^r c^i\|}$,
where $c^1, \ldots, c^r$ are the extreme directions of C.
end if.

The proposed approach of utilizing the closest vertices to determine the search direction presents an intriguing and computationally efficient alternative to existing methods. Moreover, since this method follows the general framework of the adjacent vertices approach but eliminates the need to calculate the adjacent vertices, its effectiveness is demonstrated by proposition 3.6 in [28].

Algorithm 1: Enhanced Outer Approximation Algorithm (P)

1: Initialize

$V_{\text{used}} \leftarrow \emptyset$ $k \leftarrow 0$, solve $\leftarrow$ True;

2: For $i = 1, \ldots, \eta$: Solve $(WS(w^j))$ to find an optimal solution x^i;

3: Set $\overline{\Psi}^o = \{x^1, \ldots, x^\eta\}$ and

$U^0 = \bigcap_{i \in (1,\ldots,\eta)} \left\{ y \in \mathbb{R}^p \mid (w^i)^\top y \geq (w^i)^\top \Omega (x^i) \right\}$;

4: Compute the set of vertices V^0 of U^0;

5: Fix rules for selecting a vertex from V^k and a direction parameter d;

6: while $V^k \backslash V_{\text{used}} \neq \emptyset$ do,

Solve ($\text{PS}(v, d^v))$ / $(\text{ND} - \text{PS}(v, d^v))$.

Let (x^v, z^v) and w^v be optimal solutions;

7: $V_{\text{used}} \leftarrow V_{\text{used}} \cup \{v\}, \overline{\Psi}^k \leftarrow \overline{\Psi}^k \cup \{x^v\}$;

8: if $z^v > \epsilon$ then

9: Compute $\Im$ such that

$\Im = \left\{ y \in \mathbb{R}^q \mid (w^v)^\top y \geq (w^v)^\top v + z^v \right\}$;

10: $p^{k+1} \leftarrow P^k \cap \Im, \overline{\Psi}^{k+1} \leftarrow \overline{\Psi}^k$;

11: Compute the set of vertices V^{k+1} of P^{k+1};

12: $k \leftarrow k + 1$;

13: end if

$\overline{\Psi}^k$

Proposition 3.2 *When terminates, Algorithm 1 returns a finite weak ϵ-solution.*

Proof. We begin by establishing a foundational framework for our algorithm. As an initial Step, optimal solutions x^i exist for $(WS(w^i))$ for all $i \in \{1, \ldots, l\}$ due to Ψ being compact and Ω being continuous (by the Extreme Value Theorem). For each i, we define a halfspace

$$\Im^i = \{y \in \mathbb{R}^q \mid (w^i)^T y \geq (w^i)^T \Omega(x^i)\}.$$

This halfspace contains the the upper image $\mathcal{U} = \Omega(\Psi) + C$

Indeed:

To proof that $\Im^i \supseteq \mathcal{U} = \Omega(\Psi) + C$:

We have x^i is an optimal solution to $(WS(w^i))$, implying that $\forall x \in \Psi$: $(w^i)^T \Omega(x^i) \leq (w^i)^T \Omega(x)$. Additionally, since $\inf_{c \in C} (w^i)^T c = 0$, it follows that $\forall c \in C : (w^i)^T c \geq 0$. Now, let $y \in \mathcal{U}$. Then $\exists\ x \in \Psi, c \in C$ such that $y = \Omega(x) + c$. This leads to the expression $(w^i)^T y = (w^i)^T (\Omega(x) + c) = (w^i)^T \Omega(x) + (w^i)^T c$. By optimality of x^i, we have $(w^i)^T \Omega(x) \geq (w^i)^T \Omega(x^i)$. Moreover, by property of C, $(w^i)^T c \geq 0$. Therefore, $(w^i)^T y \geq (w^i)^T \Omega(x^i)$. We conclude that, y satisfies the inequality defining $\Im^i$, hence, $y \in \Im^i$. As y was arbitrary, $\mathcal{U} \subseteq \Im^i$.

In step(3), U^0, the initial outer approximation, is defined as the intersection of all $\Im^i$ Consequently, $U^0 \supseteq \mathcal{U}$, ($U^0$ contains the upper image). For the existence of vertices of U^0, since C is pointed (it has a vertex at the origin) and (P) is bounded, it follows from Corollary 18.5.3 in [34], U^0 contains at least one vertex. Therefore, we have $V^0 \neq \emptyset$ (which implies that $V^0 \setminus V_{\text{used}} \neq \emptyset$), where V^0 is the set of vertices of U^0. For the initial solution set($\overline{\Psi}^o = \{x^1, \ldots, x^l\}$), each x^i is a weak minimizers and that

by proposition (2.1) .

The main loop for the iteration process k consists of solving the primal problem of PS along with its enhanced dual of it to obtain the solution (x^v, z^v) and weight w^v as long as there is $v \in V^k \backslash V_{\text{used}}$. The algorithm terminates when when $V^k \backslash V_{\text{used}} = \emptyset$. If this occurs after K iterations, then $V^K \subset V_{\text{used}}$, and the corresponding z^k exists for all $v \in V^k$, with $z^k \leq \epsilon$. For this step, it is necessary to establish rules for defining a direction parameter d and selecting a vertex v (if not predetermined). Notice that the solution exists by proposition (3.1) and x^v is a weak minimizer by proposition (2.2). Thus, for any iteration k, $\overline{\Psi}^k$ is finite and includes only the weak minimizers. The next step is to update the sets V_{used} by adding v to it and $\overline{\Psi}^k$ by adding x^v.

By proposition 4.7 in [34], $\Im = \{y \in \mathbb{R}^q | (w^k)^T y \geq (w^k)^T v + z^k\} \supseteq \mathcal{U}$ (step 9). Now since $U^0 \supseteq \mathcal{U}$ and $U^{k+1} = U^k \cap \Im$ for all $k \geq 0$, we have $U^k \supseteq \mathcal{U}$ for all $k \geq 0$.

To show $\overline{\Psi}^k$ is a finite weak ϵ-solution of P, we will use the definition from [9], [27]. The definition states that: for $\epsilon > 0$, a nonempty finite set $\overline{\Psi} \subseteq \Psi$ of (weak) minimizers is a finite (weak) ϵ-solution if *conv* $\Omega(\overline{\Psi}) + C + B(0, \epsilon) \supseteq \mathcal{U}$.

Since we have $U^k \supseteq \mathcal{U}$ for all $k \geq 0$, in the sense of the definition, it is sufficient to show *conv* $\Omega(\overline{\Psi}) + C + B(0, \epsilon) \supseteq U^k$. By lemma 2.5 in [9], we have the recession cone of U^k is C for any k. Moreover, since the recession cone C includes all feasible directions such that if $x \in U^K$ and $d \in C$, then $x + t \cdot d \in U^k$ for all $t \geq 0$. The convex hull of the set of vertices V^k of U^k, conv(V^K), is defined as:

$$\text{conv}(V^K) = \left\{ \sum_i \lambda_i v_i \mid v_i \in V^K, \lambda_i \geq 0, \sum \lambda_i = 1 \right\}.$$

Therefore, the outer approximation U^k can be represented as: $U^k = conv(V^k) + C$. This implies that, for any $u \in U^k$, there exist $(\lambda^v)v \in V^k \geq 0$ and $\overline{c} \in C$ such that $u = \sum_{v \in V^k} \lambda^v v + \overline{c}$.

On the other hand, for each $v \in V^k$, there exists $x^v \in \overline{\Psi}^k, z^v \leq \epsilon$ such that (x^v, z^v) is an optimal solution of $(PS(v, d^v))$. In particular, there exists $c^v \in C$ such that $v + d^v z^v = \Omega(x^v) + c^v$. Therefore,

$$
\begin{aligned}
u = \sum_{v \in V^k} \lambda^v v + \overline{c} &= \sum_{v \in V^k} \lambda^v (\Omega(x^v) + c^v - d^v z^v) + \overline{c} \\
&= \sum_{v \in V^k} \lambda^v (\Omega(x^v)) + \sum_{v \in V^k} \lambda^v c^v - \overline{c} - \sum_{v \in V^k} \lambda^v d^v z^v.
\end{aligned}
$$

It is clear that $\sum_{v \in V^k} \lambda^v (\Omega(x^v)) \in conv\ \Omega(\overline{\Psi})$ and $\sum_{v \in V^k} \lambda^v c^v - \overline{c} \in C$. Also, given that $z^v \leq \epsilon$ and $\|d^v\| \leq 1$ for every $v \in V^k$, leads to the conclusion $\sum_{v \in V^k} \lambda^v d^v z^v \in B(0, \epsilon)$. Therefore, we have shown that $conv\ \Omega(\overline{\Psi}) + C + B(0, \epsilon) \supseteq U^k$. ∎

Examples

We will run the algorithm for four direction selection rules: Fixed Direction(FD), Fixed Point Direction (FP), Ideal Point Direction (IP), Approximate Adj Direction (AP-Adj), and 4 vertex selection rules: Random Vertex(Rn), Ideal Point Vertex (Id), Inner Point Vertex (In), Cluster Vertex (Cl).

The results are shown in table 3.1, which shows the number of scalarization models (SC) and the CPU time (T).

Example 3.1 $: \Omega(x) = x, \Psi = \{x \in \mathbb{R}^n \mid \|x - e\| \leq 1, x \geq 0\}$ *for* $, n \in \{2, 3, 4\}$.

Example 3.2 : $\Omega(x) = \left(x, \frac{1}{x}\right)^\top, \Psi = \mathbb{R}_+$.

Example 3.3 : $\Omega(x) = x, \Psi = \left\{x \in \mathbb{R}^3 \,\middle|\, \left(\frac{x_1-1}{1}\right)^2 + \left(\frac{x_2-1}{a}\right)^2 + \left(\frac{x_3-1}{5}\right)^2 \leq 1\right\}$

for $a \in \{5, 7, 10, 20\}$.

Example 3.4 : $\Omega(x) = x, \Psi = \left\{x \in \mathbb{R}^4 \,\middle|\, \left(\frac{x_1-1}{1}\right)^2 + \left(\frac{x_2-1}{a}\right)^2 + \left(\frac{x_3-1}{5}\right)^2 + \left(\frac{x_4-1}{1}\right)^2 \leq 1\right\}$

for $a \in \{5, 7, 10\}$.

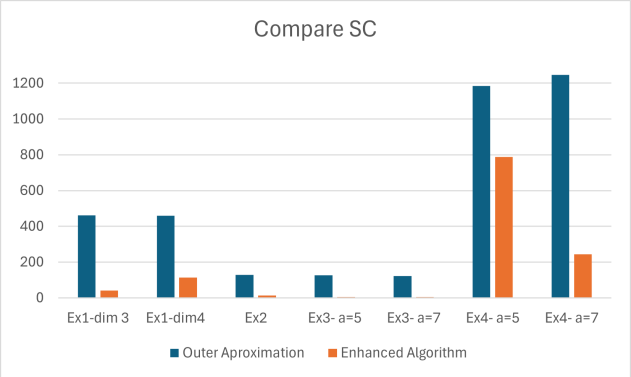
All experiments were implemented using Julia as the programming language [35] (version 1.10) and JuMP [36] as the modeling language, with Ipopt solver [37] on a laptop which features a 13th Gen Intel(R) Core(TM) i9-13900H processor running at a speed of 2.60 GHz. The laptop is equipped with 16.0 GB of installed RAM, providing 15.7 GB of usable memory. The system operates on a 64-bit operating system with an x64-based processor. We developed a package for the algorithm. This package was published on GitHub in this link .https://github.com/mmogib/OAFCVX.jl, and it contains all the implementations of the algorithms. For further details, we will illustrate how to use this package with example 3.1 which outlines the necessary steps, including defining the function, the direction, PS scalarization, and the weighted sum scalarization, with the corresponding code included in the appendix (A.1).

Applying the enhanced approximation algorithm to the examples listed before has resulted in a reduced number of scalarizations solved throughout the algorithm (SC) and the CPU time (T), with Fixed direction and Random vertex selection. The improvement in most of the examples was seen with the random and cluster vertex selection. Table B.1 compares the results of the work by Rem Nur Keskin and Firdevs Ulus in [28] and our work. The enhanced approximation algorithm was able to achieve

a substantial decrease in the number of scalarizations solved and the overall CPU time required, indicating its effectiveness in improving the efficiency of the optimization process. The random vertex selection strategies and the fixed direction approach played a crucial role in driving the performance improvements observed across the diverse set of examples examined, as evidenced by the results presented in table 3.1.

Example	Algorithm	direction	Vertex	SC	T
Example1-Dimension 3	Outer approximation algorithms	(FD)	(Rn)	461	127.60
	Enhanced approximation algorithms	(FD)	(Rn)	41	0.625
Example1-Dimension 4	Outer approximation algorithms	(FD)	(Rn)	460	153.96
	Enhanced approximation algorithms	(FD)	(Rn)	94.4	1.001
Example2-$\epsilon = 0.005$	Outer approximation algorithms	(FD)	(Rn)	127.8	27.64
	Enhanced approximation algorithms	(FD)	(Rn)	14	0.858
Example3- a=5,$\epsilon = 0.05$	Outer approximation algorithms	(FD)	(Rn)	125.8	33.56
	Enhanced approximation algorithms	(FD)	(Rn)	5	6.25
Example3- a=7,$\epsilon = 0.05$	Outer approximation algorithms	(FD)	(Rn)	123.2	32.77
	Enhanced approximation algorithms	(FD)	(Rn)	5	0.102
Example4- a=5,$\epsilon = 0.05$	Outer approximation algorithms	(FD)	(Rn)	1185	424.59
	Enhanced approximation algorithms	(FD)	(Rn)	787.2	274.368
Example4- a=7,$\epsilon = 0.05$	Outer approximation algorithms	(FD)	(Rn)	1247.4	450.86
	Enhanced approximation algorithms	(FD)	(Rn)	244	123.015

Table 3.1: Comparing The Applying of Fixed Direction with Random Vertex to The Outer Approximation Algorithms and The Enhanced Approximation Algorithms



Most notably, from table 3.1, the enhanced Pascoletti-Serafini approach dramati-

cally reduced the required number of scalarizations compared to the Outer approximation method. This represents a significant performance enhancement, as minimizing the number of scalarizations is critical for the efficiency and scalability of MOP techniques.

In the next chapter, we will apply the approximation algorithm to real-life applications to assess its effectiveness. This is an important step, as it will allow us to evaluate the practical performance of the algorithm and its potential for solving real-world problems.

Furthermore, we have observed a lack of real-world applications examined in existing research. This is an area that deserves more attention, as it could uncover new insights and lead to further improvements in the algorithm. By exploring a wider range of real-life scenarios, we can better understand the algorithm's capabilities, limitations, and the types of problems it is best suited to address. Examining real-life applications helps us identify any potential challenges or edge cases that may not have been covered in the theoretical analysis or the synthetic datasets used for testing. This information can then be used to refine the algorithm, develop new techniques, or identify areas for further research and development.

CHAPTER 4

APPLICATION: SMART GRID DEMAND RESPONSE PROBLEM

Approximation algorithms for CVOP are crucial tools that make these computationally complex optimization challenges more tractable and accessible in practice. Many real-world applications, such as in economics, finance, and engineering, involve multiple, often conflicting objectives, leading to computationally difficult CVOP. Approximation algorithms offer a solution to this problem by providing efficient methods to find high-quality optimal solutions. With this property, the ability to efficiently solve complex problems, and approximation methods can have significant real-world impact in various domains. The fundamental complexity of these problems arises from the need to find a set of optimal solutions that represent the best trade-offs between competing objectives. Traditional optimization methods can struggle to effi-

ciently solve these problems, particularly when the number of objectives is large or the problem size is large-scale with numerous variables and constraints. Approximation algorithms provide a powerful solution to this challenge by offering efficient methods to find high-quality solutions that are close to the true optimal solutions. Smart Grid (SG) optimization and Dynamic Economic emission Dispatch (DEED) are applications where approximation algorithms have demonstrated their value. These power systems problems involve balancing multiple, often conflicting objectives such as cost, emissions, and grid stability while accounting for the complex dynamics between electricity producers and consumers. The large-scale nature of these power networks, with numerous generators, consumers, and grid constraints, makes them inherently challenging to optimize using traditional methods. The ability of approximation methods to handle the computational complexity of these power systems applications highlights their crucial role in making such real-world optimization challenges more tractable and accessible. In the following sections, we will explore these applications in depth and demonstrate the key role of approximation techniques in addressing their computational complexity.

4.1 Smart Grid

There have been significant improvements in electronic communication technology since the early 21st century. As energy demand increased, smart grid technologies were developed which enabled home appliances to be controlled to save energy, reduce cost, and increase safety. A smart grid is an electrical grid that helps to transport

electricity from providers in order to satisfy the demands of the users by using digital technology and other cutting-edge technology to monitor and regulate the transmission of electricity from all generation sources. The main feature of the SG compared to the traditional grid is that the smart grid has technology that allows the two-way flow communication of electricity and information. To implement the concept of a smart grid and manage the significant energy output generated by renewable sources, we need to consider the Demand Side Management (DSM). DSM is a key feature of the smart grid that helps manage both the supply and demand more efficiently. In addition to reducing consumer's energy bills, DSM can assist energy systems by shifting low-priority demand from peak to non-peak hours.

Moreover, DSM involves managing and optimizing energy consumption on the customer side to reduce overall electricity demand during periods of high demand and energy prices. By implementing DSM strategies, utilities can decrease the peak load demand on their systems, which can help lower energy bills and improve the overall load demand profile.

