ABSTRACT

Title of dissertation: MULTIOBJECTIVE OPTIMIZATION MODELS FOR DISTRIBUTING BIOSOLIDS TO REUSE FIELDS

Prawat Sahakij, Doctor of Philosophy, 2008

Dissertation directed by: Associate Professor Steven A Gabriel
Department of Civil and Environmental Engineering

The District of Columbia Water and Sewer Authority (DCWASA) operates the Blue Plains Wastewater Treatment Plant located in Washington, DC. It serves more than two million Washington Metro Area customers, and treats more than 330 million gallons a day of raw sewage from area jurisdictions, including Montgomery and Prince George's Counties in Maryland, and Fairfax and Loudoun Counties in Virginia. Each day, DCWASA produces approximately 1,200 tons of biosolids or byproducts of wastewater that have been treated to reduce pathogens and can be used as fertilizer for agricultural purposes. These generated biosolids require removal from the treatment facility and distribution to reuse fields located in Maryland and Virginia. In spite of the benefits of reuse, biosolids are generally considered by many as potentially malodorous. Recently, DCWASA has received complaints from the surrounding communities and needed to minimize biosolids odors. However, trying to minimize biosolids odors could result in costly treatment processes. Therefore, one needs to determine how to minimize the odors while at the same time minimizing the treatment costs. This compromise of balancing the competing objectives of odors and costs results in a two-objective or more generally, multiobjective optimization problem.

In this dissertation, we develop multiobjective optimization models to simultaneously minimize biosolids odors as well as wastewater treatment process and biosolids distribution costs. A *weighting method* and *constraint method* were employed to find tradeoff, so called *Pareto* optimal, points between costs and odors. *Schur*'s decomposition and *special order set* type two variables were used to approximate the product of two decision variables. A *Dantzig-Wolfe* decomposition technique was successfully applied to break apart and solve a large optimization model encountered in this dissertation. Using the Blue Plains advanced wastewater treatment plant as a case study, we find several Pareto optimal points between costs and odors where different treatments (e.g., lime addition) and biosolids distribution (e.g., to what reuse fields biosolids should be applied) strategies should be employed. In addition, to hedge the risk of equipment failures as well as for historical reasons, an on-site dewatering contractor has also been incorporated into the model. The optimal solutions indicate different uses of the contractor (e.g., percent flow assigned) when dewatering cost employed by DCWASA varies. This model can be used proactively by any typical advanced wastewater treatment plants to produce the least malodorous biosolids at minimal costs and to our knowledge, this is the first instance of such a model.

MULTIOBJECTIVE OPTIMIZATION MODELS FOR DISTRIBUTING BIOSOLIDS TO REUSE FIELDS

by

Prawat Sahakij

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:
Associate Professor Steven A. Gabriel, Chair/Advisor
Professor Shapour Azarm, Dean's Representative
Professor Gregory B. Baecher
Associate Professor Eric A. Seagren
Associate Professor Glenn E. Moglen

DEDICATION

To my parents, for their endless love, understanding, and most importantly, making me the person who I am today.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Dr. Steven A. Gabriel, my dissertation advisor. Without his dedication, support, and guidance this work could not have been completed.

I would like to thank DCWASA for financially supporting my research throughout my Ph.D. study.

I am grateful to Mr. Chris Peot and Mr. Mark Ramirez of DCWASA for providing me the great opportunity to work at DCWASA.

Mr. Al Razik of Maryland Environmental Service (MES) was very helpful in providing relevant data.

I would like to express my sincere thanks to Dr. Glenn E. Moglen for his valuable advice on coding in Avenue script and being on my dissertation committee.

I would also like to thank Dr. Gregory B. Baecher, Dr. Shapour Azarm, and Dr. Eric A. Seagren for serving on my dissertation committee. Their insightful comments and advice really helped to polish this dissertation.

I am very thankful to Dorothea F. Brosius of the University of Maryland for kindly and patiently helping me compile Latex files.

Lastly, I would like to thank all my friends who always encouraged me to finish my dissertation.

TABLE OF CONTENTS

List of Tables vii

List of Figures ix

1 Introduction, Literature Review, and Objectives 1
- 1.1 Introduction and Literature Review 1
- 1.2 Objectives 11

2 An introduction to Multiobjective Optimization, Dantzig-Wolfe Decomposition, and the Benders Decomposition 14
- 2.1 Multiobjective Optimization 14
  - 2.1.1 Constraint Method 18
  - 2.1.2 Weighting Method 19
- 2.2 Dantzig-Wolfe Decomposition 26
  - 2.2.1 Master Problem and Pricing Subproblems 29
  - 2.2.2 Dantzig-Wolfe Decomposition: The Overall Algorithm 33
    - 2.2.2.1 Phase 1: Initialization 33
    - 2.2.2.2 Phase 2: Master Problem Feasibility 34
    - 2.2.2.3 Phase 3: Optimization of the RLPM 39
    - 2.2.2.4 Phase 4: Calculate the solution to the original problem 41
  - 2.2.3 Dantzig-Wolfe Decomposition of an Integer Program 43
- 2.3 Benders Decomposition 44

3 Wastewater Treatment Processes, Odor Prediction Statistical Models, Data and Sources, and Odor Threshold Calculations 47
- 3.1 Wastewater Treatment Processes 47
- 3.2 Odor Prediction Statistical Models 49
  - 3.2.1 Polymer Addition 50
  - 3.2.2 Number of Centrifuges or Belt Filter Presses in Service 50
  - 3.2.3 Temperature 52
  - 3.2.4 Sludge Blanket Depth 53
  - 3.2.5 Lime Addition 53
  - 3.2.6 First Odor Prediction Statistical Model 54
  - 3.2.7 Second Odor Prediction Statistical Model 55
- 3.3 Data and Sources 56
  - 3.3.1 Centrifuge and Belt Filter Press' Operation and Maintenance Cost Data 57
  - 3.3.2 On-Site Contractor's Dewatering and Lime Stabilization Costs 57
  - 3.3.3 DCWASA's dewatering Cost 58
  - 3.3.4 Chemical Costs 58
  - 3.3.5 Biosolids' Hauling Cost 58
  - 3.3.6 Temperature Data 59
  - 3.3.7 Wastewater Treatment Processing Data 59

3.3.8 Wind Direction 61
3.3.9 Reuse Field Data 61
3.3.10 Field Tonnage Capacity 65
3.3.11 Population Density 66
3.3.12 Total Number of People 66
3.3.13 Total Length of Streets 70
3.3.14 Total Number of Schools 71
3.3.15 Total Number of Hospitals 71
3.4 Odor Thresholds for Each Reuse Field 72
3.4.1 First Set of Odor Thresholds 74
3.4.2 Second Set of Odor Thresholds 78

4 Optimization Models, Formulations, and Results 86
4.1 Problem Statement 87
4.2 Notation and Constraints 89
4.2.1 Notation 89
4.2.2 Constraints 91
4.3 "Base Case" Optimization Model 95
4.3.1 Base Case Computational Results 98
4.3.2 Analysis of the Base Case 104
4.4 Sensitivity Analysis on the Base Case 136
4.4.1 Sensitivity Analysis on the Percent Flow from Blend Tank . . 136
4.4.2 Sensitivity Analysis on Odor Threshold Input 145
4.4.3 Sensitivity Analysis on DCWASA's Operating Costs 149
4.5 Conclusion 155

5 Optimization Models with Decomposition Methods 157
5.1 Approximation of the Bilinear Cost Function 159
5.2 Optimization Model with Schur's Decomposition and SOS2 Variables 164
5.2.1 Analysis on Number of Breakpoints 169
5.2.2 Algorithm and Computational Results 172
5.3 Dantzig-Wolfe Decomposition Algorithm as Applying to Multiobjective Optimization Model for Distributing Biosolids to Reuse Fields . . 179
5.3.1 Dantzig-Wolfe Decomposition Run Results 182
5.4 Modified Optimality Gaps with Updated Lower Bounds 185
5.5 Modified Multiobjective Optimization Model for Wastewater Treatment Plant (MMOPWTP) Algorithm and Computational Results . . 187
5.5.1 Computational Results 189
5.6 Compare Results with Base Case 194
5.7 Conclusion 200

6 Conclusion and Future Work 204
6.1 Conclusion 204
6.2 Future Work 207

A Avenue Script 209

Bibliography 231

LIST OF TABLES

3.1 Temperature in degrees Fahrenheit on selected days 59

3.2 Blanket depth in feet on selected days 60

3.3 Polymer and lime additions in lbs/DTS on selected days 60

3.4 Numbers of centrifuges and belt filter presses in service on selected days 60

3.5 Five selected reuse fields 64

3.6 Total number of people, schools, and hospitals and total length of streets in downwind areas within a three-mile radius of each reuse field 65