A few years ago, game theoretical models were proposed by many authors for solving some problems that deal with SG, and the development of the SG is anticipated to necessitate the application of game theory. Game theory is a field of mathematics concerned with the study of conflict and cooperation. It provides a framework for analyzing strategic interactions among players, firms, and other entities. In game theory, interactive situations are represented and analyzed using a mathematical model. The concept applies whenever multiple agents, referred to as players, are involved in interdependent relationships. Game theory can be categorized into two main branches:

cooperative and non-cooperation games. Non-cooperative game theory focuses on the strategic decision-making processes of independent agents. A game's equilibrium, a fundamental concept for analyzing strategic interactions among players, is a key method for solving these games John Nash introduced the Nash equilibrium in [38], defining the solution of n-player non-cooperative games. It consists of strategies such that each player can choose one, under the condition that no player can unilaterally change their strategy to achieve a better outcome.

Many researchers have focused on developing SG technologies using game theoretic models such as Cui *et al.* in [39], presented two models of price determination for utility companies under different assumptions. They used a Nash equilibrium solution for the n-person game model[38] in their simple first model of the energy price competition and the solution was proven to be unique under some assumptions. Atzeni et al [40] [41] examined SG in which the demand-side combines traditional users with people who have distributed energy sources or energy storage devices. The smart grid DSM problem was formulated by Atzeni *et al.* as a noncooperative game. Instead of shifting energy consumption, a DSM method has been proposed which involves a day-ahead optimization process that corresponds to energy production and energy storage scheduling. In [42], Belhaiza *et al.*introduced a new theoretic model consisting of a noncooperative multi-agent game where users and providers interact to regulate a smart grid's demand. One critical aspect in the SG is Demand Response Management (DRM). It which encourages. consumers to modify their usage patterns and levels of electricity usage. By shifting low-priority demands from peak to off-peak hours, DRM can reduce consumers' energy bills and assist energy systems. In this model, SG game

proposes a Nash equilibrium which is a situation in which each player maximizes his or her benefit.

In other direction, some researchers are interested in the multi-periodic SG demand response problem. When the objectives of the model are optimized simultaneously, the demand response program can achieve a balance between customer needs and utility requirements while maximizing overall benefits. Numerous researchers have explored this aspect across various fields. Recently, Ullah *et al.*[43] presented an optimal tri-objective problem for the implementation of DSM strategies in SG with high wind energy penetration in the day-ahead scheduling problem. Additionally, Belhaiza *et al.*[44] proposed a game theoretic model for the multi-periodic SG demand side management problem with shifted demand. Their work involved a model based on game theory, specifically agent-based multi-periodic game theory. It is assumed that users rely on their individual utilities to determine whether they can shift their demand fully or partially, from one time period to another. Moreover, they assumed that users have the capability to generate their own energy, store it, or even sell it back to the energy providers.

4.2 Mathematical Formulation

To formulate the mathematical programming model, we need to define the following variables, parameters, and functions. Suppose there are n users in total (households) and p providers (energy companies or retailers). A non-cooperative game with $n+p$-person agents in which each agent aims to maximize their own utility was presented

under several conditions. For each user i, the following conditions are considered.

1. The total energy requested from all providers must not exceed the demand.

2. The average energy requested from each provider j must not exceed the energy supplied by j to i.

Simultaneously, for each provider j, the following conditions are considered.

1. The total energy supplied to all users must not exceed their production and delivery capacity.

2. The energy supplied by provider j to each user i must not exceed the average energy requested by i from j.

For the multi-period of time t, table (4.1) defines the main variable and parameter necessary for the SG game with shifted demand model(MPSGSD) . This model provides a framework for analyzing the strategic interactions and optimal decision-making of users and providers in a multi-period SG setting with the possibility of demand shifting across periods. At the Nash equilibrium [45] of this game, users and providers simultaneously maximize their individual utility functions across the T periods.

Variable	Type	Description
f_i^t	Real	Demand of i satisfied at t
h_i^t	Real	Demand of i shifted to t
p_i^t	Real	Quantity i produces at t
s_i^t	Real	Quantity i stores at the end of t
x_{ij}^t	Real	Quantity i sends to j at t
x_i^t	Vector	Quantities i sends to all providers at t
X_i^t	Vector	All decision variables controlled by i at t
p_j^t	Real	Quantity j produces at t
s_j^t	Real	Quantity j stores at the end of t
y_{ij}^t	Real	Quantity i receives from j at t
$y_{.j}^t$	Vector	Quantities i receives from all providers at t
Y_j^t	Vector	All decision variables controlled by j at t
Parameter		
D_i^t	Real	Demand of i at t
L_i^t	Real	Distribution capacity of i at t
L_j^t	Real	Distribution capacity of j at t
C_i^t	Real	Storage capacity of i at t
C_j^t	Real	Storage capacity of j at t
K_i^t	Real	Production capacity of i at t
K_j^t	Real	Production capacity of j at t
Function		
$U_{ij}^t\left(X_i^t, Y_j^t\right)$	Real	utility of i at t with respect to j
$V_{ij}^t\left(X_i^t, Y_j^t\right)$	Real	utility of j at t with respect to i

Table 4.1: SG-DR Variables, Parameters, Functions

$$(\mathcal{P}_i^t) : \max_{X_i^t} \sum_{t=1}^{T} \sum_{j=1}^{m} U_{ij}^t(X_i^t, Y_j^t) \tag{4.1}$$

subject to

$$f_i^t + s_i^t - s_i^{t-1} - p_i^t + \sum_{j=1}^{m} x_{ij}^t = \sum_{j=1}^{m} y_{ij}^t, \forall t \tag{4.2}$$

$$h_i^{t+1} + f_i^t = h_i^t + D_i^t, \forall t \tag{4.3}$$

$$\sum_{j=1}^{m} x_{ij}^t \leq L_i^t, \forall t \tag{4.4}$$

$$p_i^t \leq K_i^t, \forall t \tag{4.5}$$

$$s_i^t \leq C_i^t, \forall t \tag{4.6}$$

$$s_i^0 = 0, s_i^T = 0, h_i^1 = 0 \tag{4.7}$$

$$X_i^t \geq 0, \forall t \tag{4.8}$$

and

$$(\mathcal{P}_j^t) : \max_{Y_j^t} \sum_{t=1}^{T} \sum_{i=1}^{n} V_{ij}^t(X_i^t, Y_j^t) \tag{4.9}$$

subject to

$$s_j^t - s_j^{t-1} - p_j^t + \sum_{i=1}^{n} y_{ij}^t = \sum_{i=1}^{n} x_{ij}^t, \forall t \tag{4.10}$$

$$\sum_{i=1}^{n} y_{ij}^t \leq L_j^t, \forall t \tag{4.11}$$

$$p_j^t \leq K_j^t, \forall t \tag{4.12}$$

$$s_j^t \leq C_j^t, \forall t \tag{4.13}$$

$$s_j^0 = 0, s_j^T = 0 \tag{4.14}$$

$$Y_j^t \geq 0, \forall t \tag{4.15}$$

The authors used linear utility functions that were obtained for each agent in the SG game with shifted demand. The utility functions are defined as follows: For each user i, the utility function with respect to each provider j is such that

$$U_{ij}^t\left(X_i^t\right) = a_i^{tf} f_i^t + a_i^{ts} s_i^t + a_i^{tp} p_i^t + a_i^{th} h_i^{t+1} + a_{ij}^{tx} x_{ij}^t.$$

For each provider j, the utility function with respect to each user i is such that

$$V_{ij}^t\left(Y_i^t\right) = a_j^{ts} s_j^t + a_j^{tp} p_j^t + a_{ij}^{ty} y_{ij}^t,$$

where $a_i^{t\cdot}$ is the utility of variable (.) of user i at t, and $a_j^{t\cdot}$ is the utility of variable (.)

of provider j at t.

The researchers employed the concept of Nash equilibrium as the foundation for their model. In this game-theoretic framework, the n consumers and m grid operators are assumed to simultaneously optimize their individual objective functions over a series of T periods. The authors formalized this dynamic interaction using a primal-dual linear programming approach. The primal linear program captured the optimization problem faced by the users and providers, while the dual linear program represented the complementary slackness conditions.

In this work, the goal is to apply the enhanced approximation algorithm directly to the model. The objective is to efficiently solve the SG demand response game without the need for the full linear programming approach taken in the main work. By applying an enhanced approximation algorithm directly to the primary constraints, the proposed approach offers a more direct and potentially faster way to approximate the solutions to the problem. This allows for achieving results that are very close to the original solution while significantly improving computational efficiency.

The key innovation in this work is the development of a novel approximation algorithm that can be applied directly to the core constraints of the SG optimization problem. This alternative approach is expected to provide significant computational advantages while maintaining the accuracy of the original solutions. Our aim is to demonstrate the efficiency of this new approximation-based methodology compared to traditional linear programming techniques.

The experiment was for 3 users, 1 provider and 2 periods of time, with $\epsilon = 0.05$. We apply the algorithm with Fixed-Direction and Random-Vertex selection.

name	solution_1	solution_2	solution_3
U_1	17.44	17.14	16.15
U_2	15.23	15.24	14.82
U_3	15.70	15.27	15.70
V_1	28.16	28.29	27.96

Table 4.2: Apply the Algorithm to the SG

In table 4.2, the algorithm produced three different solutions, representing the vertices of a polyhedron that approximates the Pareto front. Notably, these three solutions can be used to derive any other solution through convex combinations of any two of them. Furthermore,, solution 1 with values of U_1, U_2, U_3 and V_1 are very close to the results reported in [44] which were $U_1 = 14.01, U_2 = 18.70$, and $U_3 = 14.21$. This suggests that the solution from [44] can indeed be obtained as a convex combination of the three solutions presented in the table.

In the following, we illustrate the algorithm using this model by considering costumers n with the values of $200, 400, 600, 1000$ and 2000, and providers m with $4, 6, 8, 10$ over a time period of $t = 2$. The table 4.2 shows the time required to obtain the result.

t	n	$m = 4$	$m = 6$	$m = 8$	$m = 10$
2	200	1.09	1.17	1.90	1.89
	400	1.66	2.35	3.55	3.33
	600	3.29	3.72	6.85	10.74
	1000	5.71	10.96	18.39	28.62
	2000	18.83	58.08	114.41	139.70

CHAPTER 5

APPLICATION: DYNAMIC ECONOMIC EMISSIONS DISPATCH (DEED)

5.1 DEED Problem

MOP has increasingly been applied to address a variety of real-world problems, particularly in the economic domain. Dynamic Economic Emissions Dispatch DEED is an optimization technique used in power systems to achieve a balance between electricity supply and demand. The main challenges of DEED arise when researchers attempt to incorporate emission dispatch into the Dynamic Economic Dispatch (DED) problem. DED is a mathematical problem that aims to identify the most efficient output for thermal generators to meet electricity demand while simultaneously managing emissions. The objective is to achieve a balance between cost-effective power generation

and environmental sustainability. DEED allows to integrate emissions within DED, while achieving cost-effectiveness and simplicity of implementation compared to other solutions [46].

Demand Response (DR) is widely recognized as an energy-saving measure [47] and a way to reduce electricity bills [48]. DR programs aim to actively manage and modify electricity consumption patterns by customers in response to changing grid conditions or price signals. The integration of DR with DEED enables the simultaneous optimization of demand-side load reduction and supply-side generator output. This integration not only ensures optimal operational efficiency but also enhances environmental sustainability across the electricity supply chain. Moreover, the DR-DEED model is designed to be scalable, and applicable to both micro-grids and extensive conventional grid systems, thus providing a versatile tool for improving grid stability and responsiveness. DR-DEED programs have been proposed to efficiently manage both the operation of thermal generators and the electricity consumption patterns of consumers [46].

The successful implementation of DEED strategies requires a supportive infrastructure that can effectively enable real-time monitoring, bidirectional communication, and intelligent control of electricity. This can be achieved by the SG, which represents an evolution of the electrical grid system. SG utilizes advanced digital and communication technologies to enhance the management of electricity flow from generation to consumption. This system is crucial for integrating diverse energy sources, particularly renewables, which introduce variability in power generation. To effectively harness these dynamic energy outputs, DRM is essential. As a pivotal component

of SG, DRM optimizes both supply and demand by enabling real-time adjustments based on grid conditions. This not only facilitates efficient energy use but also stabilizes the grid by shifting non-essential energy consumption from peak to off-peak periods. Through the implementation of DRM, utilities can significantly mitigate peak load demands, contributing to lower energy costs for consumers and a more balanced demand profile across the system. Enhanced by DRM, SG can ensure greater reliability, foster renewable integration, and promote active consumer participation in energy management.

5.2 Dynamic Economic Emission Dispatch Of Electric Power Generation

DED, DEED, and DR-DEED have all been addressed using a variety of solution algorithms, ranging from conventional mathematical approaches to advanced soft computing techniques. Research on the integration of DEED with DR is relatively scarce, with only a few studies primarily leveraging innovative soft computing algorithms [49], [50], [51], [52]. On the mathematical front, a notable study by Arif and Javed [53] utilized a genetic algorithm to integrate economic dispatch and demand-side management within a micro-grid context. Another significant approach by Saad *et al.* [46] combined DEED with a game theory-based DR program, accommodating multiple customer scenarios and various time intervals.

More recently, Chinnadurrai and Victore [54] tackled the DEED problem as a MOP issue within a SG framework. Their formulation incorporated wind power generation,

accounting for its variability within set upper and lower bounds. They introduced DR program that adjusts consumer electricity consumption in response to fluctuating electricity prices during different demand periods valley, peak, and off-peak.

5.2.1 Mathematical Formulation

Considering emissions in the formulation of the DED problem can be approached in several ways. The approach we consider in this work involves integrating both DED and the Pollution Dynamic Economic Dispatch (PDED) into a single problem. The objective of this combined problem is to minimize both fuel costs and emissions simultaneously. It is formulated as a MOP problem [55]. In this context, the optimization process seeks to find a trade-off between minimizing fuel consumption costs and reducing emissions. The objective is to identify a set of solutions that achieve an optimal trade-off between fuel cost and emissions while satisfying the defined network constraints. Let C_j^t and E_j^t represent the fuel costs and the emissions cost for generator j at time t, respectively. The problem can then be formulated mathematically as follows:

$$\min \sum_{t \in T} \sum_{j \in J} C_j^t \left(q_j^t\right) \tag{5.1}$$

$$\min \sum_{t \in T} \sum_{j \in J} E_j^t \left(q_j^t\right) \tag{5.2}$$

with

$$C_j^t\left(q_j^t\right) = a_j + b_j q_j^t + c_j (q_j^t)^2, \quad \forall j \in J,\ \forall t \in T, \tag{5.3}$$

$$E_j^t\left(q_j^t\right) = e_j + f_j q_j^t + g_j (q_j^t)^2, \quad \forall j \in J,\ \forall t \in T. \tag{5.4}$$

Symbol	**Type**	**Description**
q_j^t	Variable	Power generated from generator j at time t
a_j, b_j, c_j	Parameter	The fuel cost coefficients of generator j respectively.
e_j, f_j, g_j	Parameter	The emission cost coefficients of generator j respectively.
J and T	Set	The set generators and the dispatch time interval respectively.

subject to the following network constraints:

$$\sum_{j \in J} q_j^t = D_t + \text{loss}_t,\ \forall t \in T, \tag{5.5}$$

$$q_j^{\min} \leq q_j^t \leq q_j^{\max},\ \forall t \in T, \tag{5.6}$$

$$-DR_j \leq q_j^{t+1} - q_j^t \leq UR_j,\ \forall t \in T, \tag{5.7}$$

where

$$\text{loss}_t = \sum_{j \in J} \sum_{k \in K} q_j^t B_{j,k} q_k^t,\ \forall t \in T. \tag{5.8}$$

Symbol	Type	Description
D_t	Real	Total system demand at time t.
loss_t	Real	Total system losses at time t.
$q_j^{\min}$, $q_j^{\max}$	Real	Minimum and maximum capacity of generator i respectively.
DR_j, UR_j	Real	Maximum ramp down and up rates of generator j respectively.
$B_{j,k}$	Matrix	The jk^{th} element of the loss coefficient square matrix of size J.

The process of converting a MOP problem into a single objective function involves incorporating a weighting factor w. It is used within the weighted scalarization method defined as follows:

$$\min \left[w \left[\sum_{t \in T} \sum_{j \in J} C_j \left(q_j^t \right) \right] + (1 - w) \left[\sum_{t \in T} \sum_{j \in J} E_j \left(q_j^t \right) \right] \right] \tag{5.9}$$

Through the use of an approximation algorithm applied to DEED scenario involving six generators and five customers, the following results were obtained. The optimization problem considers four key variables:(C) representing the cost, (E) representing the emissions, (P) representing the power generated, and ($Loss$) representing the power loss, all of which will be analyzed and optimized through the approximation algorithm. The data for this scenario was taken from [46]:

direction type	Vertex selection type	C	E	P	$Loss$
(FD)	(Rn)	316830.82	28515.87	26264.44	307.07
	(Id)	318164.87	27153.33	26253.81	297.83
	(In)	315334.17	31543.72	26283.73	326.53
	(Cl)	315900.03	30014.20	26274.54	316.89
(FD)	(Rn)	316229.30	29299.70	26269.65	312.87
	(Id)	317134.01	28020.35	26260.64	304.79
	(In)	315245.23	31865.43	26285.47	328.62
	(Cl)	315612.18	30614.22	26278.05	321.03
(IP)	(Rn)	316371.35	29071.59	26268.16	311.43
	(Id)	317118.42	28016.66	26260.55	304.98
	(In)	315548.58	30840.12	26279.57	322.22
	(Cl)	316057.38	29636.28	26271.96	314.88
(AP-Adj)	(Rn)	319520.09	26908.69	26248.80	289.29
	(Id)	317685.54	27686.17	26257.37	299.42
	(In)	315621.92	30767.80	26278.98	320.46
	(Cl)	321422.97	26079.42	26239.06	282.42

The analysis of the provided table reveals that the performance of the different vertex selection approaches varies across the multiple objectives considered. In the Fixed-Direction approach, the Ideal-Point-Vertex method outperforms the other vertex selections, yielding the minimum values for several key metrics.