3.7 Five selected fields with associated odor thresholds set 1 78

3.8 Five selected fields with associated odor thresholds set 2 84

4.1 Base Case Pareto optimal points 107

4.2 The decrease in odor level 108

4.3 Increases in total cost's calculations for moving from Pareto optimal points #1 to #2 112

4.4 Pareto optimal points 1 and 2's detailed solutions 113

4.5 Increases in total cost's calculations for moving from Pareto optimal points #7 to #8 116

4.6 Pareto optimal point numbers 7 and 8's detailed solutions 118

4.7 Increases in total cost's calculations for moving from Pareto optimal points #10 to #11 119

4.8 Pareto optimal point numbers 21 and 22's detailed solutions 122

4.9 Increases in total cost's calculations for moving from Pareto optimal points #21 to #22 123

4.10 Solutions for Pareto optimal point number eight and DCWASA actual application 134

4.11 Five categories of costs from the Pareto optimal points when total odors = −33.396 140

4.12 Changes in five categories of costs 142

4.13 Optimal objective function values when $F_d^{dc} = 0.2$ 144

4.14 "No Wind" Case Pareto optimal points 148

4.15 Pareto optimal points obtained when DCWASA's dewatering cost = $70/DTS 152

5.1 Number of breakpoints analysis 170

5.2 Computational results from Problem 5.34 173

5.3 Possible amount of dollars off from optimal solutions given optimality gaps 177

5.4 Objective function values obtained from DWIP algorithm 184

5.5 Modified bounds and optimality gaps 185

5.6 Additional objective function values obtained from DWIP algorithm . 190

5.7 Best bounds, modified bounds, best solutions, and run time from MMOPWTP algorithm 190

5.8 Solver and modified optimality gaps from MMOPWTP algorithm . . 192

5.9 Eight Pareto optimal points 194

5.10 Daily optimal F_d^{dc} values for the eight Pareto optimal points 197

5.11 Daily optimal solutions to F_d^{dc}, F_d^k, G_d^k, and C_d^k for Pareto optimal point: (total odor = −25.856, total cost = $3,909,966) 198

LIST OF FIGURES

1.1 Estimates of Biosolids Generation in 1972, 1998, 2000, 2005 and 2010 4

1.2 Biosolids Application for Biosolids Generated by DCWASA in 2005 and 2006: Tonnage Applied and Dollars Worth of Nutrient 4

1.3 Beneficial Use by Counties in Virginia and Maryland of Biosolids Generated by DCWASA in 2005: Tonnage Applied and Dollars Worth of Nutrient 5

1.4 Beneficial Use by Counties in Virginia and Maryland of Biosolids Generated by DCWASA in 2006: Tonnage Applied and Dollars' Worth of Nutrients 6

2.1 Example of a Pareto optimal set 17

2.2 Feasible region of the two-objective minimization problem in objective space: a weighted objective function appears as a linear indifference curve 21

2.3 Pareto optimal solutions as $\frac{w_1}{w_2}$ varies 22

2.4 Duality gap in weighting method 24

2.5 Overview of Dantzig-Wolfe decomposition technique [21] 29

2.6 Flowchart for Phase 1 35

2.7 Flowchart for Phase 2 37

2.8 Flowchart for Phase 3 40

2.9 Flowchart for Phase 4 42

3.1 Liquid Process Flow Diagram 48

3.2 Solids-Handling Process [25] 49

3.3 Typical centrifuge thickening and dewatering system [60] 51

3.4 Schematic of a belt filter press [59] 51

3.5 Dewatered solids cake dropping from belt filter press [59] 52

3.6 Five Selected Reuse Fields 63

3.7 Block group layer within a three-mile radius of Field 536 67

3.8 Block group in downwind areas within a three-mile radius of Field 536 68

3.9 A right triangle with a hypotenuse side = 1 mile and an adjacent side = 3 miles 69

3.10 Streets within a three-mile radius of Field 536 70

3.11 Schools within a three-mile radius of Field 536 71

3.12 Hospitals within a three-mile radius of Field 536 72

4.1 Solutions of 98 subproblems from Base Case solving by weighting method 102

4.2 Base Case Pareto Optimal Solutions 106

4.3 Pareto optimal total odors versus selected reuse fields' average odor thresholds 128

4.4 Pareto optimal total costs versus selected reuse fields' average odor thresholds 128

4.5 Optimal reuse fields when total odor scores equal 75.5727 and 139.684 130

4.6 Pareto optimal points from Base Case and DCWASA Case 132

4.7 Daily solutions for lime additions 133

4.8 Daily solutions for the number of the on-site contractor centrifuges in service 135

4.9 Daily solutions for the number of the on-site contractor belt filter presses in service 135

4.10 Pareto optimal points when $F_d^{dc} \in \{20\%, 30\%, \ldots, 90\%\}$ 138

4.11 Pareto optimal points from Base Case and No Wind Case 146

4.12 Pareto optimal points from Base Case and No Wind Case 148

4.13 Pareto optimal points when DCWASA's operating cost = \$80/DTS . 150

4.14 Pareto optimal points when DCWASA's operating cost = \$70/DTS . 151

4.15 Possible minimum total costs when $F_d^{dc} \in \{0.2, 0.3, \ldots, 0.8\}$ 154

5.1 Demonstration for the use of SOS2 variables 162

5.2 The on-site contractor piecewise linear dewatering cost function . . . 167

5.3 Percentage errors and number of breakpoints used 171

5.4 Algorithm to solve multiobjective program for wastewater treatment plant (MOPWTP) 174

5.5 Best bound, best solution, and optimality gap for Problem (5.34) with $\tau = -60.856$ 175

5.6 Number of solutions found and optimality gap for Problem (5.34) with $\tau = -85.856$–25.856 176

5.7 Optimality gap progressions for selected problems 178

5.8 Structure of MOPWTP problem presented by day 180

5.9 Pareto optimal points 186

5.10 Modified multiobjective optimization model for wastewater treatment plant algorithm 188

5.11 Eight Pareto optimal points 195

5.12 Comparison of Pareto optimal points when F_d^{dc} was determined endogenously and exogenously 196

5.13 Optimal F_d^{dc} for eight Pareto optimal points 198

5.14 Total number of the on-site contractor centrifuges and total odor for the eight Pareto optimal points 202

5.15 Total number of the on-site contractor belt filter presses and total odor for the eight Pareto optimal points 203

Chapter 1

Introduction, Literature Review, and Objectives

1.1 Introduction and Literature Review

A new era of municipal wastewater treatment began in 1972 when the Clean Water Act imposed minimum treatment requirements for municipal wastewater [58]. The Act gave the U.S. Environmental Protection Agency (EPA) a broad authority to handle water pollution [63]. Under this authority, EPA has developed several regulations and programs to reduce pollutants entering all surface waters (e.g., lakes, rivers, estuaries, oceans, and wetlands). For example, a sewage treatment plant must obtain a permit from the EPA or a state in order to discharge pollutants into the waters [63]. Later in 1987, Congress amended section 405 of the Act requiring the EPA to establish a comprehensive program to minimize environmental risk and maximize the beneficial use of biosolids [63]. As a result, in 1993, the EPA promulgated Title 40, Code of Federal Regulations, Part 503, "Standards for the Use or Disposal of Sewage Sludge" (referred to later as Part 503) [56, 63]. Part 503 requires that use or disposal of biosolids may be allowed in three circumstance: land application, disposal in landfills, and incineration. Land applications of biosolids include spraying or spreading biosolids on the land surface, injecting them below the surface, and incorporating them into the soil [63]. In addition, there are general requirements, limits on the pollutant concentrations in biosolids, management practices, and operational requirements associated with each use or disposal method [63].

Considering land applications, the quality of biosolids are major requirements and are determined by their characteristics including level of pollutants (metals), the presence of pathogens, and the degree of attractiveness to vectors (e.g., rodents and flies). Part 503 distinguishes two classes of biosolids depending on their levels of pathogen reduction:

Class A: Part 503 requires that pathogens in these biosolids are below the detectable level. In addition, Class A biosolids with exceptional quality (EQ) must meet the most rigorous metals limits. There are no restrictions on land applications of EQ biosolids. While some restrictions may apply on Class A biosolids without the exceptional quality.

Class B: Pathogens in Class B biosolids are significantly reduced, however, not below detectable levels. Consequently, Part 503 requires farm management practices and sets site restrictions when applying these biosolids.

Over the past 30 years, the quantity of biosolids production as well as their quality substantially increased. Stringent wastewater treatment requirements and a growing population have resulted in production increases [63]. Whereas, the greater prevalence of pretreatment and pollutant prevention programs have contributed greatly to the environmentally quality improvement [58]. In addition, new technologies such as an advanced mechanical dewatering equipment, automated process control systems, aeration systems, and odor control systems also help reduce volume of biosolids, lower odor, and produce biosolids that help improve soil fertility and tilth [58, 64]. These aforementioned reasons as well as the suggestions from federal and state rules and guidance have increased the trend of biosolids beneficial uses (e.g., land applied, composted, and landfill cover) rather than disposal (i.e., with no beneficial use) [58]. Other factors contributing to increased biosolids beneficial use include a landfill capacity limit, outreach and marketing attempts, prohibitions on disposal of biosolids in landfills in some areas, and continuing research into the safe beneficial use of biosolids [58]. In addition, an increase in tipping fees by landfill operators also made biosolids disposal more difficult and expensive [20] and leads to an increased trend in uses of biosolids as well.

EPA estimated the beneficial uses of biosolids to 60%, 63%, 66%, and 70% of all biosolids production, respectively in 1998, 2000, 2005, and 2010 [58]. While the estimations of all biosolids production during these years were 6.9, 7.1, 7.6, and 8.2 million dry tons[1], respectively. This results in an approximation of 4.14, 4.47, 5.02, and 5.74 million dry ton of biosolids being used beneficially. Figure 1.1 displays the estimations of the beneficial uses as well as the total production of biosolids in millions of dry tons.

Figure 1.2 displays at DCWASA alone, the amount in dry tons of biosolids that were beneficially used (e.g., land applied) as well as the dollars' worth of nutrients provided to farmers, in 2005 and 2006. In particular, the dollars' worth of nutrients were calculated according to 20% lime and 4% nitrogen in one dry ton of biosolids with dollars worth of $18.5 and $189.33 per ton of lime and nitrogen, respectively (C. Peot, personal communication, January 7, 2002). In addition, Figures 1.3 and 1.4

[1] A dry ton is measured from solids only (i.e., not including liquid substances).

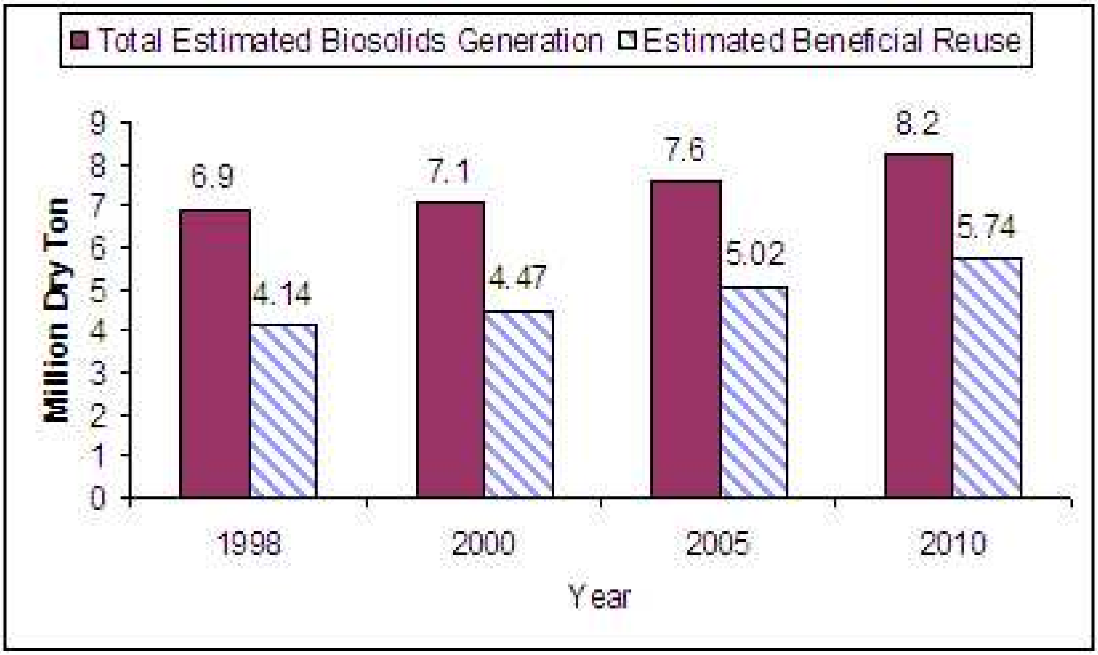


Figure 1.1: Estimates of Biosolids Generation in 1972, 1998, 2000, 2005 and 2010

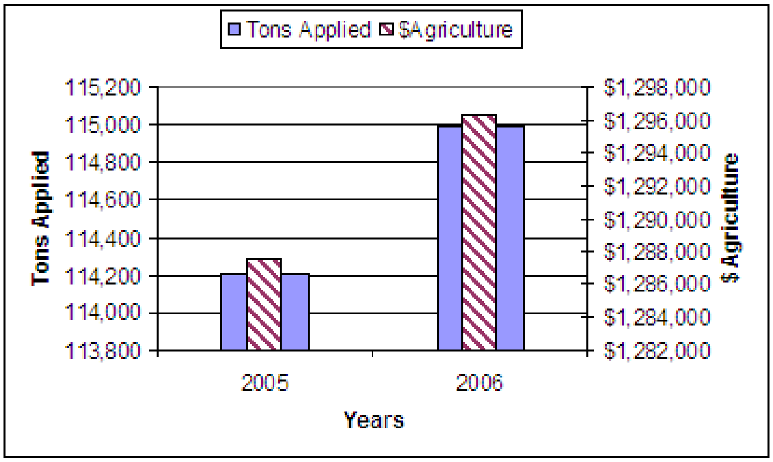


Figure 1.2: Biosolids Application for Biosolids Generated by DCWASA in 2005 and 2006: Tonnage Applied and Dollars Worth of Nutrient

display biosolids application by counties in Virginia and Maryland, whose farmers received biosolids from DCWASA in 2005 and 2006.

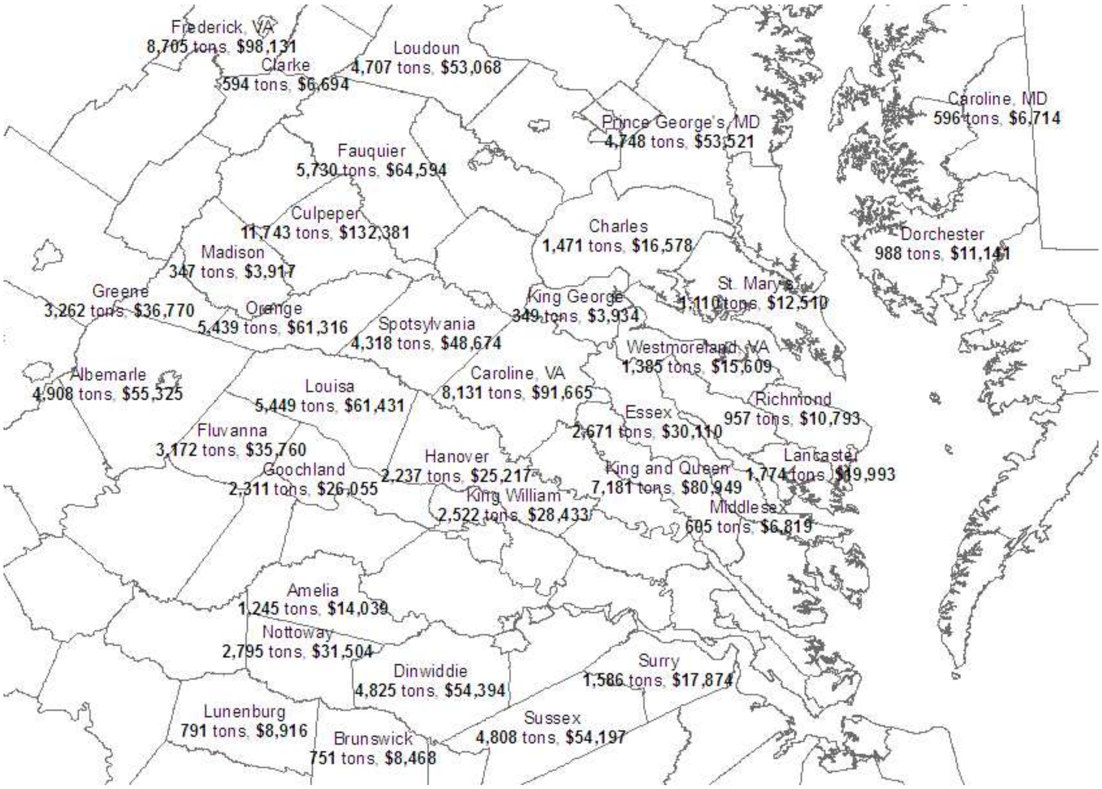


Figure 1.3: Beneficial Use by Counties in Virginia and Maryland of Biosolids Generated by DCWASA in 2005: Tonnage Applied and Dollars Worth of Nutrient