Specifically, the Ideal-Point-Vertex results in the lowest values for the objectives E(27153.33), P (26253.81), and $Loss$ (297.83). In contrast, the Inner-Point-Vertex method provides the minimum value for the C objective at 315334.17. However, the Ideal-Point-Vertex method maintains a marginally lower cost of 318164.87, which is approximately 0.90% higher than that of the Inner-Point-Vertex. Given the Ideal-Point-Vertex's demonstrable superiority across most key objectives and the relatively small cost difference, its approach emerges as the preferred vertex selection method for the Fixed-Direction.

Similar to the findings in the Fixed-Direction scenario, the analysis of the other vertex selection approaches, namely Fixed-Point Direction and Ideal-Point Direction, also shows the Ideal-Point-Vertex method as the best-performing option. In the Fixed-Point Direction category, the Ideal-Point-Vertex method demonstrates the minimum values for the key performance indicators, with E at $28,020.35$, P at $26,260.64$, and $Loss$ at 304.79.

Likewise, in the Ideal-Point Direction category, the Ideal-Point-Vertex method again provides the lowest values for E at 28016.66, P at 26260.55, and $Loss$ at 304.98.

Across these vertex selection approaches, the Ideal-Point-Vertex method emerges as the preferred choice, as it consistently delivers the best performance.

Finally, for the Approximate-Adj Direction, the Cluster-Vertex method demonstrates the minimum values for the key performance indicators, with E at $26,079.42$, P at $26,239.06$, and $Loss$ at 282.42. While the Inner-Point-Vertex method provides the minimum cost value of 315621.92, the Cluster-Vertex method has a cost of 321422.97, resulting in a difference of approximately 1.84% compared to the minimum cost. Based on the information provided, if we need to choose a single direction method and vertex selection approach that offers the overall best performance, the Approximate-Adj Direction using the Cluster-Vertex method would be the recommended choice.

5.2.2 A Game-Theoretic DR-DEED formulation

When developing demand response programs for electric utilities, a crucial requirement is to ensure that the monetary benefits provided to customers through incentives always outweigh the potential costs they may incur from service interruptions. Game

theory can be effectively employed to model the behavior of electricity consumers and market dynamics in demand response programs. By considering consumers' preferences, costs, and constraints, game theory enables us to understand their responses to price signals and incentives. The concept of demand contract formulations based on game theory was introduced and illustrated by Fahridoglu and Alvarado [56], which focused on a single customer scenario. Building upon this work, Nwulu and Xia [46] expanded the formulations of demand management contracts to incorporate multiple time intervals and integrated them into the Dynamic Economic Dispatch (DEED) problem.

By considering the following mathematical variables and parameters, we can define the Demand Response-DEED (DR-DEED) model:

Symbol	**Type**	**Description**
θ_i	Real	Customer type variable, normalized in the range $[0, 1]$. It represents the willingness of customers to participate, where 0 indicates no interest in participating and 1 indicates a high willingness to participate.
χ_i^t	Real	Power reduction by participating customer i at time t.
ω_i^t	Real	Monetary compensation received by customer i at time t.
λ_i^t	Matrix	Power interruptibility value, representing the cost of not supplying power to a specific location on the grid.
K_{1i}, K_{2i}	Real	The cost coefficients.

For more than one customer i and more than one period of time t, the cost incurred by a customer of type θ who decreases power consumption by χ MW represented by $c(\theta_i^t, \chi_i^t)$ can be defined as:

$$c(\theta_i, \chi_i^t) = K_{1i}(\chi_i^t)^2 + K_{2i}\chi_i^t(1 - \theta_i).$$

This cost function must satisfy the following conditions:

- The term $K_{2i}\chi_i^t\theta_i$ is used to rank customers based on their willingness (θ_i).
- As the willingness factor (θ_i) increases, the marginal cost decreases. The customer with the highest willingness ($\theta_i = 1$) has the lowest marginal cost and thus the highest marginal benefit. Conversely, the customer with the lowest willingness ($\theta_i = 0$) has the highest marginal cost and the lowest marginal benefit.
- The derivative of the cost function with respect to χ_i^t is given by $\frac{\partial c}{\partial \chi_i^t} = 2K_{1i}\chi_i^t + K_{2i}(1 - \theta_i)$.
- The marginal cost is non-negative.
- The cost function exhibits increasing marginal cost, indicating a convex cost structure.
- The cost of curtailing zero power should be zero ($c(\theta_i, 0) = 0$).

The customer benefit is derived from :

$$V_1(\theta_i^t, \chi_i^t, \omega_i^t) = \omega_i^t - c(\theta_i^t, \chi_i^t), \quad i = 1, \ldots, I.$$

Notice that the customer will participate if $V_1 \geq 0$.

The utility benefit is determined from:

$$u_0 = \sum_{i \in I} \lambda_i^t \chi_i^t - \omega_i^t.$$

The objective is to maximize the expected utility benefit:

$$\max_{\chi,\omega} \sum_{t \in T} \sum_{i \in I} \lambda_i^t \chi_i^t - \omega_i^t$$

subject to:

$$\sum_{t \in T} \left[\omega_{i,t} - \left(K_{1,i}\chi_{i,t}^2 + K_{2,i}\chi_{i,t} - K_{2,t}\chi_{i,t}\theta_i\right)\right] \geqslant 0, \text{ for } i = 1, \ldots, I.$$

$$\sum_{t \in T} \left[\omega_{i,t} - \left(K_{1,i}\chi_{i,t}^2 + K_{2,i}\chi_{i,t} - K_{2,t}\chi_{i,t}\theta_i\right)\right]$$

$$\geqslant \sum_{t \in T} \left[\omega_{i-1,t} - \left(K_{1,i-1}\chi_{i-1,t}^2 + K_{2,i-1}\chi_{i-1,t} - K_{2,i-1}\chi_{i-1,t}\theta_{i-1}\right)\right],$$

$$\text{for } i = 2, \ldots, I.$$

$$\sum_{t \in T} \sum_{i \in I} \omega_{i,t} \leqslant UB.$$

$$\sum_{t \in T} \chi_{i,t} \leqslant CM_i \quad \text{for } i = 1, \ldots, I.$$

where UB is the utility's total budget and CM_i is the daily limit of interruptible power for customer I.

5.2.3 Integrated DEED and Multi-Periodic Game-Theoretic Formulation

As presented previously, the mathematical formulation of the DEED problem involves two conflicting objective functions. These objectives are the fuel costs associated with the power system's generators and the costs related to emissions. Nwulu and Xia [46] combined the DEED with a game theoretic-based mathematical model into a single objective DR-DEED as follows:

$$\min \left[w_1 \left[\sum_{t \in T} \sum_{j \in J} C_j \left(q_j^t \right) \right] + w_2 \left[\sum_{t \in T} \sum_{j \in J} E_j \left(q_j^t \right) \right] - w_3 \left[\sum_{t \in T} \sum_{i \in I} \left[\lambda_i^t \chi_i^t - \omega_i^t \right] \right] \right]$$

where w_1, w_2 and w_3 are the weights satisfying $w_1 + w_2 + w_3 = 1$.

This modification incorporates a game theory-based demand response (DR) program with DEED formulation. The results indicate that this combined model can determine the optimal generator output from the available generators, while the game theory-based demand response program helps the utility identify the optimal customer load to curtail and the optimal incentive to offer customers who agree to reduce their load. The key findings from this model demonstrate the effectiveness of a combined approach in reducing total electricity demand over 24 hours. Moreover, the model shows that engaging willing customers through demand response programs provides a cost-efficient way for the power system to lower overall energy consumption.

5.3 Applying Approximation Algorithm

The first scenario is for six generator units and five customers with the same data from [46]. The enhanced approximation algorithm will simultaneously minimize the factors C,E,P, and Loss while also maximizing a new variable, the utility (U).

Direction	Vertex	C	E	U	P	$Loss$
Fixed-Direction	Random-Vertex	290566.15	25372.42	65822.59	24211.84	265.79
	Ideal-Point-Vertex	290548.68	24866.60	64800.17	24186.83	263.99
	Inner-Point-Vertex	289214.97	27429.61	64755.39	24203.27	279.14
	Cluster-Vertex	290897.98	24900.40	65831.55	24208.55	263.01
Fixed-Point Direction	Random-Vertex	290360.81	25358.82	64836.80	24195.96	265.45
	Ideal-Point-Vertex	291188.53	24151.56	64760.15	24182.37	259.59
	Inner Point	289316.45	27168.96	64672.08	24201.66	277.54
	Cluster	290779.35	24782.57	64854.41	24192.10	262.04
Ideal-Point Direction	Random-Vertex	290360.81	25358.82	64836.80	24195.96	265.45
	Ideal-Point-Vertex	291188.53	24151.56	64760.15	24182.37	259.59
	Inner-Point-Vertex	289316.45	27168.96	64672.08	24201.66	277.54
	Cluster-Vertex	291832.07	23813.05	65001.52	24190.26	255.01
Approximate-Adj Direction	Random-Vertex	292827.88	23640.38	68101.22	24220.44	251.73
	Ideal-Point-Vertex	293351.22	23070.73	67854.94	24204.86	248.27
	Inner-Point-Vertex	293351.22	23070.73	67854.94	24204.86	248.27
	Cluster-Vertex	292827.88	23640.38	68101.22	24220.44	251.73

For the 'Fixed Direction', the Ideal Point vertex demonstrates the minimal values for C (290548.68), E (24866.60), and P (24186.83), while the Cluster-Vertex exhibits the lowest $Loss$ (263.01) and the maximal U (65831.55).

Transitioning to the 'Fixed Point Direction', the Inner Point-Vertex records the lowest C(289316.45), whereas the Ideal Point-Vertex has the minimal E(24151.56), P(24182.37), and $Loss$(259.59). The Cluster-Vertex manifests the maximal U in this setting with a value of (64854.41).

Turning to the 'Ideal Point Direction', the Inner Point-Vertex has the lowest $C(289316.45)$, the Ideal Point-Vertex has the minimal $P(24182.37)$, and the Cluster-Vertex shows the lowest $E(23813.05)$ and $Loss(255.01)$, as well as the maximal $U(65001.52)$.

Finally, for the 'Approximate Adj Direction', the Random-Vertex achieves the minimal $C(292827.88)$ and maximal $U(68101.22)$, while the Ideal Point-Vertex and Inner Point-Vertex both have the lowest $E(23070.73)$, $P(24204.86)$, and $Loss(248.27)$. The Cluster-Vertex has the minimal $C(292827.88)$and maximal $U(68101.22)$.

Overall, the analysis suggests that the 'Ideal Point Direction' with the 'Cluster Point-Vertex would be the most appropriate choice to compare the results with those in [57].

For the 'Approximate Adj Direction' scenario, the best vertex selection to compare to the other paper's results would be the Inner Point-Vertex. We will also compare the result of this direction selection to the result in [57].

Model	C	E	U	P	$Loss$
Enhanced Approximation Algorithm with IP	291832.07	23813.05	65001.52	24190.26	255.01
Enhanced Approximation Algorithm with Adj	293351.22	23070.73	67854.94	24204.86	248.27
DR-DEED	291898.16	24474.04	48046.98	24266.59	265.61

Table 5.1: Comparing Results Scenario 1

Regarding the total cost C, the Enhanced Approximation Algorithm managed to lower it by 0.02% compared to the DR-DEED model. This slight yet meaningful reduction showcases the optimization capacity of our work. In the domain of emissions E, the Enhanced Approximation Algorithm outperformed the DR-DEED model by achieving a 2.69% decrease, confirming its efficiency in reducing emissions output. When examining power generation P, the Enhanced Approximation Algorithm

demonstrated a 0.31% decrease. Furthermore, the Enhanced Approximation Algorithm distinguished itself by reducing the overall power loss by 4.00% compared to the DR-DEED model. This improvement in minimizing losses underscores the algorithm's capabilities. The most remarkable enhancement, however, lies in the area of utility U, where the Enhanced Approximation Algorithm achieved an impressive 35.31% increase over the DR-DEED model.

For the second experiment with ten generator units and seven customers, we have the following results:

Direction	Vertex	C	E	U	P	$Loss$
Fixed-Direction	Random-Vertex	978705.41	233403.42	85657.13	38621.59	1164.80
	Ideal-Point-Vertex	981604.55	218575.85	93410.74	38608.02	1154.41
	Inner-Point-Vertex	976281.72	247148.19	77970.16	38631.11	1174.67
	Cluster-Vertex	980414.69	229297.78	85940.95	38647.95	1172.13
Fixed-Point Direction	Random-Vertex	982317.65	248743.51	85565.65	38675.66	1180.38
	Ideal-Point-Vertex	979028.31	226282.24	89200.32	38612.23	1158.44
	Inner Point	975296.00	266455.57	75116.46	38647.60	1188.77
	Cluster	983177.93	240840.13	88263.17	38668.02	1174.21
Ideal-Point Direction	Random-Vertex	982317.65	248743.51	85565.65	38675.66	1180.38
	Ideal-Point-Vertex	979028.31	226282.24	89200.32	38612.23	1158.44
	Inner-Point-Vertex	975296.00	266455.57	75116.46	38647.60	1188.77
	Cluster-Vertex	982317.65	248743.50	85565.65	38675.66	1180.38
Approximate-Adj Direction	Random-Vertex	982625.18	241061.13	91158.83	38626.07	1175.88
	Ideal-Point-Vertex	979277.80	225540.15	91118.08	38572.68	1157.03
	Inner-Point-Vertex	982307.85	228180.92	93024.73	38528.67	1160.38
	Cluster-Vertex	983278.67	219877.52	99053.52	38516.45	1155.31

The data presented in the provided table suggests that the 'Ideal-Point-Vertex' method for vertex selection under the 'Fixed-Direction' category offers the most favorable results. This can be substantiated by the following observations: the Ideal-Point-Vertex method exhibits the minimum values for the performance metrics Emis-

sions E (218575.85), Power Generation P (38608.02), and Power $Loss$ (1154.41) when compared to the other vertex selection methods in the Fixed-Direction category. Furthermore, this method demonstrates the highest value for the Utility U (93410.74), indicating its ability to maximize this metric while simultaneously minimizing the other performance measures. Notably, the 'Random-Vertex' method presents the minimal cost C (978705.41). However, the cost associated with the Ideal-Point-Vertex method is 0.296% higher than this minimum Cost.

Furthermore, the 'Ideal-Point-Vertex' approach for vertex selection within the' Fixed-Point' Direction category delivers the most advantageous results. The Ideal-Point-Vertex method demonstrates the lowest values for the performance indicators E (226282.24), P (38612.23), and $Loss$ (1158.44) when compared to the other vertex selection methods in this category. Additionally, this method exhibits the highest value for the Utilization (89200.32).

The 'Inner-Point' method presents the minimal C (975296.00). However, the cost associated with the Ideal-Point-Vertex method is 0.382% greater than this minimum Cost. Nonetheless, considering the superior performance of the Ideal-Point-Vertex method across the other metrics, it can be reasonably concluded that this method represents the most appropriate choice for vertex selection within the Fixed-Point Direction category.

The 'Ideal-Point' Direction category has the same scenario as the Fixed-Point Direction, and therefore the Ideal-Point-Vertex method is also the most appropriate choice for vertex selection in this category.

The analysis now turns to the 'Approximate-Adj Direction' category, where the

data reveals another intriguing set of findings. In this instance, the 'Cluster-Vertex' method emerges as the front-runner, demonstrating the most desirable performance across the key metrics. Specifically, this technique boasts the lowest values for the E (219877.52), P (38516.45), and Power $Loss$ (1155.31) indicators, while also achieving the highest Utilization rate U (99053.52). These results strongly suggest that the Cluster-Vertex method is the optimal choice for vertex selection within the Approximate-Adj Direction scenario. It is noteworthy that the Inner-Point method presents the overall minimal cost of 979277.80 across all the categories examined. However, the cost associated with the Cluster-Vertex method in the Approximate-Adj Direction is 0.405% higher, at 983278.67. Nonetheless, given the Cluster-Vertex method's superior performance on the other critical metrics, this marginal increase in Cost can be considered a worthwhile trade-off to achieve the most well-rounded and effective vertex selection solution for this particular direction.

Based on the information provided, the Cluster-Vertex method for the Approximate-Adj Direction category would be the most appropriate choice to consider as a benchmark when comparing results with those in [57], as it offers the best balance of the key performance indicators.

Model	C	E	U	P	$Loss$
Enhanced Approximation Algorithm with Adj	983278.67	219877.52	99053.52	38516.45	1155.31
DR-DEED	989439.22	192743.96	83341.97	38574.55	1137.11

Table 5.2: Comparing results Scenario 2

The Enhanced Approximation Algorithm with Approximate-Adj Direction(Adj) shows some close results to the DR-DEED model, but also some notable differences. For total cost C, the Enhanced Approximation Algorithm with Adj has a value of

$983,278.67$ compared to $989,439.22$ for the DR-DEED model, a difference of only 0.62%. This indicates the Enhanced Approximation Algorithm with Adj is able to achieve a total cost that is quite close to the DR-DEED model. However, for emissions E, Compared to the DR-DEED model, the Enhanced Approximation Algorithm with Adj has a notable rise in emissions, with a 14.07% increase. On the utility U metric, the Enhanced Approximation Algorithm with Adj achieves $99,053.52$, which is 18.76% higher than the $83,341.97$ for DR-DEED. This demonstrates a notable improvement in overall system utility. For power generation P, the Enhanced Approximation Algorithm with Adj shows a decrease in the value with a 0.15%. This is a very minor difference, indicating the power generation is quite close between the two models.