Land applications of biosolids contribute greatly to the farmers in terms of nutrients provided. However, in spite of these benefits and being carefully regulated by EPA to protect human health and the environment [28, 55, 57], these biosolids are considered by many as a potentially malodorous product to the local receiving

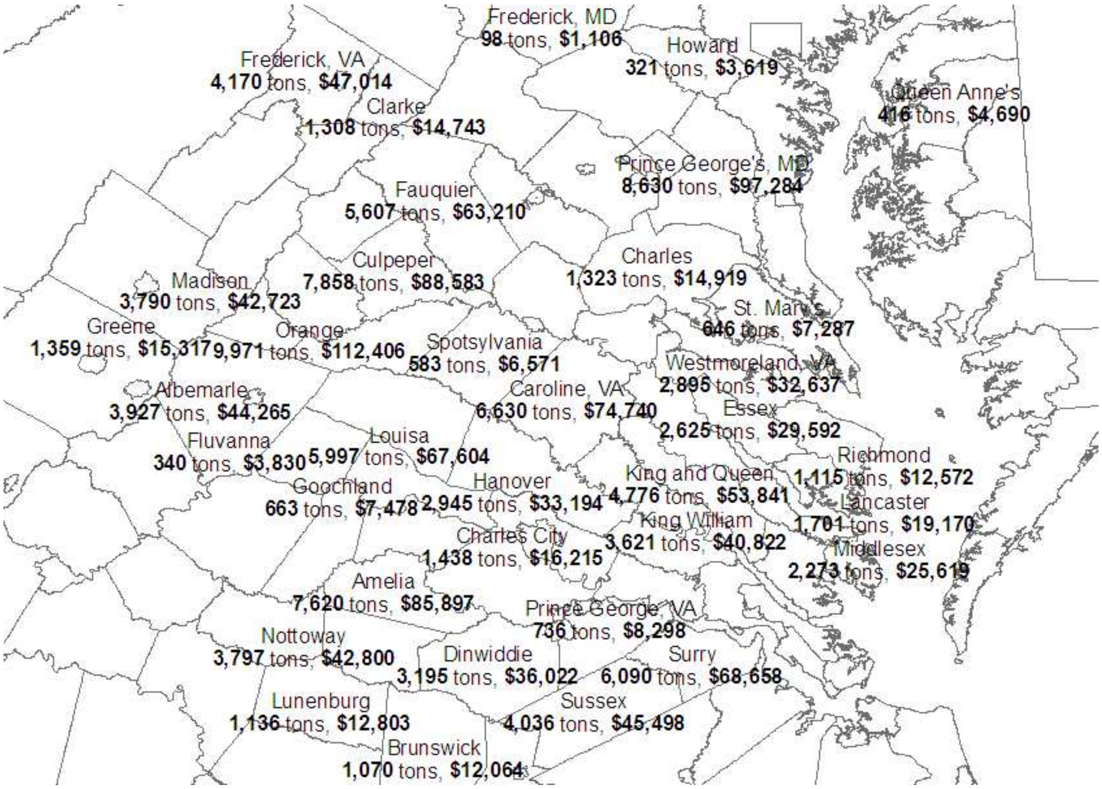


Figure 1.4: Beneficial Use by Counties in Virginia and Maryland of Biosolids Generated by DCWASA in 2006: Tonnage Applied and Dollars' Worth of Nutrients

communities. EPA officials have quoted the following statement from one of the land application companies when they made a status report on land application of biosolids in 2002 [63].

"...better tracking of odor and any health complaints is essential for improving land application of biosolids and its public acceptance.... [O]ne resolution is to initiate a regulatory requirement for a comprehensive cradle-to-grave tracking system...."

To address odor problems, in Maryland for example, a database and an electronic form for recording and investigating biosolids odor complaints was developed [63]. In particular, at DCWASA, they regularly collect odor complaint data compiled by Maryland Environmental Service (MES). Moreover, DCWASA along with its research colleagues has studied the factors of odor generation [40, 41], developed both field and on-plant odor prediction statistical models [31, 32, 38], and constructed an odor dispersion model for measuring the effect of biosolids odor in reuse fields [37].

These studies were carried out to understand the factors contributing to odor generation and finally be able to predict and/or minimize biosolids odors. In addition, to be more proactive, it is better to have these biosolids on farms with less potential to generate odor complaints (e.g., remote farms further from residential areas). However, trying to minimize the odor footprint and also apply biosolids to remote sites could generate high processing and distribution costs. Therefore, one needs a tool that finds a tradeoff between biosolids odor and their associated processing and distribution costs.

Like many other real-world applications, when there are two or more objectives being considered, it is often that these objectives compete with each other. In other words, one can only do better in one objective at the expense of at least one of the other objectives. Fortunately, there is a tool that can help decision-makers arrive at suitable alternatives among several management considerations, for example, potential conflicts between cost and liability, cost and risk, time and cost, economic development and environmental impact, to name a few. These conflicts as well as a problem of finding a tradeoff between odors and costs discussed above are regarded as common problems in real-world applications and can be handled with *multiobjective programming*. In fact, there are far more studies in multiobjective optimization than we are able to include in this literature review. Therefore, we narrow our interest to only literature concerning environmental studies which a major focus on wastewater treatment problems.

Early in 1983, Kansakar and Polprasert [39] applied multiobjective goal programming to minimize linear cost functions including costs, water quality impacts, and land use impact for the wastewater and sludge management systems. Chang et al. [12] established, in 1993, a location/allocation model for solid waste management planning in the USA and Taiwan that combined the efforts of the environmental objectives, such as air pollution, leachate, noise, and traffic congestion. During the same year, Ciric and Huchette [16] has also analyzed the sensitivity of maximum net profits of a chemical process to changes in the waste treatment cost using the multiobjective programming approach. Later in 1994, Minor and Jacobs [44] developed a multiobjective optimization model to find optimal land allocation for solid and hazardous waste landfill siting. While, the competing objectives being considered are to minimize land purchase cost, compactness, and contiguity of the selected subregion. In 1997, two other competing objectives: the conservation of reservoir watershed and the benefits from various uses of land within the watershed were analyzed in a land development optimization problem by Chang et al. [14] using fuzzy multiobjective programming. In 1997, Chang and his colleagues also studied three other multiobjective problems involving waste management systems: Chang and Wang [13] applied fuzzy goal programming to quantify the imprecise objectives of the decision maker in balancing between economic and environmental goals; a multiobjective, mixed-integer programming model was developed to find optimal vehicle routing and scheduling for solid waste management systems synthesized within a GIS environment. The systems allow a decision maker to choose optimal waste collection strategies among many other alternatives [11]; the optimal wastewater treatment strategies for water pollution control in a river basin have been investigated using interactive, fuzzy interval, multiobjective, mixed-integer programming [10].

Later in 1998, Crohn and Thomas [20] applied mixed-integer programming to minimize costs for land application of biosolids, where costs being analyzed include biosolids storage, digestion, composting, transportation, and land application. In 2001, hauling costs and environmental impact were studied as two competing objectives in the transportation planning for some industrial wastes [51]. Nema and Gupta [47] presented, in 2003, a multiobjective integer goal model to minimize total risk (including risk associated with transportation, treatment, and disposal) and total cost (including treatment, disposal, and transportation costs) of planning and design of hazardous waste management systems. Bhattacharya et al. [9] used neural networks and reinforcement learning to control and replicate optimal control strategies produced from a dynamic real-time control (RTC) of regional water systems. While, the RTC was embedded with a multiobjective nonlinear and/or dynamic programming based on simulation models to balance different interests, ranging from flood control to recreation. In 2004, a multiobjective genetic algorithm approach was used to model the optimization of the technical specifications of a nuclear safety system [42]. More recently, in 2006, Gabriel et al. [30] applied a multiobjective program to find optimal wastewater management strategies including processing and transportation costs of biosolids as well as odor reduction.