Finally, for power loss, the algorithm has a 1.59% increase. This represents a small worsening in power loss reduction compared to the DR-DEED model. In summary, the Enhanced Approximation Algorithm with Adj achieves results quite close to the DR-DEED model for total cost, power generation, and power loss, but has a notable worsening in emissions and significant improvement in overall utility.

CHAPTER 6

DYNAMIC ECONOMIC EMISSION DISPATCH IN MULTI-PERIOD SMART GRID DEMAND RESPONSE (NEW MODEL)

Renewable energy sources and smart grid technology have introduced new challenges and opportunities for optimizing power systems. Traditional approaches to economic dispatch and emission reduction have primarily focused on the generation side of the equation. However, demand-side flexibility enabled by smart grid demand response programs is critical to achieving sustainable and cost-effective power systems. In this work, we present a model that integrates game-theoretic and multi-objective

models for dynamic economic emission dispatch in multi-period smart grid demand response. By actively involving customers as strategic participants in the optimization process, our model utilizes the power of the smart grid to better balance economic, environmental, and customer satisfaction objectives. The integration of game theory-based demand response into the DR-DEED framework opens up new avenues for research and practical implementation, driving the transition toward a more flexible, efficient, and consumer-oriented power system can be achieved.

Economic dispatch and demand-side management are considered to be among the most critical tools for ensuring efficient energy management in the grid. Economic dispatch aims to optimize the allocation of power generation resources to minimize fuel and emissions costs while meeting electricity demand. On the other hand, demand-side management focuses on influencing consumer behavior and load patterns to achieve a more balanced and sustainable energy consumption. In this chapter, we introduce a new method to solve the dynamic economic emission problem (DR-DEED) using game theoretic models integrated with smart grids. This integration empowers customers to actively shift their electricity usage, generate renewable energy, store excess energy, and even sell it back to utility companies. In the following section, we will define our modifications that represent demand response using the context of the smart grid.

6.1 Game Theoretic Models For Smart Grids With Shifted Demand

The capability of DR within SG is centered around enabling effective interaction between the supply and demand sides. This interaction involves the exchange of price and demand information, allowing participants to make informed decisions for efficient energy management [58]. The integration of energy storage plays a critical role in enhancing the effectiveness of demand response within the smart grid. Additionally, the successful performance of demand response is closely tied to the effective utilization of Renewable Energy Resources (RERs)[59]. By enabling real-time adjustments to electricity demand, demand response enhances the flexibility of the smart grid, ensuring grid stability, reliability, and efficiency by dynamically modifying energy consumption patterns.

The primary mathematical concept shared by SG and DEED, in conjunction with MOP, is the utilization of a game theory-based demand response model with multiple customers across various time periods. The concept of the game theory applies whenever multiple agents, termed players, are involved in a relationship of interdependence. Game theory can be classified into two main branches: cooperative games and non-cooperative games. Cooperative games involve analyzing scenarios where players have the ability to collaborate, form coalitions, and establish binding agreements. Non-cooperation games, on the other hand, explore situations where players make decisions independently, without forming explicit agreements or engaging in cooperative behavior [60].

Several authors have proposed game theoretical models to address various challenges in the field of smart grids and their advancement. For instance, Cui *et al.* in [39] presented two models for determining prices in utility companies, considering different assumptions. They employed a Nash equilibrium solution from the n-person game model [45] in their initial model of energy price competition, which was demonstrated to be unique under certain assumptions. Additionally, Atzeni *et al.* [41], [40] examined smart grids that incorporate a combination of traditional users along with individuals who possess distributed energy sources or energy storage devices. Atzeni *et al.* formulated the problem of demand-side management in smart grids as a non-cooperative game.

Instead of shifting energy consumption, the Demand Response Management (DRM) method has been proposed, which involves a day-ahead optimization process for scheduling energy production and storage. Belhaiza *et al.* [42] introduced a new theoretical model based on a non-cooperative multi-agent game, where users and providers interact to regulate the demand in a smart grid. Their objective was to motivate consumers to modify their electricity usage patterns by adjusting their consumption during peak load periods. As a result, DRM can reduce consumers' energy bills and support the energy system by shifting low-priority demands from peak to off-peak hours. The proposed model in this context suggests a Nash equilibrium for the smart grid game. Recently, Ullah *et al.* [43] presented an optimal tri-objective problem for the implementation of demand side strategies in smart grids with high wind energy penetration in the day-ahead scheduling problem.

Additionally, Belhaiza *et al.* in [44] proposed a game-theoretic model for the

multi-periodic smart grid demand response management problem with shifted demand, considering it as a MOP problem. Their work focused on developing a model for electricity consumption and storage in residential and industrial demand response, where electricity storage is distributed as needed to users. They also considered the comparative advantage of delaying consumption to later periods. The primary objective of the proposed game-theoretical model is to optimize the utility functions of each agent involved while ensuring that the interactions between users and providers meet the required conditions.

In this work, our objective is to utilize the approach of Belhaiza *et al.* [44] and apply its methodologies to a different domain. While we do not apply the model in its entirety, we will capitalize on its utilization by incorporating certain variables, parameters, and the constraints of the existing optimization model. With these specific elements, we can enhance the effectiveness of our analysis and derive significant benefits. In table 6.1, we explicitly define the variables and parameters that we will use from the optimization model.

Symbol	Type	Description
x_i^t	Real	The power sent to the grid from customer i at time t.
f_i^t	Real	Demand of user i satisfied at period t.
h_i^{t+1}	Real	Demand of user i during period t to be shifted to period $t+1$.
p_i^t	Real	Quantity of energy user i produces at period t .
s_i^t	Real	Quantity of energy user i stores during period t.
$a_{i,t}^{\cdot}, b_{i,t}^{\cdot}, c_{i,t}^{\cdot}, v_{i,t}^{\cdot}$ $d_{i,t}^{\cdot}, e_{i,t}^{\cdot}, (ah)_{i,t}^{\cdot}, (af)_{i,t}^{\cdot}$	coefficients	The cost coefficients of user i respectively.
Dm_i^t	Parameter	Demand of customer i at time t.
L_i^t	Parameter	maximum capacity of power sent to the grid from customer i at time t.

Table 6.1: Variable and Parameter of SG

6.2 The Proposed Model

In this work, we introduce a comprehensive model that incorporates the concepts of DEED, SG with shifted demand, and DR. Thus, it is a DR-DEED for smart grids with shifted demand and hence we shall label it form here and after as a model for **Smart-Grid Shifted Demand Deed Model (SGSD-DEED)**. By integrating these key concepts, our model provides a holistic approach to address challenges and optimize energy management in modern power systems. Our model takes into account the dynamic nature of the smart grid and the capacity to shift demand across various periods. This dynamic characteristic permits the optimization of energy consumption patterns and load balancing through strategic adjustments of demand response actions. The incorporation of DR enables the adaptation of electricity consumption patterns in response to price signals or grid conditions. Moreover, this integration facilitates the achievement of load flexibility and the effective management of peak demand periods, ultimately contributing to the stability of the grid and the optimization of costs. This approach includes the utilization of cutting-edge technologies the encouragement of active consumer involvement in load management and the reduction of peak demand. Additionally, our model actively contributes to reducing the environmental impact of electricity generation, promoting sustainability, and ensuring compliance with regulatory standards by ensuring that it minimizes the overall emissions while still meeting the demand requirements.

To optimize customer satisfaction, we establish utility functions for each individual. These functions serve as mathematical representations of their preferences in game

theory, allowing us to identify the choices or outcomes that yield the highest level of satisfaction or value when maximized.

The utility function for each customer $i \in I$ is formulated as :

$$V_i(p_i^t, x_i^t, s_i^t, h_i^t, f_i^t) = a_{i,t}^{\cdot} + b_{i,t}^{\cdot} p_i^t + c_{i,t}^{\cdot} (p_i^t)^2 + \nu_{i,t}^{\cdot} x_i^t + d_{i,t}^{\cdot} s_i^t + e_{i,t}^{\cdot} (s_i^t)^2 + (ah)_{i,t}^{\cdot} h_i^t + (af)_{i,t}^{\cdot} f_i^t.$$

We will now introduce a new utility function. This cooperative utility function represents the combined benefit or utility obtained from the integration of SG and DR-DEED strategies. The new utility function will serve as a comprehensive measure of the overall system performance.

$$U_i(\chi_i^t, p_i^t, x_i^t, s_i^t, h_i^t, f_i^t, \omega_i^t) = \lambda_i^t \chi_i^t + a_{i,t}^{\cdot} + b_{i,t}^{\cdot} p_i^t + c_{i,t}^{\cdot} (p_i^t)^2 + \nu_{i,t}^{\cdot} x_i^t + d_{i,t}^{\cdot} s_i^t + e_{i,t}^{\cdot} (s_i^t)^2 + (ah)_{i,t}^{\cdot} h_i^t + (af)_{i,t}^{\cdot} f_i^t - \omega_i^t.$$

The objective is to maximize the utility function, aiming to achieve the highest benefit :

$$\max_{\chi,p,x,s,h,f,\omega} U_i(\chi_i^t, p_i^t, x_i^t, s_i^t, h_i^t, f_i^t, \omega_i^t)$$

This mathematical representation assigns a numerical value to measure the overall benefit or advantage that an individual customer obtains from a specific action or decision. We construct the customers' benefit function from their ability to store,

along with their ability to shift demand to the next hour. Moreover, through the integration of distributed energy resources, customers can generate power using smart grid technologies that provide the necessary infrastructure and communication capabilities. With smart grid technology, electric grid customers can reduce their reliance on traditional grid electricity, lower their energy costs, contribute to the sustainability of the environment, and enhance their energy independence.

To outline the parameters and limitations for achieving a viable and optimal solution, we will now define the constraints of the model. By explicitly defining these constraints, we establish the necessary framework and conditions that must be satisfied to ensure the feasibility of the solution.

6.2.1 Energy Flow Conservation Constraints

In order to ensure a good balance between energy flows, we formulate the subsequent flow conservation constraints.

$$f_i^t + s_i^t - s_i^{t-1} - p_i^t + x_i^t = \sum_{j \in J} y_{ij}^t, \qquad \forall t \in T, \tag{6.1}$$

$$DM_i^t + h_i^t = h_i^{t+1} + f_i^t, \qquad \forall t \in T, \tag{6.2}$$

$$\sum_{i \in I} y_{ij}^t = q_j^t, \qquad \forall t \in T, \tag{6.3}$$

$$\sum_{i \in I} x_i^t + \sum_{j \in J} q_j^t = \sum_{i \in I} DM_i^t + losst_t - \sum_{i \in I} \chi_i^t, \quad \forall t \in T. \tag{6.4}$$

In the first balance constraint (6.1), there is a requirement for flow conservation such that the sum of the quantities received, stored before, or produced by user i at the beginning or during period t equals the sum of the quantities that user i sends to the grid, consumes or stores during or at the end of the given period.

The second balance constraint (6.2) states that the sum of the consumption of user i during period t and the demand shifted to the next period $t+1$ equals the demand for the period t and the demand of user i for period t.

The third balance constraint (6.3) ensures that the sum of all the quantities sent by provider j during period t is equal to the power generated from the generator j at time t.

The fourth balance constraint (6.4), is about electricity conservation which states that the total electricity received from the power plants should equal the sum of the power sent to the grid from customer i with the total electricity demand of users and the total system losses at time t subtracted by the amount of power reduced by a participating customer i at time t.

Individual Rationality Constraints

The following constraints are about the individual rationality constraint which is a fundamental principle in economic and game theory. This principle ensures that participants derive benefit from their involvement in the system. Additionally, we incorporate the incentive compatibility condition, which ensures that customers are appropriately compensated for the level of load curtailment they experience. This condition guarantees that customers receive fair and suitable compensation based on the extent to which their loads are reduced. These constraints will be considered for

every customer and every period as follows:

$$\omega_i^t - \chi_i^t \left(K_{1,i}(\chi_i^t) + K_{2,i}(1-\theta_i\right) \geqslant 0, \quad \forall i \in I \ , \forall t \in T, \tag{6.5}$$

$$\begin{aligned} \omega_i^t - \chi_i^t \left(K_{1,i}(\chi_i^t) + K_{2,i}(1-\theta_i)\right) \geqslant \omega_{i-1}^t - \chi_{i-1}^t \big(K_{1,i-1}(\chi_i^t) \\ + K_{2,i-1}(1-\theta_{i-1})\big), \\ \forall i \in I, i > 1, \forall t \in T. \end{aligned} \tag{6.6}$$

Limits Constraints

The upcoming constraints are about the limit constraint, which plays a crucial role in defining the restrictions or bounds on the variables or parameters within the model. By incorporating the limit constraint, we establish a framework that prevents values from exceeding certain limits or falling below defined minimums. This constraint serves to maintain the integrity and feasibility of the model, as it reflects real-world limitations or practical considerations.

$$q_j^{\min} \leq q_j^t \leq q_j^{\max}, \; j \in J, \quad t \in T, \tag{6.7}$$

$$p_i^{\min} \leq p_i^t \leq p_i^{\max}, \; i \in I, \quad t \in T, \tag{6.8}$$

$$p_i^{\min} \leq s_i^t \leq p_i^{\max}, \; i \in I, \quad t \in T, \tag{6.9}$$

$$-DR_j \leq q_j^{t+1} - q_j^t \leq UR_j, \; j \in J, \quad t \in T, \tag{6.10}$$

$$\sum_{i \in I} \omega_i^t \leq UB, \qquad \forall t \in T, \tag{6.11}$$

$$\sum_{t \in T} x_i^t \leq L_i^t, \qquad \forall i \in I, \tag{6.12}$$

$$\sum_{t \in T} \chi_i^t \leq CM_i, \qquad \forall i \in I. \tag{6.13}$$

Constraints (6.7),(6.8), and (6.9) are about the power output constraints where the power output from each power plant q_j^t should be within the specified minimum and maximum limits. Moreover, the amount of energy p_i^t, and the storage s_i^t produced by energy user i during period t need to fall within predetermined minimums and maximums. Constraint (6.10) is about the power ramp rate constraints where the change in power output from one time period to the next for each power plant should be within the specified up and down ramp rates. Constraint (6.11) refers to the transmission budget limits for power plants, which aim to ensure a balance between the financial incentives provided by the utility and the budgetary constraints of the utility. Constraint (6.12) states the sum of all the quantities sent by users i during cannot exceed power generated maximum capacity. Constraint (6.13) is about the electricity market participation which limits the total amount of electricity that a

household can sell in the market over all time periods.

Initial and Final State Constraints

For initial and final conditions, we will present the following constraints:

$$s_i^0 = s_i^T = h_i^1 = 0, \quad i \in I. \tag{6.14}$$

The statement indicates that the quantities stored by users at the beginning and the end of the period as well as the initially shifted demand to period 1 are all equal to zero.

Our model aims to find a compromise between the following three functions:

$$\begin{aligned}
\mathcal{C}(\mathbf{q}) &= \sum_{t\in T}\sum_{j\in J} C_j\left(q_j^t\right) \\
\mathcal{E}(\mathbf{q}) &= \sum_{t\in T}\sum_{j\in J} E_j\left(q_j^t\right) \\
\mathcal{U}(\chi,\mathbf{p},\mathbf{x},\mathbf{s},\mathbf{h},\mathbf{f},\omega) &= \sum_{t\in T}\sum_{i\in I} U_i(\chi_i^t, p_i^t, x_i^t, s_i^t, h_i^t, f_i^t, \omega_i^t)
\end{aligned}$$

Hence the mathematical formulation of our model SGSD-DEED is :

$$\left.\begin{array}{l}
\min \quad w_1\mathcal{C}(\mathbf{q}) + w_2\mathcal{E}(\mathbf{q}) - w_3\mathcal{U}(\chi,\mathbf{p},\mathbf{x},\mathbf{s},\mathbf{h},\mathbf{f},\omega) \\
\text{subject to} \\
\bullet \text{ The energy flow conservation conditions: Equations (6.1) to (6.4).} \\
\bullet \text{ The individual rationality conditions: Equations (6.5) and (6.6).} \\
\bullet \text{ The boundary conditions: Equations (6.7) to (6.13).} \\
\bullet \text{ The initial and final state conditions: Equation (6.14).}
\end{array}\right\}_{SGSD-DEED}$$

The choice of this model is due to the fact that it represents a significant advancement in addressing the DEED problem through the utilization of smart grid technology. By providing customers with the ability to shift, store, send power to the grid, and generate power, a new level of engagement is introduced to the energy sector. The ability to shift power consumption helps to balance demand and reduce pressure on the grid during peak periods. Furthermore, customers have the ability to store excess energy for future usage, enhancing grid resilience and minimizing losses. By sending power back to the grid, customers become valuable contributors to the overall energy supply, promoting a more distributed and sustainable energy ecosystem. Moreover, customers' ability to generate power reduces dependence on traditional energy sources and facilitates the integration of renewable energy. Overall, providing customers with these capabilities not only benefits individual consumers but also strengthens the entire energy system.

Through the appropriate selection of the w_i values, where $i = 1, 2, 3$, and the incorporation of specific constraints derived from our model, the DEED and DR-DEED models can be established. When $w_3 = 0$, the DEED objective function can be formulated, subject to the constraints outlined in constraints (6.4), (6.7), and (6.10). On the other hand, for the DR-DEED model, if we set the utility function as $\lambda_i^t \chi_i^t - \omega_i^t$ and consider the constraints (6.5), (6.6), (6.11), and (6.12), in addition to the constraints from the DEED problem, we can establish the model presented in [46].

6.3 Numerical Simulations and Results

In this section, we shall conduct two main experiments. The first one aims to compare the proposed model section 6.2.1 with the DR-DEED model proposed in [46]. The second experiment is to study the correlations (if any) between the different parameters of the section 6.2.1 and the number of customers and generators. A Julia package, DRDeed.jl [61], was developed to implement the model. Details are provided in (A.2).