We can see that there has been a considerable amount of research involving the optimization of environmental or waste management problems. However, according to our knowledge and literature we have reviewed, there is no research (except [30]) that takes into account the minimization of wastewater treatment processing and biosolids distributing costs while simultaneously minimizing biosolids odor. The detailed models presented in this dissertation also distinguish themselves from the research studies reviewed above. In addition, the decomposition techniques employed in this dissertation are another improvement and allow the models to cover a longer time horizon planning period compared to [30].

1.2 Objectives

The objective of this dissertation is to develop multiobjective optimization models to minimize biosolids odor as well as the processing and distributing costs. These models can assist DCWASA as well as typical wastewater treatment plants in finding optimal wastewater treatment management policies (e.g., treatment processes and biosolids distribution patterns). In particular, this dissertation provides:

1. Tradeoff points between costs (including processing and distributing costs) and biosolids' odors

2. Optimal wastewater treatment processing strategies (e.g., lime additions, number of centrifuges in service);

3. Optimal biosolids distributions strategies (e.g., to what reuse fields biosolids should be applied and for how many tons)

4. Optimal percentage flows from the blend tank to DCWASA and the on-site contractor providing additional dewatering equipments

5. Fast algorithms to solve the specific wastewater multiobjective optimization problem for wastewater treatment plants

The rest of this dissertation is organized as follows. Chapter 2 introduces the multiobjective optimization programs and decomposition techniques. In particular two methods used to find the trade-off points, so-called *Pareto optimal*, are presented: weighting method [17, 52] and constraint method [17, 52]. Later, we discuss the Dantzig-Wolfe decomposition technique [3, 19, 23, 21, 33, 48, 70] and present the related solution algorithm. Finally, we briefly discuss the Benders decomposition technique [19, 34, 53]. The first technique, Dantzig-Wolfe decomposition, was successfully applied to our optimization problem and significantly improved our run times. The computational results are shown in Chapter 5. The latter technique, Benders decomposition, was also tried. However, according to our experiments, it was not as helpful from a computational point of view.

Chapter 3 discusses the wastewater treatment processes at the Blue Plains advance wastewater treatment plant as well as the characteristics of some wastewater treatment processing variables relevant to our optimization models. Later, we present odor prediction equations developed by Gabriel et al. [32] as well as data and their sources. Finally, we develop an odor threshold index for each reuse field. These indices are normalized among the reuse field candidates and indicate how sensitive to odor each reuse field is, compared to others in the set of candidates.

Chapter 4 discusses the multiobjective optimization models when the percentage of flow from the blend tank to DCWASA[2] was exogenously determined. In other words, the percentage of flow handled by DCWASA (as opposed to an on-site contractor) was given as a parameter. The resulting model is a multiobjective linear integer program, which can be solved by an existing optimization solver. Later, the weighting method and constraint method were applied to the Base Case. The Pareto optimal points that were generated were analyzed and we found for various situations (e.g., how many index points of odor level need to be decreased), different

[2] See Section 3.1 for wastewater treatment processes.

marginal activities were required. For example, at one point, the total number of centrifuges in service was a key activity for reducing biosolids odor levels. Next, we performed a sensitivity analysis on the percentage of flow from the blend tank, odor threshold input, and DCWASA's operating costs. The sensitivity analysis indicates that when 20% of flows were assigned to DCWASA, the best Pareto optimal curve was obtained. However, there was no obvious effect when different sets of odor threshold input were used. Lastly, it is shown that changes in DCWASA's operating costs may affect the non-dominated sets of solutions.

Chapter 5 discusses a successor to the optimization model from Chapter 4. Specifically, the percentage of flow from the blend tank is endogenously determined and defined as a decision variable. The resulting model is a multiobjective, nonlinear, integer program. In order to be able to solve the resulting problem with existing solvers, we transformed the nonlinear, integer program into a linear integer one by employing some approximation techniques involving the *Schur's decomposition* and *Special Order Set* of type 2 variables (SOS2) [5, 6, 27, 29, 36]. Finally, we present several solution algorithms and discuss computational results.

Lastly, Chapter 6 summarizes major findings and limitations of optimization models as well as provides a list of potential future work.

Chapter 2

An introduction to Multiobjective Optimization, Dantzig-Wolfe Decomposition, and the Benders Decomposition

In this chapter we introduce a multiobjective optimization program and present two methods used to find tradeoff points between costs and odors. These methods are weighting method and constraint method. Later, we discuss two decomposition techniques: Dantzig-Wolfe decomposition and Benders decomposition.

2.1 Multiobjective Optimization

A single-objective constrained optimization problem can be represented as

$$\begin{aligned} &\min f(x) \\ &\text{subject to} \\ &\qquad x \in S \subseteq R^n \end{aligned} \tag{2.1}$$

where $f(x)$ is the objective function and S is the feasible region. The purpose of the optimization problem is to find the point(s) in S, which yields the lowest objective function value $f(x)$. If $f(x)$ is linear and S is defined by all linear functions, we call (2.1) a single-objective linear optimization problem. If (2.1) is linear and one or more of the constraints defining S requires that x be an integer, we then call (2.1) a single-objective linear integer optimization problem. If $f(x)$ is nonlinear or one or more of the constraints defining S is also nonlinear, we call (2.1) a single-objective nonlinear optimization problem. Some examples of single objectives that can be optimized are cost, risk, profit, and environmental influences.

Furthermore, methods for solving single-objective optimization problems have been studied by many authors, such as Chvatal [15], Nash and Sofer [46], and Bertsimas and Tsitsiklis [8] to name a few. However, in real world situations, decision-makers have encounter problems where more than one objective is required to be taken into account. Illustrations of some problems that may be more adequately modeled with multiobjective programming are oil refining problems and portfolio selection problems. To be more specific, for oil refining problems, we may want to simultaneously minimize cost, imported crude oil, high sulfur crude oil, and flaring of gases. Similarly, for portfolio selection problems, investors may want to minimize risk as well as maximize their return on the investment. After having introduced a few applied real world examples, we now present the general form of a multiobjective mathematical program.

$$\begin{array}{c} \min f_1(x) \\ \vdots \\ \min f_k(x) \\ \text{subject to} \\ \qquad\qquad x \in S \subseteq R^n \end{array} \tag{2.2}$$

where $f_1(x), \ldots, f_k(x)$ are different objectives to be minimized and S is the feasible region. To avoid the trivial case where all objectives are aligned, we assume that there is no solution that will simultaneously minimize each objective. In other words, we can only do better in one objective at the expense of at least one of the other objectives. Actually, this assumption is the concept for finding the solutions to the multiobjective optimization problem, which will be discussed later. Again, the linear, nonlinear, and integer nature of the problem can be applied to a multiobjective optimization problem with the same criteria as discussed earlier.

For many years, multiobjective mathematical programs have played an increasingly important role in aiding decision-makers to choose suitable alternatives among several options. Several authors have proposed numerous methods for solving multiobjective mathematical programs. Some of these various methods include the constraint method [52, 17], the weighting method [52, 17], genetic algorithms [26], and the multiobjective simplex method [52], to name a few.

The set of the solutions to a multiobjective optimization problem is often referred to by welfare economists as a *Pareto optimal set*, by other disciplines as a *noninferior set* or *efficient set* [17, 52]. Nevertheless, all these terms have the same meaning. For consistency, the term Pareto optimal set will be used to refer to the solutions to the multiobjective optimization problem throughout this work. A solution (call it A) to a multiobjective optimization problem is Pareto optimal if no other feasible solution is at least as good as A with respect to every objective and strictly better than A with respect to at least one objective [69]. Mathematically, according to problem (2.2):

Definition 2.1.1 *[17]*

Given functions $f_i : R^n \to R, i = 1, 2, \dots, k$ and the set $S \subseteq R^n$, a decision vector $x^* \in S$ is Pareto optimal *if there does not exist another vector $x \in S$ such that*

$f_i(x) \leq f_i(x^*)$ for all $i = 1, 2, \ldots, k$ and $f_j(x) < f_j(x^*)$ for at least one index j.

Figure 2.1 displays an example of a Pareto optimal set, where $f_1(x)$ and $f_2(x)$ are two objectives being minimized in an multiobjective integer optimization problem. From Figure 2.1, we can see that Point B is inferior to Point C because Point C has a lower value of the objective function #2, although it has the same value of the objective function #1. However, Point D is obviously dominated by Points B and C because Point D attains higher values of both objective functions. In all, Points A, C, and E are in the Pareto optimal set.