6.3.1 Experiment 1

In the first experiment, we use the two scenarios discussed in [46] as a basis of comparison between the new model and DR-DEED in terms of effectiveness in reducing the cost and emission of the grid and maximizing the customers' utilities.

Customers Data

We discuss several scenarios by varying the number of customers and generators. The underlying assumption is that the customer interruptible power limits, CM_j, are provided to the utility by the customer. This daily limit of interruptible power is then used to determine the customer's willingness θ_i. The customer's willingness can be understood as their willingness or readiness to have their power supply interrupted within the specified limit. Moreover, the utility knows the different coefficients of the outage cost function for the customer $K_{1,i}$ and $K_{2,i}$.

The objective is to achieve the optimal amount of customer power to be curtailed χ_i^t, the optimal incentive to be paid to customers ω_i^t, the power generated by all generators

q_j^t, and the energy produced by customers p_j^t. Several scenarios will be applied to the model to evaluate the effectiveness of the proposed mathematical formulations. In the first scenario, we will consider six generator units and five customers. In the second scenario, we will consider ten generators units and seven customers. Then, we will generalize the work for more than 100 customers with several generators. Various weight cases are considered from 6.2 for all scenarios. The data are obtained from [46]. These data include the cost and emission coefficients of the generators, the hourly values of power interruptibility λ_i^t, the transmission loss formula coefficients B, the customer type θ_i, and the daily customer power limit CM_j. The hourly values of power interruptibility λ_i^t obtained from the PJM Market on the 1st of May 2014 where PJM denotes the Pennsylvania–New Jersey–Maryland Market.

	w_1	w_2	w_3
Base Case (BC)	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
Case 2 (C2)	1	0	0
Case 3 (C3)	0	1	0
Case 4 (C4)	0	0	1

Table 6.2: Various weighting factor values

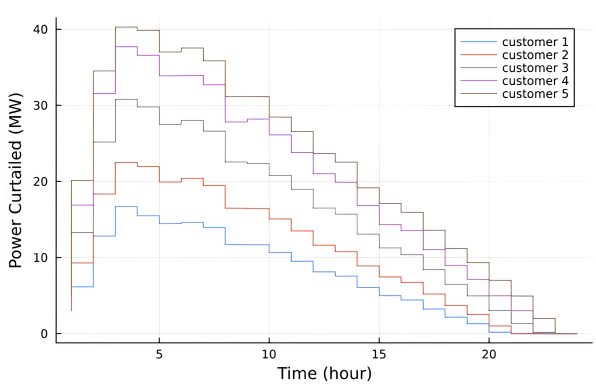


Figure 6.1: Optimal power curtailed for all customers (Scenario 1)

.

Scenario 1 (generators=6, customers=5)

For the generators, the fuel cost coefficients and the emission cost coefficients are obtained from Table A.1 in [46]. For the customers, the cost coefficients, the power boundaries, and the demand of the customers are shown in table B.3.

Scenario 2 (generators=10, customers=7)

For the generators, the fuel cost coefficients and the emission cost coefficients are obtained from Table A.2 in [46]. For the customers, the cost coefficients, the power boundaries, and the demand of the customers are shown in table B.4.

Discussion of the Results

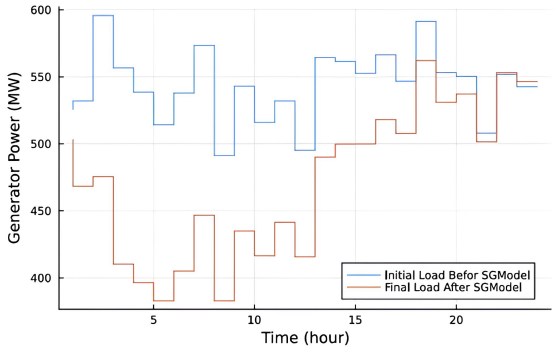


Figure 6.2: Total load profile before and after SGSD-DEED model (Scenario 1)

.

Considering Scenario 1, fig. 6.2 shows the total load demand profile before and after applying SGSD-DEED model model. The figure clearly illustrates the positive impact of the SGSD-DEED model model. The load demand decreases significantly

and this reduction indicates improved efficiency and effective energy consumption. Moreover, fig. 6.1 shows the optimal power curtailed. A major impact of our model is the reduction of total system losses for the power system. Comparing the results of our model with the DR-DEED model from [46] and table B.5, we noticed that: the losses reduce in the BC case from 265.61 MW to 54.87 MW, with a percentage reduction of 79.34%. Furthermore, in other cases of scenario 1, our model demonstrates significant decreases in losses. The losses reduce with a percentage reduction of 79.71% in the C2 case. In the case of C3, the losses decrease by 78.03%. Finally, in C4, there is a percentage reduction of approximately 78.33%. A comprehensive overview of the loss reductions and percentage reductions for each case is in table 6.3.

	Loss			
	BC	C2	C3	C4
DR-DEED	265.61	308.20	231.01	234.85
SGSD-DEED model	54.87	62.41	50.85	50.77
Percentage (%)	79.34	79.71	78.03	78.33

Table 6.3: Comparison of Loss Improvement between DR-DEED and SGSD-DEED Models

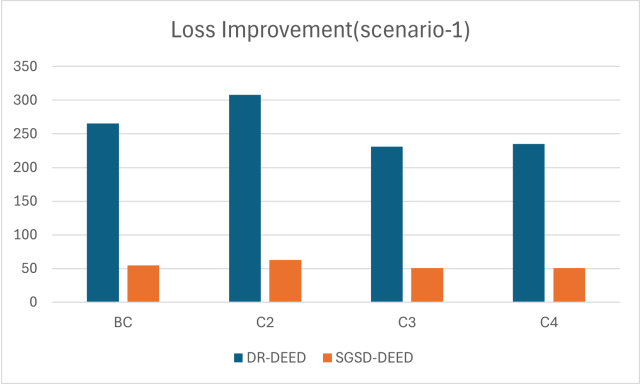


Figure 6.3: Comparison of Loss Improvement between DR-DEED and SGSD-DEED Models

These significant reductions in losses highlight the superior performance of our model in optimizing the power system and minimizing inefficiencies. By achieving such substantial reductions, our model leads to increased efficiency, cost savings, and overall benefits for the power system compared to the DR-DEED work.

In terms of efficiency, our model focuses on minimizing energy losses that occur during power generation, transmission, and distribution processes. Consequently, the power system operates more efficiently, reducing fuel consumption and the need for additional generation capacity.

For the cost-saving aspect, lower system losses translate to reduced energy wastage, resulting in decreased fuel consumption and operational costs for utilities. These savings can potentially be passed on to end consumers, leading to lower electricity bills. A comparison of fuel cost reductions between our model and the DR-DEED model reveals significant advantages in favor of our approach. In case BC, our model achieves a cost reduction of 50.91% compared to the DR-DEED model. Similarly, in case C2, our model outperforms DR-DEED with a 52.37% reduction. In case C3, our model achieves a 51.07% reduction in costs compared to DR-DEED. Finally, in case C4, our model demonstrates a 52.77% reduction. These comparisons highlight the superior cost-saving capabilities of our model. Further details on costs and percentage reductions are in table 6.4.

	Cost			
	BC	C2	C3	C4
DR-DEED	291898.16	288430.67	299397.36	300815.08
SGSD-DEED	142959.81	137332.87	144356.03	142087.62
Percentage (%)	50.91	52.37	51.07	52.77

Table 6.4: Comparison of Cost Improvement between DR-DEED Model and SGSD-DEED Model

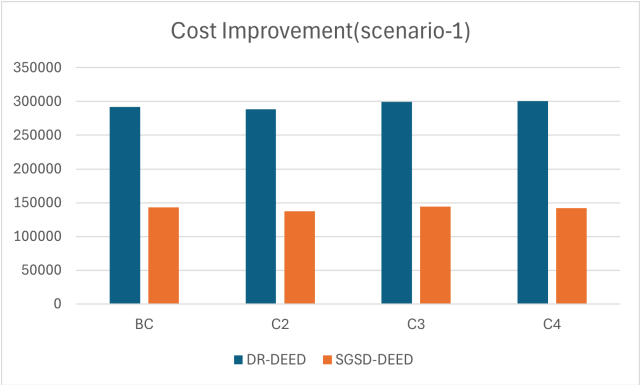


Figure 6.4: Comparison of Cost Improvement between DR-DEED Model and SGSD-DEED Model

The model contributes to minimizing power system emissions, demonstrating its effectiveness in reducing environmental impact and promoting sustainability. In the BC case, our model achieves a significant reduction in emissions of 66.90%. In the C2 case, emissions are reduced to 71.71% compared to the RD-DEED model. Similarly, in the C3 case, our model achieves a reduction in emissions of 69.57%. Finally, in the C4 case, our model results in a reduction in emissions of 62.68%. These substantial reductions underscore the model's capability to minimize environmental impact and contribute to a cleaner and more sustainable energy generation process compared to the DR-DEED approach.

	Emissions			
	BC	C2	C3	C4
DR-DEED	24474.04	31426.55	21106.33	21473.48
SGSD-DEED	8093.74	8901.93	6425.69	8000.68
Percentage (%)	66.90	71.71	69.57	62.68

Table 6.5: Comparison of Emissions Improvement between DR-DEED and SGSD-DEED Models

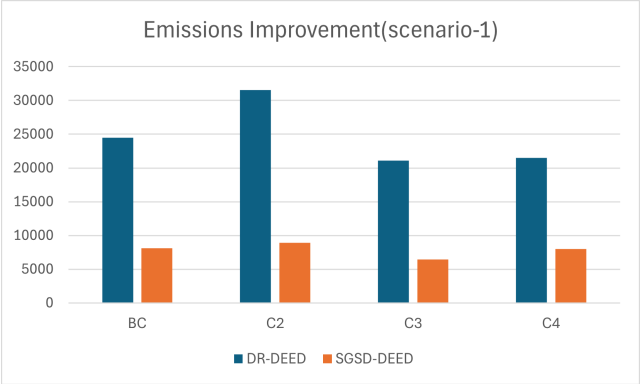


Figure 6.5: Comparison of Emissions Improvement between DR-DEED and SGSD-DEED Model

Furthermore, our model exhibits an additional notable impact in terms of minimizing power generation by the generator, which is achieved by minimizing losses. When compared to the DR-DEED approach, our model achieves substantial decreases in power generation across various scenarios. Specifically, in the BC case, our model reduces the power generated, resulting in a significant percentage reduction of 53.63%. Similarly, in the C2 case, the power generated experiences a decrease, leading to a percentage reduction of 54.92%. Likewise, in the C3 case, our model effectively reduces the power generated, resulting in a percentage reduction of 54.33%. Moreover,

in the C4 case, our model achieves a reduction in power generation, resulting in a considerable percentage reduction of 54.59%. These reductions in power generation exemplify the effectiveness of our model in optimizing the power system and minimizing inefficiencies. In table 6.6, you can find detailed information about these reductions.

	Power Generated			
	BC	C2	C3	C4
DR-DEED	24266.59	24266.59	24176.39	24291.10
SGSD-DEED	11259.04	10940.40	11031.50	11022.17
Percentage (%)	53.63	54.92	54.33	54.59

Table 6.6: Comparison of Power Generated Improvement between DR-DEED Model and SGSD-DEED Model

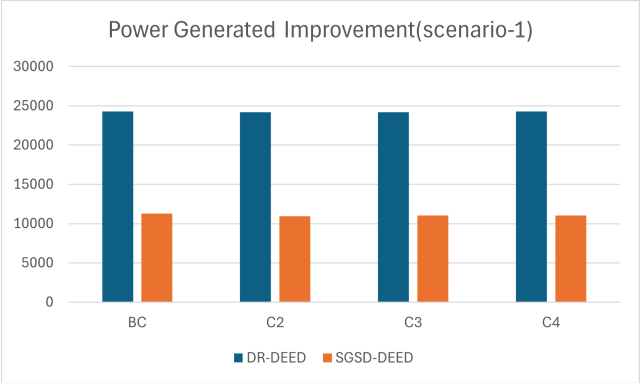


Figure 6.6: Comparison of Power Generated Improvement between DR-DEED Model and SGSD-DEED Model

In scenario 2, Our model successfully achieved significant reductions in losses, resulting in increased efficiency, cost savings, and overall benefits for the power system similar to the previous scenario. In table B.7, a comprehensive summary of the results obtained for scenario 2. These findings demonstrate that the advantages achieved through our model's implementation are not limited to a specific scenario but can be replicated and applied to different contexts, such as scenario 2. The ability

of our model to consistently deliver these favorable results across multiple scenarios underscores its effectiveness and potential for widespread adoption in power system optimization efforts.

6.3.2 Experiment 2

The second experiment aims to investigate the correlations between various parameters of the SGSD-DEED model and the number of customers and generators. This study is essential to understand how scaling the number of customers and generators affects the performance and outcomes of the SGSD-DEED model. Such insights can inform the optimization of the model's application in diverse real-world scenarios with different grid sizes.

To study these correlations, we conducted simulations with varying numbers of customers and generators. Specifically, the methodology followed these steps:

- Multiple scenarios were created with different combinations of customer and generator counts. The number of customers was varied among [50, 100, 150, 200, 250, 300, 350, 400], and the number of generators was varied among [4, 8, 12, 16, 20].

- Since the data is synthesized randomly, for each scenario SGSD-DEED model was solved three times and the average was reported. See Table table B.2 in the Appendix for details.

- All models were solved with the BC case.

Experiment Data

The data required to run these simulations was obtained by controlled randomization based on the data gathered in Scenario 1 and Scenario 2 in [46]. Each original data set (e.g., fuel cost, emission cost, utility benefits) was used to fit a normal distribution, and new data points were selected from this distribution. This approach ensured that the simulated data retained realistic characteristics and variability observed in actual grid operations. Additionally, the data needed for the utility function were synthesized based on the insights discussed in [44]. This ensured that the utility function parameters accurately reflected realistic customer behaviors and responses in a demand response context.

Parameters Analyzed

The parameters analyzed in this experiment include the overall fuel cost associated with power generation, the total emissions cost incurred from power generation, the benefits derived from integrating demand response, including customer satisfaction and compensation, the energy lost in the transmission and distribution process, and the reduction in total demand profile which is the difference between the total power generated by the system and the initial demand.

Discussion of The Results

Figure fig. 6.7 includes five distinct data series, each representing a different number of generators: 4, 8, 12, 16, and 20. For each number of generators, the cost generally decreases as the number of customers increases, then stabilizes or shows minor

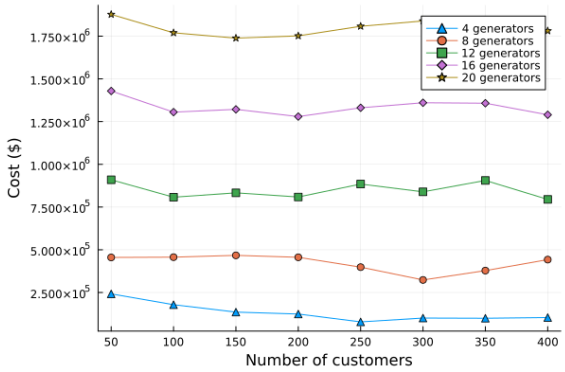


Figure 6.7: Cost, $\mathcal{C}$, for different generators

fluctuations. This trend indicates that increasing the number of customers generally leads to more efficient demand response and better load distribution, resulting in reduced costs. The cost increases with the number of generators for the same number of customers. This is expected as more generators usually mean higher operational and maintenance costs, even though they might provide more flexibility and reliability. This figure indicates that systems with fewer generators (4 or 8) benefit more significantly from increased customer numbers, showcasing higher scalability and efficiency in managing costs through demand response. Systems with more generators (16 or 20) have higher fixed costs, and the relative benefit from demand response diminishes as the operational costs dominate. For cost efficiency, systems with fewer generators appear optimal, especially when customer numbers are increased. There is a balance to be struck between having enough generators to ensure reliability and keeping the number low enough to manage costs effectively. Furthermore, we note that increasing the number of customers generally helps in reducing the overall cost due to better load balancing and more effective demand response strategies. The marginal benefit of adding more customers reduces as the system size (number of generators) increases,

indicating diminishing returns in large-scale systems.

Figure fig. 6.8 illustrates the relationship between the number of customers and emissions (measured in pounds) for different numbers of generators. The graph includes five distinct data series, each representing a different number of generators: 4, 8, 12, 16, and 20. The general trend of the graph shows that for each number

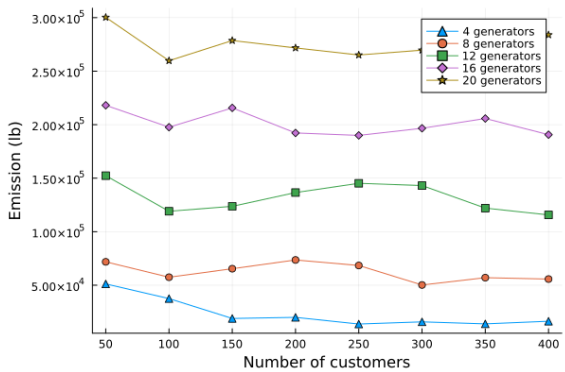


Figure 6.8: Emission, $\mathcal{E}$, for different generators

of generators, the emissions generally decrease as the number of customers increases, then stabilize or show minor fluctuations. This trend indicates that increasing the number of customers generally leads to more efficient energy usage and better load distribution, resulting in reduced emissions. The figure also demonstrates that systems with fewer generators (4 or 8) benefit more significantly from increased customer numbers, showcasing higher efficiency and lower environmental impact in managing emissions through demand response. Systems with more generators (16 or 20) have higher baseline emissions, and the relative benefit from demand response diminishes as the operational emissions dominate.

Figure fig. 6.9 provides the relationship between the number of customers and the utility for different numbers of generators. The utility metric is an aggregate mea-

sure of customer satisfaction, cost savings, and compensation. The graph includes five distinct data series, each representing a different number of generators: 4, 8, 12, 16, and 20. Utility generally decreases as the number of customers increases across

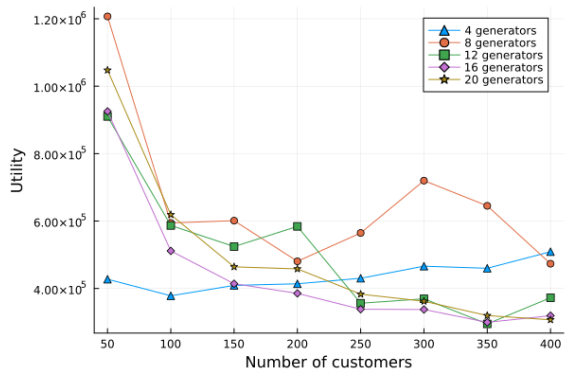


Figure 6.9: Utility, $\mathcal{U}$, for different generators

all generator configurations, then stabilizes or fluctuates. This trend indicates that while the initial utility is high with fewer customers, it decreases as the customer base grows, likely due to diminishing marginal benefits from demand response programs and increased competition for limited incentives. It is also noted that utility is highest with 8 generators at lower customer numbers but shows variability and eventual stabilization across different customer numbers. More generators generally lead to more distributed utility benefits due to increased grid stability and flexibility but also bring higher costs which can impact the net utility. Higher initial utility with fewer customers is observed across all configurations, with 8 generators showing the highest initial utility. As customer numbers increase, utility generally decreases and stabilizes due to the distribution of benefits over a larger base and the potential saturation of demand response incentives.

Figure fig. 6.10 illustrates the relationship between the number of customers and

the losses (measured in MW) for different numbers of generators. The graph includes five distinct data series, each representing a different number of generators: 4, 8, 12, 16, and 20. Losses generally decrease as the number of customers increases across

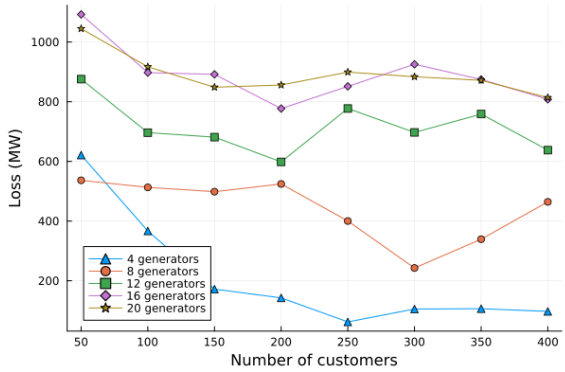


Figure 6.10: Power Loss

all generator configurations, then stabilize or show minor fluctuations. This trend indicates that increasing the number of customers generally leads to more efficient energy usage and better load distribution, resulting in reduced losses. Losses increase with the number of generators for the same number of customers. More generators usually mean more complex grid dynamics and potential inefficiencies, leading to higher losses.

Figure fig. 6.11 shows the relationship between the number of customers and the load reduction (measured in MW) for different numbers of generators. The graph includes five distinct data series, each representing a different number of generators: 4, 8, 12, 16, and 20. Load reduction is defined as the difference between the power generated by the grid (including generators and customer-generated power) and the initial network demand. We note that the load reduction generally increases as the number of customers increases across all generator configurations. This trend indicates

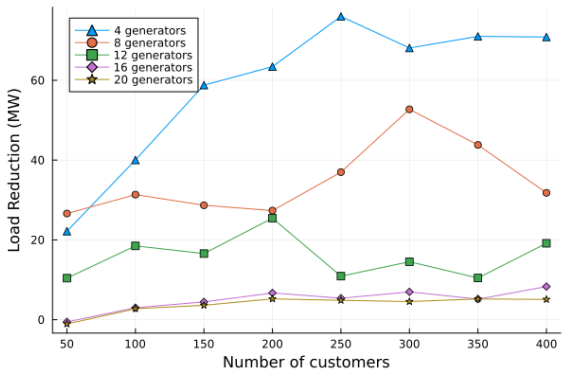


Figure 6.11: Power Load Reduction for different generators

that increasing the number of customers leads to more effective load management and higher reductions in the load due to better participation in demand response programs. Furthermore, load reduction varies significantly with the number of generators, with fewer generators generally showing higher load reductions.

The results of SGSD-DEED in Experiment 2 can be summarized as follows:

1. Cost Analysis

- **4 Generators:** Lowest cost, significant cost reduction as the number of customers increases.
- **8 Generators:** Moderate cost, cost benefits from increased customer numbers.
- **12, 16, 20 Generators:** Higher costs with diminishing cost benefits.

2. Emissions Analysis

- **4 Generators:** Lowest emissions, significant emission reduction as the number of customers increases.

- **8 Generators:** Moderate emissions, emission benefits from increased customer numbers.
- **12, 16, 20 Generators:** Higher emissions, diminishing emission benefits.

3. **Utility Analysis**
   - **8 Generators:** Highest initial utility for small customer bases, decreases with more customers.
   - **4 Generators:** Stable utility, lower but consistent performance.
   - **12, 16, 20 Generators:** Moderate to low initial utility, stable over larger customer bases.
4. **Losses Analysis**
   - **4 Generators:** Lowest losses, significant reduction with more customers.
   - **8 Generators:** Moderate losses, significant reduction with more customers.
   - **12, 16, 20 Generators:** Higher losses, modest reduction benefits.
5. **Load Reduction Analysis**
   - **4 Generators:** Highest load reduction, significant increase with more customers.
   - **8 Generators:** Significant load reduction, benefits from increased customer numbers.

- **12, 16, 20 Generators:** Moderate to low load reduction.

As a recommendation optimal configuration, the results suggest the following strategic recommendations.

For Small to Medium-sized Grids

- **Configuration:** 4 Generators with a high number of customers (300-400).
- **Reasoning:** This setup provides the lowest cost, emissions, and losses, with the highest load reduction and reasonable utility. It is highly efficient and effective for small to medium-sized grids.

For Medium to Large-sized Grids

- **Configuration:** 8 Generators with a high number of customers (300-400).
- **Reasoning:** This configuration balances cost, emissions, utility, and losses effectively, providing significant load reduction benefits. It is suitable for larger grids that require a balance between operational efficiency and flexibility.

For Large-sized Grids

- **Configuration:** 12 Generators with a high number of customers (300-400).
- **Reasoning:** While having higher costs and emissions, this setup offers stable utility and moderate load reduction, suitable for large grids that need higher reliability and redundancy.

6.4 Applying the Approximation Algorithm to SGSD-DEED

In this chapter, we will apply the enhanced approximation algorithm to the SGSD-DEED model. The results of this application will highlight the practical benefits and implications of integrating the improved approximation algorithm within the SGSD-DEED model framework. By applying the approximation algorithm with different approaches for selecting vertices and directions, we obtained the following results: For the first scenario, we have six generator units, five customers, 24 periods of time, $\epsilon = 0.05$.

direction type	Vertex selection type	C	E	U	P	Loss
Fixed-Direction	Random-Vertex	137424.27	7499.14	1472299.95	10762.04	52.36
	Ideal-Point-Vertex	136597.25	7876.68	1510324.25	10762.21	53.94
	Inner-Point-Vertex	136150.43	8089.59	1412782.56	10745.42	55.61
	Cluster-Vertex	137424.27	7499.14	1472299.95	10762.04	52.36
Fixed-Point Direction	Random-Vertex	137452.43	7761.79	1490762.07	10808.81	53.78
	Ideal-Point-Vertex	137344.86	7941.91	1540260.96	10829.23	54.54
	Inner-Point-Vertex	136366.18	8076.36	1416993.84	10764.40	55.49
	Cluster-Vertex	137765.13	7553.76	1464997.54	10797.79	52.82
Ideal-Point Direction	Random-Vertex	137452.43	7761.79	1490762.07	10808.81	53.78
	Ideal-Point-Vertex	137344.86	7941.91	1540260.96	10829.23	54.54
	Inner-Point-Vertex	136366.18	8076.36	1416993.84	10764.40	55.49
	Cluster-Vertex	137452.43	7761.79	1490762.07	10808.81	53.78

The analysis of the Fixed-Direction method reveals that the Cluster-Vertex selection type appears to be the overall best choice. While the Cluster-Vertex approach does not produce the absolute minimum values for energy (E) and losses (Loss), which are shared with the Random-Vertex selection, it achieves remarkably close results.

Specifically, the Cluster-Vertex method has an energy (E) of $7,499.14$ and losses (Loss) of 52.36, which are only slightly higher than the optimal values. Furthermore, the Cluster-Vertex solution for the power generated (P) of 10762.04 is exceptionally close to the best-known minimum of 10745.42 obtained using the Inner-Point-Vertex approach, with a difference of only 0.15%. For the maximum utility, the Ideal-Point-Vertex selection type has been identified as the optimal choice, as it generates the maximum utility (U) of 1510324.25.

The optimal vertex selection for the 'Fixed-Point Direction' and 'Ideal-Point Direction' methods hinges on the specific priorities and needs of the application. If maximizing utility (U) is the primary goal, then the 'Ideal-Point-Vertex' delivered the highest utility value of 1540260.96. Conversely, if the focus is on minimizing energy (E) and losses (Loss), the 'Cluster-Vertex' selection stands out as the preferred option, achieving the lowest values for both E (7553.76) and Loss (52.82). Additionally, demonstrates a noteworthy capability in minimizing the cost (C) at 136366.18 and the power (P) at 10764.40.

Comparing the performance of the SGSD-DEED model to the results obtained by applying the Enhanced Approximation Algorithm offers a valuable perspective. For the Fixed-Direction method, we selected the Cluster-Vertex as the optimal vertex selection type. The analysis reveals that the Enhanced Approximation Algorithm de-

Model	C	E	P	Loss
Enhanced Approximation Algorithm with FD	137424.27	7499.14	10762.04	52.36
SGSD-DEED	142 959.81	8093.74	11259.04	54.87

Table 6.7: Comparing results Scenario 1

livers a compelling performance advantage over the SGSD-DEED model. Specifically,

the Enhanced Approximation Algorithm demonstrates a 3.87% lower cost, 7.32% lower emissions, 4.41% lower power generation, and 4.57% reduced losses compared to the SGSD-DEED model. These notable percentage improvements across the key metrics of cost, energy, power, and losses underscore the superior optimization capabilities of the Enhanced Approximation Algorithm.

CHAPTER 7

CONCLUSION AND FUTURE WORKS

In this chapter, the key findings and contributions of this research are summarized, reflecting on how the objectives set forth at the beginning of the study have been achieved. Additionally, this chapter outlines potential avenues for future research, suggesting how the work can be extended or applied to new contexts to further advance the field.

- We conducted a comprehensive literature review of approximation algorithms for VOP, with a particular focus on the outer approximation algorithm of CVOP. This foundational work allowed us to explore existing methodologies and develop a new enhanced outer approximation algorithm to address the challenges associated with CVOP. We performed an extensive study summarizing various vertex selection and direction methods associated with outer approximation.

- Our contributions include refinements to the PS scalarization method that sig-

nificantly enhances the algorithm's performance and the development of novel direction techniques. This achievement demonstrates the algorithm's effectiveness in tackling complex optimization problems. Notably, our enhancements lead to a significant reduction in the number of scalarization problems solved in each iteration and overall computation time, resulting in a faster optimization process and improved exploration of the solution space.

- The improved framework has shown remarkable effectiveness in real-world applications, particularly in smart grid demand response, dynamic economic dispatch, and dynamic economic emissions dispatch. These applications are critical for efficient energy management within power systems, where economic dispatch and demand-side management are essential. Our ability to navigate these challenging problems opens up new opportunities across various fields, leading to substantial reductions in energy losses, decreased electricity generation needs, and lower emissions and fuel costs. This not only benefits utility companies but also contributes to broader environmental and economic goals.

- Throughout this study, we underscore the significant potential of smart grid technology, enabling customers to engage actively in the energy system with a new model. The new model, SGSD-DEED, modifies the DR-DEED formulation by incorporating a smart grid demand response program. By utilizing tools such as demand response management, renewable energy generation, energy storage, and the ability to trade excess energy, the smart grid fosters a more decentralized and participatory energy environment. We made improvements to enhance the

utility objective, focusing on increased efficiency, reduced costs and emissions, improved reliability, and enhanced overall customer satisfaction.

- We published two packages using the Julia programming language: one focused on outer approximation (OAFCVX.jl), and the other on the new model for dynamic economic emission dispatch within smart grids, incorporating shifted demand (DRDeed.jl). Additionally, two papers have been submitted regarding this work.

In conclusion, this thesis not only advances the theoretical framework of convex vector optimization but also offers practical tools for tackling critical real-world issues. Looking ahead, several promising avenues for further research arise from our findings. We could refine the outer approximation algorithm to improve its performance on larger-scale problems with greater efficiency. This may involve strengthening the mathematical foundations of the algorithm to enhance its convergence properties and robustness, allowing it to better address complex optimization scenarios. Additionally, we aim to apply this work to various multi-objective domains, such as environmental management, transportation planning, and healthcare, which could yield significant benefits.

APPENDIX A

REPRODUCING THE RESULTS

A.1 OAFCVX.jl

Algorithm 3.3.2 and all direction and vertex selection methods were implemented in Julia and published as a stand-alone package. The package is called OAFCVX.jl [62] (Outer Approximation For Convex Vector Optimization Problems). The code can be found on GitHub at the address https://github.com/mmogib/OAFCVX.jl. To illustrate how one can use this pacake, we consider the following example.

Example (Example 3.1): Solve the following VOP.

$$\begin{aligned} &\min && x \\ &\text{subject to} \\ & && x \in \Psi = \{x \in \mathbb{R}^4 \mid \|x - e\| \leq 1, x \geq 0\} \end{aligned}$$

```
# (1) Defining the function and the dimensions (primal and dual) of the prob
f(x) = x
ddim = pdim = 4 # same since the objective function here is the identity fu
# (2) The matrix whose columns generate C
Z = Float64.(Matrix(I(ddim)))  # generating C
```

```
C = NMCone(Z)
# (3) The matrix whose columns generate C+
Zplus = copy(Z) # generating C+
Cp = NMCone(Z)

# (4) Defining the direction inside C
d = ones(ddim)
d = d / norm(d)

# (5) Pascoletti-Serafini scalarization
function ps(v::Vector{<:Number}, d::Vector{<:Number})
  model = Model(solver)
  @variable(model, z)
  @variable(model, x[1:pdim] >= 0)
  @NLconstraint(model, sum((x[i] - 1)^2 for i = 1:pdim) <= 1)
  @constraint(model, Zplus * (v + z * d - f(x)) .>= 0)
  @objective(model, Min, z)
  optimize!(model)
  if (has_values(model))
    xv = value.(x)
    return xv, value.(z)
  else
    return nothing
  end
end

# (6) To find the realized approximation error is dH(P K, P)
function ps2(v::Vector{<:Number})
  model = Model(solver)
  @variable(model, z[1:ddim])
  @variable(model, x[1:pdim] >= 0)
  @NLconstraint(model, sum((x[i] - 1)^2 for i = 1:pdim) <= 1)
  @constraint(model, Zplus * (v + z - f(x)) .>= 0)
  @NLobjective(model, Min, sum(z[i]^2 for i = 1:ddim))
  optimize!(model)
  if (has_values(model))
    return norm(value.(z))
  else
    return nothing
  end
end

# (7) The Lagrange dual of the PS problem
function dps(v::Vector{<:Number}, d::Vector{<:Number})
  model = Model(solver)
  @variable(model, t)
  @variable(model, x[1:pdim] >= 0)
  @variable(model, w[1:ddim])
  @constraint(model, dot(w, d) == 1)
  @constraint(model, Z * w .>= 0)
  @NLconstraint(model, sum((x[i] - 1)^2 for i = 1:pdim) <= 1)
  @constraint(model, dot(f(x), w) >= t)
  @objective(model, Max, t - dot(w, v))
  optimize!(model)
  if (has_values(model))
    return value.(w)
  else
    return nothing
  end
end

# (8) the weighted sum scalarization
function ws(w::Vector{<:Number})
```

```
  model = Model(solver)
  @variable(model, x[1:pdim] >= 0)
  @NLconstraint(model, sum((x[i] - 1)^2 for i = 1:pdim) <= 1)
  @objective(model, Min, dot(w, f(x)))
  optimize!(model)
  if (has_values(model))
    return value.(x)
  else
    return nothing
  end
end

# (9) Get the ideal point using the weighted sum scalarization
function getIdealPoint()
  yI = map(1:ddim) do j
    model = Model(solver)
    @variable(model, x[1:pdim] >= 0)
    @NLconstraint(model, sum((x[i] - 1)^2 for i = 1:pdim) <= 1)
    @objective(model, Min, f(x)[j])
    optimize!(model)
    if (has_values(model))
      xv = value.(x)
      return xv[j]
    else
      return nothing
    end
  end
  return yI
end

# (10) The ideal point
yI = abs.(getIdealPoint())
# (11) The problem
problem = MOAProblem(pdim, ddim, C, Cp, f, ws, ps, dps, d, yI)
X, t = @timed paa(direction, vertex, problem, options) # X is the solution

# Xs[1].message: contains a message from the solver
# Xs[1].X: the solution
# Xs[1].Vk: An array of vertices of the image set.
# Xs[1].Pk: A polyherdon structure (via Polyhedra.jl package)
```