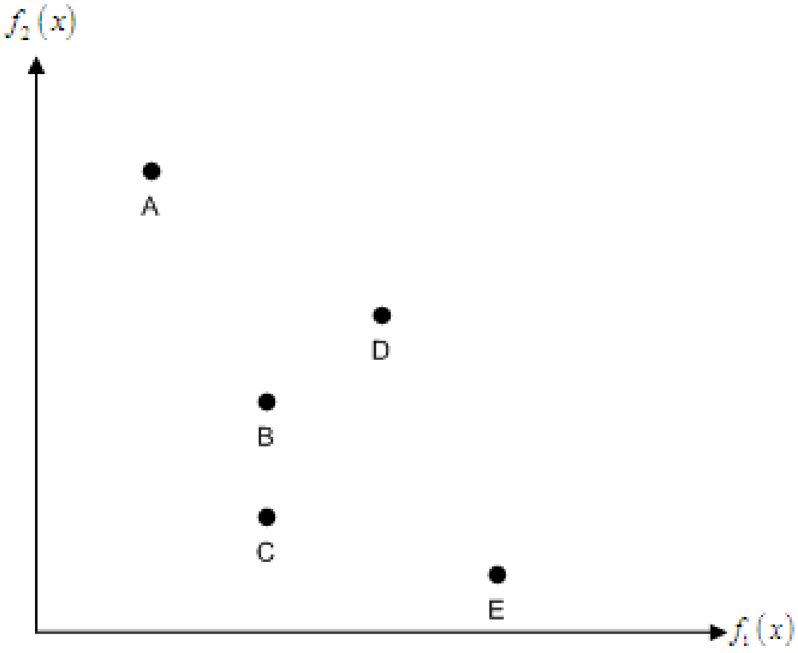


Figure 2.1: Example of a Pareto optimal set

Next, we examine two methods used to find the Pareto optimal set: constraint method and weighting method.

2.1.1 Constraint Method

Given a multiobjective optimization with k objectives as in (2.2), the related constrained problem is

$$\begin{aligned}
&\min f_h(x)\\
&\text{subject to}\\
&\qquad x \in S\\
&\qquad f_p(x) \leq U_p, p = 1, 2, \ldots, h-1, h+1, \ldots, k
\end{aligned} \tag{2.3}$$

where the objective member h is arbitrarily chosen for minimization. Problem (2.3) is a single-objective optimization problem, which can be solved by conventional methods, such as using simplex method for linear programs [17]. The optimal solution to (2.3) is a Pareto optimal solution to the original problem (2.2) if certain conditions are satisfied as discussed below.

The optimal solution to (2.3) is a Pareto optimal solution to the original problem (2.2) if all the constraints on the objectives are binding at the optimal solution to (2.3) [17]. In other words, evaluating at the optimal solution x^*, $f_p(x^*) = U_p$ for all $p = 1, 2, \ldots, h, h-1, h+1, \ldots, k$. However, if this is not the case (i.e., not all constraints on the objectives are binding), the optimal solution to (2.3) is a Pareto optimal solution to the original problem (2.2) if there is no alternative optimum to (2.3). In the latter case, if there are alternative optima to (2.3), then some of these alternative optimal solutions may not be Pareto optimal to the original problem (2.2). If this is the case, we may proceed to search among the alternative optimal for the Pareto optimal solution(s) by using the following procedures. First, without loss of generality, suppose that the constraints on $f_p(x) \leq U_p$ for $p = 1, 2, \ldots, h-1$ are binding and those for $p = h+1, h+2, \ldots, k$ are satisfied with strict inequalities. Then we can set up a sub-problem to search for non-inferior solutions among those alternative optima as follows:

$$\begin{aligned}
&\min f_h(x) \\
&\text{subject to} \\
&\qquad x \in S \\
&\qquad f_p(x) = U_p, p = 1, 2, \ldots, h-1 \\
&\qquad f_q(x) \leq U_q, q = h+1, h+2, \ldots, k
\end{aligned} \tag{2.4}$$

Solving (2.4) is equivalent to solving the multiobjective optimization program by the constraint method as applied to a subset of the original feasible region. Again, if the alternative optima are observed, we proceed to search for the Pareto optimal solutions among the alternative optima as described before [17].

Next, we investigate another method used to find the Pareto optimal solutions of the multiobjective optimization problem, which is the weighting method.

2.1.2 Weighting Method

The weighting method is another effective method to find Pareto optimal solutions for the multiobjective optimization problem (2.2). The technique is to weight the objectives to obtain Pareto optimal solutions. Given a multiobjective optimization with k objectives as in (2.2), the weighted problem is

$$\begin{aligned} &\min \sum_{i=1}^{k} w_i f_i(x) \\ &\text{subject to} \\ &\qquad x \in S \\ &\qquad \sum_{i=1}^{k} w_i = 1 \\ &\qquad w_i \geq 0; i = 1, 2, \ldots, k, \end{aligned} \tag{2.5}$$

It is noted here that any sets of nonnegative weights w_i may be used in (2.5). However, without loss of generality, we can normalize all weights such that $\sum_{i=1}^{k} w_i = 1$. The optimization problem (2.5) is a single-objective optimization problem that can be solved by existing methods. To illustrate the role of weights, Figure 2.2 displays the feasible region F_0, in objective space, of the two-objective minimization problem (2.6) (see Cohon [17] for maximization problem).

$$\begin{aligned} &\min \sum_{i=1}^{2} w_i f_i(x) \\ &\text{subject to} \\ &\qquad x \in S \\ &\qquad \sum_{i=1}^{2} w_i = 1 \\ &\qquad w_i \geq 0; i \in \{1, 2\} \end{aligned} \tag{2.6}$$

Referring to Figure 2.2, N_0 indicated by the bold line represents the Pareto optimal set. The contours of the objective function or the linear indifference curves, with a slope of $-\frac{w_1}{w_2}$, are defined by the equation

$$w_1 f_1(x) + w_2 f_2(x) = A \tag{2.7}$$

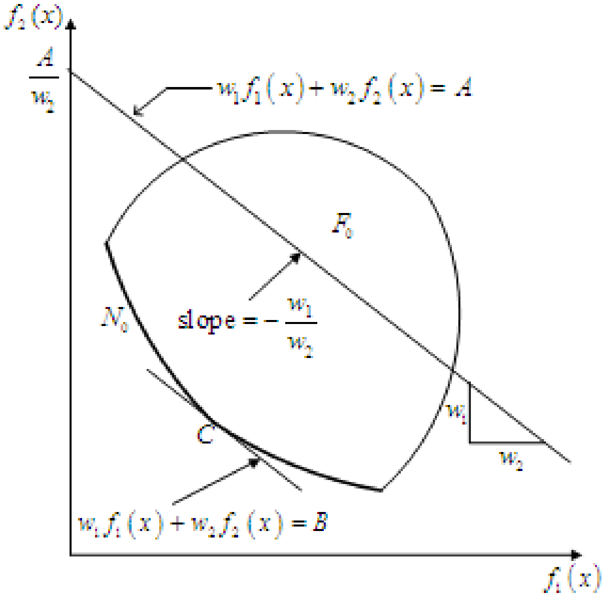

$f_2(x)$
$\frac{A}{w_2}$
$w_1 f_1(x) + w_2 f_2(x) = A$
F_0
N_0
slope $= -\frac{w_1}{w_2}$
w_1
w_2
C
$w_1 f_1(x) + w_2 f_2(x) = B$
$f_1(x)$

Figure 2.2: Feasible region of the two-objective minimization problem in objective space: a weighted objective function appears as a linear indifference curve

where A is an arbitrarily chosen constant. The solution of the problem is obtained by pushing the contour as far to the southwest as possible until it just touches the boundary of F_0. This happens at point C, where the constant in (2.7) is B. Other Pareto optimal solutions in the Pareto optimal set N_0 can be obtained by varying the value of $\frac{w_1}{w_2}$ as indicated in Figure 2.3.

The optimal solutions to problem (2.5) are Pareto optimal to the original multiobjective optimization problem (2.2) as long as all of the weights are positive [17, 43]. However, if not all the weights are positive and if there are alternative optima to the weighted problem (2.5), then some of these optimal solutions may not be Pareto optimal to the original multiobjective problem (2.2). If this is the case, we may use the following procedures developed by Cohon [17] to search among these alternative optima for the Pareto optimal solutions.

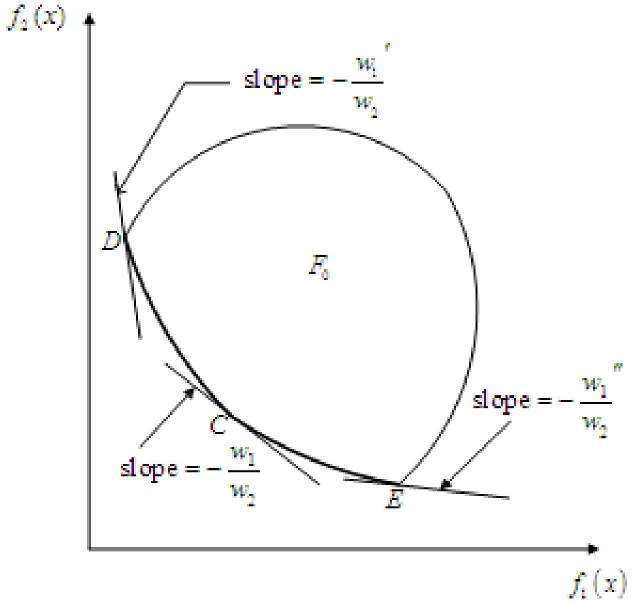

$f_2(x)$
slope = $-\frac{w_1'}{w_2}$
D
F_0
slope = $-\frac{w_1''}{w_2}$
C
slope = $-\frac{w_1}{w_2}$
E
$f_1(x)$

Figure 2.3: Pareto optimal solutions as $\frac{w_1}{w_2}$ varies

Suppose that the alternative optimal solutions to (2.5) gave $f_i(x^*)$ for $i = 1, 2, \ldots, h$ and that $w_i > 0$ for $i = h+1, h+2, \ldots, k$. Then we can solve a new problem to find the Pareto optimal solutions.

$$\begin{aligned}
&\min \sum_{i=h+1}^{k} w_i f_i(x) \\
&\text{subject to} \\
&\qquad\qquad x \in S \\
&\qquad\qquad f_i(x) = f_i(x^*), i = 1, 2, \ldots, h
\end{aligned} \tag{2.8}$$

In solving problem (2.8), we set $w_i > 0$, for $i = h+1, h+2, \ldots, k$. The values of w_i are chosen in various combinations to approximately ascertain Pareto optimal solutions among those alternative optima. For some particular multiobjective optimization problems, it is a computationally challenging task to search for all Pareto optimal solutions. Theoretically, if one wants to guarantee that all Pareto optimal solutions can be obtained by the weighting method, some additional conditions on the objective functions and the feasible region must be satisfied. These conditions require that all objective functions are convex and the feasible region is a convex set [43]. The definitions of a convex set and a convex function are given below.

Definition 2.1.2 *(page 60 in [69])*

A set of points S convex *if the line segment joining any pair of points in S is wholly contained in S. That is, $\forall x, y \in S, C \in [0,1]$, the point $z = Cx + (1 - C)y \in S$, where Z is a convex combination of x and y*

Let $f\left(x_1, x_2, \ldots, x_n\right)$ be a function that is defined for all points $\left(x_1, x_2, \ldots, x_n\right)$ in a convex set S.

Definition 2.1.3 *(page 652 in [69])*

A function $f : R^n \rightarrow R$ is convex *on a convex set $S \subseteq R^n$ if for any $x' \in S$ and $x'' \in S, f\left(cx' + (1-c)\,x''\right) \leq cf\left(x'\right) + (1-c)\,f\left(x''\right)$ holds for $0 \leq c \leq 1$*

The phenomenon where some of the Pareto optimal solutions cannot be obtained because of the optimization problem being nonconvex is referred to as a *duality gap* [17]. To illustrate this phenomenon, Figure 2.4 depicts the nonconvex Pareto optimal set of an integer program (minimization) in objective space. It is indicated that the Pareto optimal point C could never be obtained by the weighting method. Suppose the slope of the objective function contour $\frac{w_1}{w_2} \leq -\beta$, point A will be the optimal solution to the weighted problem and if $\frac{w_1}{w_2} \geq -\beta$, point B will be the optimal solution. Therefore, there is no value for $\frac{w_1}{w_2}$ such that C is the optimal solution of the weighted problem.

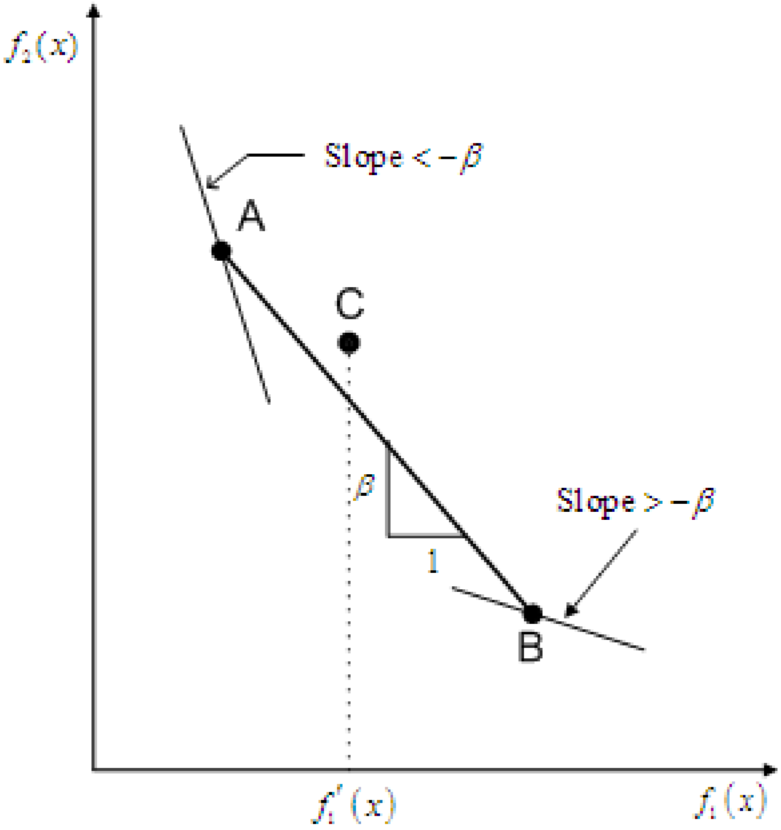


Figure 2.4: Duality gap in weighting method

The convexity requirements seem to be a disadvantage for the weighting method. However, for practical purposes, we may not need all Pareto optimal solutions. In which case, the analyst may just determine only the range of the Pareto optimal solutions that interest the decision-makers. Moreover, the fact that the weighting method only requires positive weights to guarantee Pareto optimal solutions perhaps makes it more favorable over the constraint method unless all Pareto optimal solutions are required. Another point to note is that for multiobjective integer programs as used in this dissertation, when there is a nonzero optimality gap, at best only local Pareto optimal solution can be found.

In the next section, Dantzig-Wolfe Decomposition, a technique used to decompose a big problem into several small problems and one master problem coordinating those small problems, is discussed. In particular, the decomposition technique is employed in Chapter 5 to improve the problem's lower bounds on the objective function (see Section 5.3).

2.2 Dantzig-Wolfe Decomposition

In this section, the optimization problem of the following form is discussed.

$$\min \sum_{d=1}^{|D|} c^d x^d$$

subject to

$$\begin{array}{ccccccc} A^1x^1 & +A^2x^2 & +\dots & +A^{|D|}x^{|D|} & \leq & b \\ E^1x^1 & & & & \leq & f_1 \\ & \dots & & & \leq & . \\ & & \dots & & \leq & . \\ & & & E^{|D|}x^{|D|} & \leq & f_{|D|} \\ x^1 \in R_+^{n_1}, & \dots & \dots & ,x^{|D|} \in R_+^{n_{|D|}} & & \end{array} \tag{2.9}$$

It is noted here that $D = \{1, 2, \dots, |D|\}$, where $|D|$ represents the cardinality of the set D. Also note that $x^1, x^2, \dots, x^{|D|}$ represent sub-vectors of the overall vector X.