A.2 DRDeed.jl

All models of Chapters 4, 5, and 6 were implemented in Julia and published in a stand-alone package DRDeed.jl [61]. To show how to use it, consider the model discussed in Chapter 6 with **5 customers, 6 generators and 24 periods (hours)**.

```
customers, generators, periods = 5, 6, 24
ddim = 3
our_model = gtdrdeed(customers, generators, periods)
pdim = generators * periods + customers * (7 * periods + 2) + customers * g
```

```
ws = our_model[:ws]
ps = our_model[:ps]
dps = our_model[:dps]
f = our_model[:f]

function PS1()
  return function ps_fun(v::Vector{<:Number}, d::Vector{<:Number})
    X, z = ps(v, d)
    if (isnothing(X))
      return nothing
    else
      return X[:], z
    end
  end
end

function DPS1()
  return function dps_fun(v::Vector{<:Number}, d::Vector{<:Number})
    w = dps(v, d)
    return w
  end

end

function WS1()
  return function scalarization(w::Vector{<:Number})
    sol = ws(w)
    q = sol.solution.deedSolution.q
    p = sol.solution.p
    s = sol.solution.s
    Chi = sol.solution.deedSolution.Chi
    omega = sol.solution.deedSolution.omega
    x = sol.solution.x
    f = sol.solution.f
    h = sol.solution.h
    y = sol.solution.y
    x = vcat(q[:], p[:], s[:],  [:],  [:], x[:], f[:], h[:], y[:])
    if (isa(sol, SuccessResult))
      return x
    else
      return nothing
    end
  end
end

function getIdealPoint()
  sol1 = ws([1.0; 0.0; 0.0])
  sol2 = ws([0.0; 1.0; 0.0])
  sol3 = ws([0.0; 0.0; 1.0])
  yI = vcat(
    sol1.solution.deedSolution.Cost,
    sol2.solution.deedSolution.Emission,
    sol3.solution.deedSolution.Utility,
  )
  yI
end

Z = Float64.(Matrix(I(ddim)))
C = NMCone(Z)
Cp = NMCone(Z)
d = ones(ddim)
d = d / norm(d)
yI = getIdealPoint()
```

```
problem = MOAProblem(pdim, ddim, C, Cp, f, WS1(), PS1(), DPS1(), d, yI)

options = MOAOptions('\epsilon', maxiters, (c, crd) -> begin
  cond1 = c >= maxiters
  cond2 = crd >= 10
  return cond1 || cond2
end)
X, t = @timed paa(FixedDirection, RandomVertex, problem, options)
```

APPENDIX B

SUPPLEMENTARY MATERIAL

		Direction	Vertex	SC	T
Example1	Dimintion 2				
		(FD)	(Rn)	12	0.235
			(Id)	-	-
			(In)	-	-
			(Cl)	20	1.64
		(FP)	(Rn)	129.2	111.554
			(Id)	36	30.368
			(In)	-	-
			(Cl)	16	0.55
		(IP)	(Rn)	72	0.90
			(Id)	-	-
			(In)	-	-
			(Cl)	88	0.71
		(AP-Adj)	(Rn)	8	0.07
			(Id)	-	-
			(In)	-	-
			(Cl)	20	0.11
Example1	Dimintion 4				
		(FD)	(Rn)	94.4	1.001
			(Id)	-	-
			(In)	-	-
			(Cl)	142	1.47
		(FP)	(Rn)	103.6	1.027
			(Id)	-	-
			(In)	-	-
			(Cl)	554	5.57

		Direction	Vertex	SC	T
Example1	Dimintion 3				
		(FD)	(Rn)	41	0.625
			(Id)	-	-
			(In)	-	-
			(Cl)	45	0.29
		(FP)	(Rn)	49	0.69
			(Id)	-	-
			(In)	-	-
			(Cl)	97	0.64
		(IP)	(Rn)	564	10.22
			(Id)	-	-
			(In)	-	-
			(Cl)	1687	13.69
		(AP-Adj)	(Rn)	35	0.27
			(Id)	-	-
			(In)	-	-
			(Cl)	47	2.34
Example2	$\epsilon = 0.005$				
		(FD)	(Rn)	14	0.858
			(Id)	-	-
			(In)	-	-
			(Cl)	254	5.29
		(FP)	(Rn)	19	0.54
			(Id)	-	-
			(In)	-	-
			(Cl)	308	7.74

		(IP)	(Rn)	1350.0	34.387			(IP)	(Rn)	15	0.35
			(Id)	-	-				(Id)	-	-
			(In)	-	-				(In)	-	-
			(Cl)	1736	17.80				(Cl)	252	5.71
		(AP-Adj)	(Rn)	85.2	1.125			(AP-Adj)	(Rn)	21	0.465
			(Id)	-	-				(Id)	-	-
			(In)	-	-				(In)	-	-
			(Cl)	208	3.9213				(Cl)	254	11.465
Example3	a=5,$\epsilon = 0.05$					Example3	a=7,$\epsilon = 0.05$				
		(FD)	(Rn)	5	6.23			(FD)	(Rn)	5	0.102
			(Id)	5	6.31				(Id)	5	0.11
			(In)	5	5.70				(In)	5	0.10
			(Cl)	5	5.51				(Cl)	5	0.10
		(FP)	(Rn)	5	0.72			(FP)	(Rn)	5	1.74
			(Id)	5	0.50				(Id)	5	5.28
			(In)	5	0.59				(In)	5	4.80
			(Cl)	5	0.78				(Cl)	5	1.82
		(IP)	(Rn)	5	5.80			(IP)	(Rn)	5	5.34
			(Id)	5	5.93				(Id)	5	6.05
			(In)	5	5.96				(In)	5	6.12
			(Cl)	5	5.65				(Cl)	5	5.47
		(AP-Adj)	(Rn)	5	5.14			(AP-Adj)	(Rn)	5	1.67
			(Id)	5	5.75				(Id)	5	1.56
			(In)	5	5.20				(In)	5	1.60
			(Cl)	5	5.23				(Cl)	5	1.66

Table B.1: Compares the Results of the Work by Rem Nur Keskin and the Enhanced Outer Approximation in Mathematics Example

Customers	Generators	Cost	Emission	Utility	Power Load Reduction	Loss
50	4	2.41844e5	51202.7	4.27483e5	22.1727	621.204
	8	455159.0	71835.0	1.20697e6	26.6116	536.609
	12	9.09425e5	1.52323e5	9.10228e5	10.4191	875.694
	16	1.42895e6	2.18096e5	9.25364e5	-0.53481	1092.53
	20	1.87708e6	3.00257e5	1.04759e6	-1.04443	1044.79
100	4	1.78604e5	37484.5	3.78417e5	40.0361	366.888
	8	4.56768e5	57459.6	5.9435e5	31.3451	512.932
	12	8.07607e5	1.19095e5	5.86936e5	18.4996	696.365
	16	1.30546e6	1.97634e5	511450.0	3.00497	897.525
	20	1.76955e6	2.5975e5	6.18877e5	2.75404	916.887
150	4	1.35337e5	18930.2	409218.0	58.7838	171.98
	8	4.67344e5	65409.9	6.01151e5	28.6752	498.552
	12	8.33085e5	1.2371e5	524357.0	16.5745	681.297
	16	1.32156e6	2.15662e5	4.14091e5	4.46811	891.707
	20	1.73837e6	2.786e5	4.64216e5	3.62162	848.524
200	4	1.24073e5	19962.8	4.13954e5	63.4141	142.61
	8	4.56015e5	73583.5	480642.0	27.3441	524.447
	12	8.08863e5	1.36576e5	5.84168e5	25.4518	598.368
	16	1.27931e6	1.92261e5	3.85515e5	6.70606	776.992
	20	1.7516e6	271704.0	4.58137e5	5.22481	855.964
250	4	78044.6	13731.8	4.30353e5	76.0048	62.2114
	8	3.98319e5	68432.5	5.64685e5	36.9804	400.239
	12	8.84467e5	1.45205e5	3.56302e5	10.9215	777.285
	16	1.33079e6	1.89973e5	3.38509e5	5.39039	851.435
	20	1.80768e6	2.65104e5	3.83101e5	4.8625	899.231
300	4	1.00434e5	15760.6	4.66149e5	68.0979	105.367
	8	3.23487e5	50224.3	7.19909e5	52.7182	242.796
	12	8.39066e5	1.43104e5	3.69374e5	14.5316	697.087
	16	1.36024e6	1.96631e5	3.37713e5	6.97863	925.416
	20	1.83888e6	2.69629e5	3.6212e5	4.55027	883.609
350	4	99291.2	13880.7	4.59955e5	71.006	106.415
	8	3.78109e5	57035.2	6.45181e5	43.7955	339.303
	12	9.05618e5	1.22039e5	295011.0	10.4562	758.666
	16	1.35751e6	2.05766e5	2.99672e5	5.15944	874.5
	20	1.76523e6	2.71834e5	3.1968e5	5.20916	871.626
400	4	1.035e5	16335.3	509185.0	70.8421	97.2503
	8	442456.0	55741.5	4.73481e5	31.7949	464.385
	12	7.94787e5	1.15675e5	3.72551e5	19.1577	637.722
	16	1.28993e6	190653.0	3.19329e5	8.30491	807.976
	20	1.78093e6	2.83917e5	3.07186e5	5.08603	813.606

Table B.2: Results of Experiment 2.

i	$a^{\cdot}(\$/h)$	$b^{\cdot}(\$/h)$	$c^{\cdot}(\$/h)$	$\nu^{\cdot}(\$/h)$	$d^{\cdot}(\$/h)$	$e^{\cdot}(\$/h)$	$(ah)^{\cdot}(\$/h)$	$(af)^{\cdot}(\$/h)$	$p_i^{\min}$	$p_i^{\max}$	DM
1	45.67791083	1.636056887	0.001662884	47.3528901	1.636056887	0.001662884	45.67791083	47.77163492	3	4	117.
2	39.75098342	1.956973415	0.001501585	41.42596269	1.956973415	0.001501585	39.75098342	41.84470751	3	4	86.3
3	39.46223396	1.859715154	0.001711129	41.13721323	1.859715154	0.001711129	39.46223396	41.55595805	4	6	112.
4	38.48519085	2.441243377	0.001587138	40.16017012	2.441243377	0.001587138	38.48519085	40.57891494	3	6	104.
5	42.72294069	2.222259013	0.001484345	44.39791996	2.222259013	0.001484345	42.72294069	44.81666478	4	7	107.

Table B.3: Customer cost coefficient (Test System 1).

i	$a_i^{\cdot}$	$b_i^{\cdot}$	$c_i^{\cdot}$	$\nu_i^{\cdot}$	$d_i^{\cdot}$	$e_i^{\cdot}$	$ah_i^{\cdot}$	$af_i^{\cdot}$	$p_i^{\min}$	$p_i^{\max}$	CDe
1	74.030 527 11	3.218 430 292	0.004 991 017	83.183 927 22	3.218 430 292	0.004 991 017	64.877 126 99	87.760 627 27	2	3	93.5
2	116.380 125 9	2.501 200 786	−0.003 466 97	125.533 526	2.501 200 786	−0.003 466 97	107.226 725 8	130.110 226	3	6	102.
3	109.926 653 1	2.825 177 138	0.006 133 988	119.080 053 2	2.825 177 138	0.006 133 988	100.773 252 9	123.656 753 2	2	3	160.
4	72.323 922 56	2.973 791 076	0.004 077 454	81.477 322 67	2.973 791 076	0.004 077 454	63.170 522 44	86.054 022 72	3	6	129.
5	86.867 413 28	2.853 089 905	−0.002 711 57	96.020 813 39	2.853 089 905	−0.002 711 57	77.714 013 17	100.597 513 4	5	6	123.
6	125.269 662 4	2.930 246 665	0.001 190 8	134.423 062 5	2.930 246 665	0.001 190 8	116.116 262 3	138.999 762 6	2	3	133.
7	164.369 026 2	2.611 366 473	0.006 695 541	173.522 426 3	2.611 366 473	0.006 695 541	155.215 626 1	178.099 126 4	2	3	107.

Table B.4: Customer cost coefficient (Test System 2).

	Cost	Emission	Power	Loss
BC	122253.117	6329.699	9259.294	36.843
C2	120902.399	6176.966	9119.999	35.154
C3	124053.578	6170.417	9381.137	37.052
C4	122226.710	6304.627	9246.382	36.787

Table B.5: Optimal results with various weighting factor values (Test System 1).

	Cost	Emission	Power	Loss
BC	519489.1663	54 642.511 01	16 711.074 45	222.620 811 9
C2	580775.1568	76 195.966 06	19 758.860 46	316.212 339 7
C3	546309.5155	54 327.5535	17 815.008 61	249.948 374 1
C4	522818.0403	57 149.190 83	16 845.938 25	228.552 985 7

Table B.6: Optimal results with various weighting factor values (Test System 2).

Compare	**Cost**				**Emissions**				**Power**				**Loss**			
	BC	**C2**	**C3**	**C4**	**BC**	**C2**	**C3**	**C4**	**BC**	**C2**	**C3**	**C4**	**BC**	**C2**	**C3**	**C4**
DR-DEED	989439.22	968899.75	994464.46	1006653.84	192743.96	356332.96	183068.82	210026.09	38574.55	38682.66	38548.88	39125.16	1137.11	1254.66	1120.88	1190.38
SGSD-DEED	519489.17	580775.16	546309.52	522818.04	54642.51	76195.97	54327.55	57149.19	16711.08	19758.86	17815.01	16845.94	222.62	316.21	249.95	228.55
Percentage %	47.49	40.07	45.06	48.01	71.63	78.60	70.33	72.80	56.57	48.90	53.73	56.99	80.42	74.76	77.68	80.80

Table B.7: Comparison of costs, emissions, power generated, and loss (Test System 2).

	$\chi_{t,i}$				
Time	customer_1	customer_2	customer_3	customer_4	customer_5
1	0.35018	1.45509	3.82709	3.99999	4.99999
2	2.99999	4.07407	3.99999	3.99999	9.77741
3	3.45712	4.01505	4.64996	5.00000	5.93769
4	3.32110	3.99999	5.00000	5.00000	6.00000
5	3.34732	4.52966	5.00000	5.00000	6.00000
6	5.00000	5.00000	5.00000	5.00000	7.10045
7	5.00000	5.00000	8.81959	5.00000	6.00000
8	5.00000	5.00000	5.00000	6.70508	8.34495
9	5.00000	5.00000	7.36906	6.88036	6.17144
10	4.93396	5.48896	5.84217	6.92381	7.60821
11	3.86212	4.76043	8.55916	5.00000	7.95939
12	3.64128	4.36703	5.00000	6.11487	8.59536
13	4.36959	4.69263	5.00000	5.00000	8.46413
14	5.00000	5.00000	7.76385	5.00000	6.00000
15	5.00000	5.75769	5.00000	7.72486	6.00000
16	4.63455	5.00000	5.00000	5.00000	9.99880
17	5.00000	5.00000	5.00000	5.00000	9.94840
18	4.09914	5.00000	8.56776	5.00000	7.44778
19	5.00000	5.00000	8.42011	6.36606	6.00000
20	5.00000	5.00000	5.01047	9.83251	8.12202
21	5.00000	5.00000	5.00000	7.05399	14.67246
22	6.27398	6.90137	12.62664	11.84499	7.15226
23	2.99999	8.63973	8.28431	13.90414	27.04173
24	4.35502	2.11750	7.41512	10.76103	7.59471

Table B.8: The Optimal Customer Power Curtailed (Test System 1)

	$\omega_{t,i}$				
Time	customer_1	customer_2	customer_3	customer_4	customer_5
1	4.30262	16.90587	38.21025	27.44160	21.98500
2	51.54300	62.03770	40.68282	27.44160	84.06871
3	62.31565	60.81234	50.55446	38.86400	31.00432
4	59.02947	60.50143	56.24852	38.86400	31.65840
5	59.65772	71.81888	56.24852	38.86400	31.65840
6	104.37500	82.51679	56.24852	38.86400	44.33617
7	104.37500	82.51679	135.56638	38.86400	31.65840
8	104.37500	82.51679	56.24852	62.54851	61.23988
9	104.37500	82.51679	101.73660	65.28384	33.49341
10	102.39465	94.28460	71.03146	65.97072	50.90396
11	72.50494	76.99159	129.15793	38.86400	55.71166
12	66.87371	68.26169	56.24852	53.74975	64.97034
13	86.12747	75.45664	56.24852	38.86400	63.00147
14	104.37500	82.51679	110.49442	38.86401	31.65840
15	104.37500	101.03280	56.24852	79.24913	31.65840
16	93.61787	82.51679	56.24852	38.86400	87.91882
17	104.37500	82.51679	56.24852	38.86400	87.03486
18	78.74900	82.51679	129.36740	38.86400	48.77983
19	104.37500	82.51679	125.79649	57.41687	31.65840
20	104.37500	82.51679	56.42295	119.77954	58.01150
21	104.37500	82.51679	56.24852	68.04898	189.31816
22	145.73248	131.97803	245.95238	166.04515	44.98561
23	51.54300	185.91758	122.55361	221.03311	643.06572
24	85.72303	26.53499	102.74107	140.20735	50.72352

Table B.9: Optimal Customer Incentive(Test System 1)

	$p_{t,i}$				
Time	customer_1	customer_2	customer_3	customer_4	customer_5
1	5.00000	5.00000	5.00000	5.00000	6.00000
2	5.00000	5.00000	5.00000	5.00000	6.00000
3	4.80589	4.00000	4.00000	5.00000	4.99999
4	3.00000	3.99999	4.00000	3.99999	6.00000
5	3.00000	4.00000	4.75354	4.00000	4.99999
6	3.00000	4.00000	5.00000	5.00000	5.60488
7	5.00000	5.00000	5.00000	5.00000	6.00000
8	5.00000	5.00000	5.00000	5.00000	6.00000
9	5.00000	5.00000	5.00000	5.00000	6.00000
10	5.00000	5.00000	5.00000	5.00000	6.00000
11	5.00000	5.00000	5.00000	5.00000	6.00000
12	5.00000	5.00000	5.00000	5.00000	6.00000
13	5.00000	5.00000	5.00000	5.00000	6.00000
14	5.00000	5.00000	5.00000	5.00000	6.00000
15	5.00000	5.00000	5.00000	5.00000	6.00000
16	5.00000	5.00000	5.00000	5.00000	6.00000
17	5.00000	5.00000	5.00000	5.00000	6.00000
18	5.00000	5.00000	5.00000	5.00000	6.00000
19	5.00000	5.00000	5.00000	5.00000	6.00000
20	5.00000	5.00000	5.00000	5.00000	6.00000
21	5.00000	5.00000	5.00000	5.00000	6.00000
22	3.00000	4.00000	4.00000	4.00000	4.99999
23	3.00000	4.00000	4.00000	4.00000	4.99999
24	3.00000	4.00000	4.00000	4.00000	4.99999

Table B.10: Optimal Power Generated by the Customer $p_{i,t}$ (Test System 1)

	$s_{t,i}$				
Time	customer_1	customer_2	customer_3	customer_4	customer_5
1	3	4	4	4	5
2	3	4	4	4	5
3	3.457121	4.01505	4.649963	5	5.937695
4	3.3211	4	5	5	6
5	3.347324	4.529664	5	5	6
6	5	5	5	5	6
7	5	5	5	5	6
8	5	5	5	5	6
9	5	5	5	5	6
10	4.933962	5	5	5	6
11	3.862123	4.760433	4.033766	5	6
12	3.64128	4.367034	5	5	5.127527
13	4.369593	4.692633	5	5	6
14	5	5	5	5	6
15	5	5	5	5	6
16	4.634546	5	5	5	6
17	5	5	5	5	6
18	4.099139	5	5	5	6
19	5	5	5	5	6
20	5	5	5	5	6
21	5	5	5	5	5.778519
22	5	5	5	5	6
23	3	4	4	4	5
24	0	0	0	0	0

Table B.11: Optimal Power Stored by the Customer $s_{i,t}$ (Test System 1)

	$h_{t,i}$				
Time	customer_1	customer_2	customer_3	customer_4	customer_5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0.01047	4.83251	2.12202
21	0	0	0	2.054	8.89394
22	1.27399	1.90137	7.62664	6.84499	1.15226
23	0	4.63973	4.28431	9.90414	22.04173
24	44.84113	2.1175	7.41512	10.76104	7.59471

Table B.12: Optimal Demand to be Shifted by the Customer $h_{i,t}$ (Test System 1)

	$f_{t,i}$				
Time	customer_1	customer_2	customer_3	customer_4	customer_5
1	105.84735	103.22329	130.946	105.184	115.29143
2	121.33344	129.70847	99.25969	127.76515	103.41395
3	87.69110	114.45529	123.59326	104.64888	115.15896
4	92.01830	110.85214	74.22101	109.10073	102.43442
5	101.19212	85.64159	73.48176	115.14573	112.34772
6	122.62246	127.35795	95.03540	121.81259	127.80650
7	110.25819	127.82201	100.59248	109.60529	116.90224
8	99.51553	115.49759	111.19354	135.90984	108.44558
9	124.91934	118.51066	120.20847	99.66521	122.48991
10	90.97171	113.09547	113.94646	110.94565	98.62741
11	128.26480	119.32215	115.70517	117.73838	96.16636
12	101.80571	115.17186	90.58072	126.41521	111.45808
13	113.37365	102.50652	109.15427	104.19484	104.34831
14	94.97481	91.95206	101.49323	123.64734	108.63585
15	95.40399	106.02850	93.33941	117.82649	110.06106
16	99.92139	122.62157	105.24921	110.77082	110.66807
17	95.70443	95.15325	111.51839	116.00131	123.71479
18	97.84385	126.60212	103.91007	103.84206	125.88235
19	110.37099	103.64023	116.61388	124.83040	107.84809
20	99.22637	127.98388	92.55995	117.71056	86.12230
21	122.16800	114.03070	84.16659	91.53747	86.71363
22	101.58106	119.47791	121.91621	89.18792	85.35886
23	89.22711	96.31413	103.47437	104.71281	137.03085
24	153.30893	7.99999	7.99999	101.83441	38.96917

Table B.13: Optimal Demand Satisfied by the Customer $f_{i,t}$ (Test System 1)

	$q_{t,j}$					
Time	generator_1	generator_2	generator_3	generator_4	generator_5	generator_6
1	207.96	52.50	121.49	50.00	72.54	50.00
2	201.35	56.37	123.50	50.00	74.26	50.00
3	190.09	50.63	118.00	50.00	67.08	50.00
4	166.84	50.00	101.05	50.00	50.00	50.00
5	166.70	50.00	100.92	50.00	50.00	50.00
6	203.85	62.42	128.15	50.00	79.72	50.00
7	203.49	50.00	117.78	50.00	67.92	50.00
8	206.83	50.00	118.60	50.00	69.14	50.00
9	219.31	50.00	119.38	50.00	71.11	50.00
10	191.15	50.00	106.47	50.00	53.90	50.00
11	210.58	50.00	118.73	50.00	69.61	50.00
12	195.99	50.00	112.11	50.00	60.80	50.00
13	188.72	50.00	111.38	50.00	59.40	50.00
14	184.08	50.00	107.26	50.00	54.30	50.00
15	181.63	50.00	108.96	50.00	56.08	50.00
16	192.87	50.00	115.52	50.00	64.47	50.00
17	191.65	50.00	113.15	50.00	61.66	50.00
18	197.24	50.00	117.20	50.00	66.74	50.00
19	197.55	51.56	119.60	50.00	69.50	50.00
20	182.37	50.00	109.03	50.00	56.21	50.00
21	170.53	50.00	101.86	50.00	50.00	50.00
22	179.10	50.00	110.72	50.00	57.92	50.00
23	178.95	50.00	114.07	50.00	61.74	50.00
24	100.00	50.00	80.00	50.00	50.00	50.00

Table B.14: Optimal Power Generated by Generators q_j, t (Test System 1)

REFERENCES

[1] K. Miettinen, *Nonlinear multiobjective optimization.* Springer Science & Business Media, 1999, vol. 12.

[2] Y. Sawaragi, H. NAKAYAMA, and T. TANINO, *Theory of multiobjective optimization.* Elsevier, 1985.

[3] A. Löhne and B. Weißing, "Bensolve-vlp solver, version 2.0. 1," *URL http://bensolve. org*, 2015.

[4] ——, "The vector linear program solver bensolve–notes on theoretical background," *European Journal of Operational Research*, vol. 260, no. 3, pp. 807–813, 2017.

[5] D. T. Luc, *Theory of vector optimization.* Springer, 1989.

[6] X.-B. Hu, M. Wang, and E. Di Paolo, "Calculating complete and exact pareto front for multiobjective optimization: a new deterministic approach for discrete problems," *IEEE Transactions on cybernetics*, vol. 43, no. 3, pp. 1088–1101, 2013.

[7] A. H. Hamel and A. Löhne, "Lagrange duality in set optimization," *Journal of Optimization Theory and Applications*, vol. 161, pp. 368–397, 2014.

[8] F. Heyde and A. Löhne, "Solution concepts in vector optimization: a fresh look at an old story," *Optimization*, vol. 60, no. 12, pp. 1421–1440, 2011.

[9] Ç. Ararat, F. Ulus, and M. Umer, "A norm minimization-based convex vector optimization algorithm," *Journal of Optimization Theory and Applications*, pp. 1–32, 2022.

[10] J. P. Evans and R. E. Steuer, "A revised simplex method for linear multiple objective programs," *Mathematical Programming*, vol. 5, no. 1, pp. 54–72, 1973.

[11] H. Isermann, "The enumeration of the set of all efficient solutions for a linear multiple objective program," *Journal of the Operational Research Society*, vol. 28, pp. 711–725, 1977.

[12] J. G. Ecker and I. Kouada, "Finding all efficient extreme points for multiple objective linear programs," *Mathematical programming*, vol. 14, pp. 249–261, 1978.

[13] B. Rudloff, F. Ulus, and R. Vanderbei, "A parametric simplex algorithm for linear vector optimization problems," *Mathematical Programming*, vol. 163, pp. 213–242, 2017.

[14] M. Ehrgott, J. Puerto, and A. Rodriguez-Chia, "Primal-dual simplex method for multiobjective linear programming," *Journal of optimization theory and applications*, vol. 134, no. 3, pp. 483–497, 2007.

[15] J. P. Dauer and Y.-H. Liu, "Solving multiple objective linear programs in objective space," *European Journal of Operational Research*, vol. 46, no. 3,

pp. 350–357, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377221790900109

[16] S. Ruzika and M. M. Wiecek, "Approximation methods in multiobjective programming," *Journal of optimization theory and applications*, vol. 126, no. 3, pp. 473–501, 2005.

[17] H. P. Benson, "Further analysis of an outcome set-based algorithm for multiple-objective linear programming," *Journal of Optimization Theory and Applications*, vol. 97, pp. 1–10, 1998.

[18] ——, "An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem," *Journal of Global Optimization*, vol. 13, pp. 1–24, 1998.

[19] A. Löhne, *Vector optimization with infimum and supremum*. Springer Science & Business Media, 2011.

[20] L. Shao, M. Ehrgott, and A. Löhne, "A dual variant of benson's outer approximation algorithm," *http://www.esc.auckland.ac.nz/research/tech/esc-tr-654.pdf*, 01 2007.

[21] L. Csirmaz, "Using multiobjective optimization to map the entropy region," *Computational Optimization and Applications*, vol. 63, pp. 45–67, 2016.

[22] A. H. Hamel, A. Löhne, and B. Rudloff, "Benson type algorithms for linear vector optimization and applications," *Journal of Global Optimization*, vol. 59, pp. 811–836, 2014.

[23] L. Shao and M. Ehrgott, "Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning," *Mathematical Methods of Operations Research*, vol. 68, pp. 257–276, 2008.

[24] S. Ruzika and M. M. Wiecek, "Approximation methods in multiobjective programming," *Journal of optimization theory and applications*, vol. 126, no. 3, pp. 473–501, 2005.

[25] M. Ehrgott, L. Shao, and A. Schöbel, "An approximation algorithm for convex multi-objective programming problems," *Journal of Global Optimization*, vol. 50, no. 3, pp. 397–416, 2011.

[26] A. Löhne, B. Rudloff, and F. Ulus, "Primal and dual approximation algorithms for convex vector optimization problems," *Journal of Global Optimization*, vol. 60, pp. 713–736, 2014.

[27] D. Dörfler, A. Löhne, C. Schneider, and B. Weißing, "A benson-type algorithm for bounded convex vector optimization problems with vertex selection," *Optimization Methods and Software*, pp. 1–21, 2021.

[28] I. N. Keskin and F. Ulus, "Outer approximation algorithms for convex vector optimization problems," *Optimization Methods and Software*, vol. 38, no. 4, pp. 723–755, 2023.

[29] K. Klamroth, J. Tind, and M. M. Wiecek, "Unbiased approximation in multicriteria optimization," *Mathematical Methods of Operations Research*, vol. 56, no. 3, pp. 413–437, 2003.

[30] A. Aliano Filho, A. C. Moretti, M. V. Pato, and W. A. de Oliveira, "An exact scalarization method with multiple reference points for bi-objective integer linear optimization problems," *Annals of Operations Research*, vol. 296, pp. 35–69, 2021.

[31] C. Tammer, "J. jahn: Vector optimization, theory, applications and extensions," *Jahresbericht der Deutschen Mathematiker Vereinigung*, vol. 109, no. 4, pp. 37–38, 2007.

[32] A. A. Khan, C. Tammer, and C. Zalinescu, *Set-valued optimization*. Springer, 2016.

[33] K. Fukuda and A. Prodon, "Double description method revisited," in *Franco-Japanese and Franco-Chinese conference on combinatorics and computer science*. Springer, 1996, pp. 91–111.

[34] R. T. Rockafellar, *Convex analysis*. Princeton university press, 1970, vol. 18.

[35] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: https://doi.org/10.1137/141000671

[36] M. Lubin, O. Dowson, J. Dias Garcia, J. Huchette, B. Legat, and J. P. Vielma, "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization," *Mathematical Programming Computation*, 2023.

[37] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." [Online]. Available: https://optimization-online.org/?p=9506

[38] J. F. Nash, "Equilibrium points in n-person games." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36 1, pp. 48–9, 1950.

[39] T. Cui, Y. Wang, S. Yue, S. Nazarian, and M. Pedram, "A game-theoretic price determination algorithm for utility companies serving a community in smart grid," in *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2013, pp. 1–6.

[40] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Non-cooperative and cooperative optimization of distributed energy generation and storage in the demand-side of the smart grid," *IEEE transactions on signal processing*, vol. 61, no. 10, pp. 2454–2472, 2013.

[41] ——, "Demand-side management via distributed energy generation and storage optimization," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 866–876, 2012.

[42] S. Belhaiza and U. Baroudi, "A game theoretic model for smart grids demand management," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1386–1393, 2015.

[43] K. Ullah, T. Khan, G. Hafeez, I. Khan, S. Murawwat, B. Alamri, F. Ali, S. Ali, and S. Khan, "Demand side management strategy for multi-objective day-ahead scheduling considering wind energy in smart grid," *Energies*, vol. 15, p. 6900, 09 2022.

[44] S. Belhaiza, U. Baroudi, and I. Elhallaoui, "A game theoretic model for the multiperiodic smart grid demand response problem," *IEEE Systems Journal*, vol. 14, no. 1, pp. 1147–1158, 2019.

[45] J. F. Nash Jr, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[46] N. I. Nwulu and X. Xia, "Multi-objective dynamic economic emission dispatch of electric power generation integrated with game theory based demand response programs," *Energy Conversion and Management*, vol. 89, pp. 963–974, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890414009510

[47] S. Wang, S. Bi, and Y.-J. A. Zhang, "Demand response management for profit maximizing energy loads in real-time electricity market," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6387–6396, 2018.

[48] C. Eid, E. Koliou, M. Valles, J. Reneses, and R. Hakvoort, "Time-based pricing and electricity demand response: Existing barriers and next steps," *Utilities Policy*, vol. 40, pp. 15–25, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957178716300947

[49] E. Afzalan and M. Joorabian, "Emission, reserve and economic load dispatch problem with non-smooth and non-convex cost functions using epsilon-multi-objective genetic algorithm variable," *International Journal of Electrical*

Power & Energy Systems, vol. 52, pp. 55–67, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061513001348

[50] A. Elaiw, X. Xia, and A. Shehata, "Hybrid de-sqp and hybrid pso-sqp methods for solving dynamic economic emission dispatch problem with valve-point effects," *Electric Power Systems Research*, vol. 103, pp. 192–200, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779613001430

[51] D. Aydin, S. Özyön, C. Yaşar, and T. Liao, "Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem," *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 144–153, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061513002809

[52] S. Sayah, A. Hamouda, and A. Bekrar, "Efficient hybrid optimization approach for emission constrained economic dispatch with nonsmooth cost curves," *International Journal of Electrical Power & Energy Systems*, vol. 56, pp. 127–139, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061513004596

[53] A. Arif and F. Javed, "Integrating renewables economic dispatch with demand side management in micro-grids: A genetic algorithm-based approach," *Energy Efficiency*, vol. 7, 04 2014.

[54] C. L. Chinnadurrai and T. A. A. Victoire, "Enhanced multi-objective crisscross optimization for dynamic economic emission dispatch considering demand re-

sponse and wind power uncertainty," *Soft Computing*, vol. 24, no. 12, pp. 9021–9038, 2020.

[55] A. Elaiw, X. Xia, and A. Shehata, "Application of model predictive control to optimal dynamic dispatch of generation with emission limitations," *Electric power systems research*, vol. 84, no. 1, pp. 31–44, 2012.

[56] M. Fahrioglu and F. L. Alvarado, "Designing incentive compatible contracts for effective demand management," *IEEE Transactions on power Systems*, vol. 15, no. 4, pp. 1255–1260, 2000.

[57] N. I. Nwulu and X. Xia, "Multi-objective dynamic economic emission dispatch of electric power generation integrated with game theory based demand response programs," *Energy Conversion and Management*, vol. 89, pp. 963–974, 2015.

[58] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 570–582, 2015.

[59] P. Fairley, "Energy storage: Power revolution," *Nature*, vol. 526, no. 7575, pp. S102–S104, 2015.

[60] W. Saad, Z. Han, H. V. Poor, and T. Basar, "Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications," *IEEE signal processing magazine*, vol. 29, no. 5, pp. 86–105, 2012.

[61] M. Ashahrani and N. Almuraysil, "Drdeed.jl: A julia package for demand response and dynamic economic emission dispatch," https://github.com/mmogib/DRDeed.jl, 2024.

[62] ——, "Oafcvx.jl: A julia package for outer approximation algorithms for convex vector optimization," https://github.com/mmogib/OAFCVX.jl, 2024.

VITAE

- Name: Norah Abdullatif Mohammed Almuraysil
- Nationality: Saudi
- Date of Birth: August 11, 1986
- Email: *nalmuraisel@iau.edu.sa*
- Permanent Address: 8341 Alfaruq 12 Street, 4716-35816 Al Jubail, Saudi Arabia.

- M.Sc. in Pure Mathematics from Kent State University, United States, May 2017. Thesis title:"MEASURING CONVEXITY OF A SET".
- B.Sc. in Mathematics from Imam Abdulrahman Bin Faisal University(IAU), Jubail, Saudi Arabia, April 2008.