It can be seen that when the *joint constraints*[1] $\sum_{d=1}^{|D|} A^d x^d \leq b$ are removed, the sets $X^d = \{x^d \in R_+^{n_d} : E^d x^d \leq f_d\}$ are independent for all $d \in D$. Before delving more deeply into its solution process, we provide an example of application that may be represented by the optimization problem structure presented above. Dantzig [21] one of the inventors of this decomposition technique[2] gave a very clear example of a typical situation where the decomposition principle can be applied. Consider a plant with two almost independent shops. Each shop has its own constraints unaffected by the activities of the other shop. However, the two shops are tied together with a few constraints and a common objective. Now considering the optimiza-

[1] Constraints that link the different sets of variables together [70].

[2] Another inventor is P. Wolfe

tion problem we presented above, each shop's constraints would correspond to the constraint $E^d x^d \leq f_d$. The common objective and constraints that tie the shops together would correspond to $\min \sum_{d=1}^{|D|} c^d x^d$ and $\sum_{d=1}^{|D|} A^d x^d \leq b$, respectively. Another example corresponds directly to the problem being studied in this thesis. Considering the operation of a wastewater treatment plant over a certain time period, on each day the plant must process wastewater and deliver its end product, so-called biosolids, to the field sites. Although the daily processes are almost independent from each other, at the end of each month, the plant must achieve their objectives of minimizing processing and distributing cost and also overall biosolids odor levels generated throughout the month. In addition, the biosolids delivered to each field must not exceed its capacity limit. According to this example, the common objectives; minimizing cost and odor and the field capacity limits constraints tie the daily plant processes together.

The decomposition technique was first developed by Dantzig and Wolfe [23] to break apart linear programs with decomposable structure similar to the one shown in Problem (2.9). Although originally invented for linear programs, with slight modification, the decomposition technique has proved to be useful for integer programs as well. In particular, it has been proven that the optimal objective function value provided by the Dantzig-Wolfe decomposition, although not always providing an integer solution, can still be used as the lower bounds (minimization problem) or the upper bound (maximization problem) for the original integer program [70, 66]. In addition, the solution obtained can then be utilized later in the *branch and price* algorithm[3]. In this dissertation, we do not further utilize the solution from the Dantzig-Wolfe decomposition in a branch and price algorithm. Instead, the optimal objective function values obtained are used as lower bounds for our original integer program and help us numerically prove that the integer solutions obtained from the original integer program are close to optimal (see Section 5.4). The proof on why the solution from the Dantzig-Wolfe decomposition technique can provide the lower or upper bounds for the original integer program can be found in [70, 66].

Next, we describe a solution idea of the Dantzig-Wolfe decomposition technique as it is applied to linear programs. However, with a very slight modification, this solution algorithm can be applied to integer programs without much modification. It is noted here that the technique is explained in general terms first and then more detail is added as we proceed.

The Dantzig-Wolfe decomposition technique decomposes the original linear program into two parts; a master problem and pricing subproblems. The joint constraints go in the master problem and each set of the independent constraints go in the pricing subproblems. The algorithm starts with the master problem whose solution is used to construct the objective function for the pricing subproblems. Then the pricing subproblems are solved and their solutions, called *proposals*, are introduced to the master problem. Subsequently, new variables, so-called weights, associated with proposals are generated and added to the master problem. The master problem is then solved again with new added variables, the prices are adjusted

[3] Branch and price also known as IP column generation [2, 67] is a solution technique of embedding the column generation in a branch-and-bound framework.

and again sent to the pricing problems. The process continues in this manner until the optimality test is passed after a finite number of iterations [21]. To put it differently, the master problem acts like the plant manager that coordinates and adjusts the prices for each shop or pricing subproblem. While each shop proposes the best of what they can do. Figure 2.5 is adapted from [21] and illustrates the overview of this decomposition technique. Next, we discuss the mathematical formulations for the master problem and the pricing subproblems.

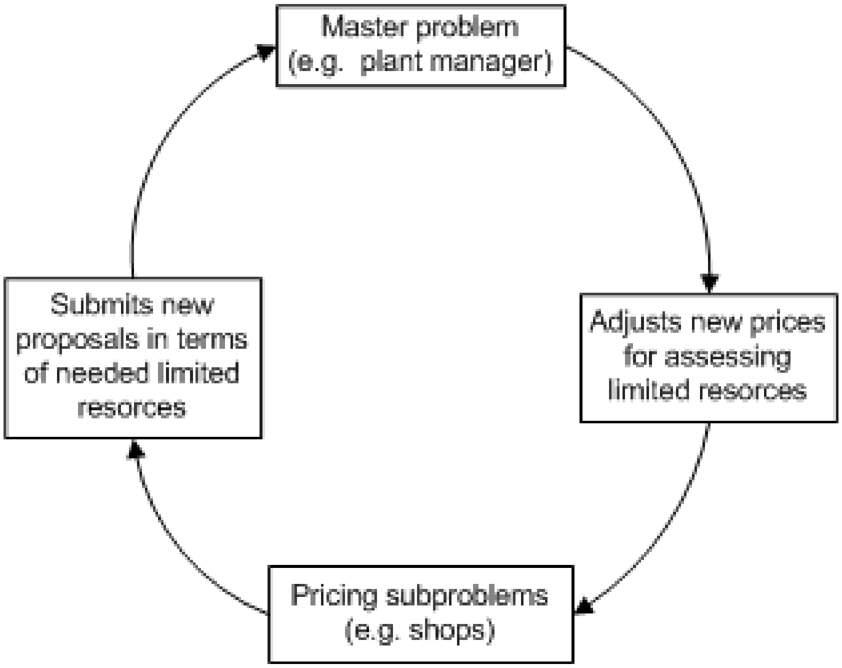


Figure 2.5: Overview of Dantzig-Wolfe decomposition technique [21]

2.2.1 Master Problem and Pricing Subproblems

First, consider the original problem (2.9), which may be rewritten as:

$$\text{(LP)} \qquad \begin{aligned} &\min \sum_{d=1}^{|D|} c^d x^d : \sum_{d=1}^{|D|} A^d x^d = b, x^d \in X^d; \forall d \in D \\ &X^d = \{x^d \in R_+^{n_d} : E^d x^d <= f_d\}; \forall d \in D \end{aligned} \tag{2.10}$$

For the sake of discussion, let's assume the feasible sets for X^d to be bounded[4] convex sets. Only slight modifications in the formulas are required for the unbounded case [21] and will not be discussed since it is not relevant to this dissertation.

Definition 2.2.1 *(page 60 in [69])*

For any convex set S, a point P in S is an extreme point *if each line segment that lies completely in S and contains the point P has P as an endpoint of the line segment. Alternatively, P is an extreme point of S, if it is not a convex combination of other points in S.*

Under this assumption, each set X^d contains a a large but finite set of *extreme points* $\{x^{d,t}\}_{t=1}^{|T_d|}$ and we may write $X^d = \{x^d \in R^{n_d} : x^d = \sum_{t=1}^{|T_d|} \lambda_{d,t} x^{d,t}, \sum_{t=1}^{|T_d|} \lambda_{d,t} = 1, \lambda_{d,t} \geq 0; \forall t \in T_d$. Now substituting for x^d leads to an equivalent *extremal problem* or the *full master problem* [21]. It is also identical to the *Linear Programming Master Problem* (LPM) discussed in the Dantzig-Wolfe reformulation of an integer program by Wolsey [70].

$$\text{(LPM)} \qquad \begin{aligned} & \min \sum_{d=1}^{|D|} \sum_{t=1}^{|T_d|} (c^d x^{d,t}) \lambda_{d,t} \\ & \sum_{d=1}^{|D|} \sum_{t=1}^{|T_d|} (A^d x^{d,t}) \lambda_{d,t} = b \qquad ; \pi_i \\ & \sum_{t=1}^{|T_d|} \lambda_{d,t} = 1 \forall d \in D \qquad ; \mu_d \\ & \lambda_{d,t} \geq 0; \;\; \forall t \in T_d, d \in D \end{aligned} \qquad (2.11)$$

[4] A bounded set is a set of finite site. Also, see Definition 2.1.2 on page 23 for a definition of a convex set.

It is noted here that there is a column $\begin{pmatrix} c^d x \\ A^d x \\ e_d \end{pmatrix}$ associated with each $x \in X^d$. In addition, $\{\pi_i\}_{i=1}^{m}$ and $\{\mu_d\}_{d=1}^{|D|}$ are the dual variables associated with the joint constraints and the second set of constraints (also known as *convexity* constraints [70]), respectively.

If the linear programming master problem (2.11) is solved, so is the original problem (2.10). Unfortunately, for most problems, the extreme points $\{x^{d,t}\}_{t=1}^{|T_d|}$ of each set X^d are far too numerous to be expressed explicitly. Therefore, problem (2.11) is solved iteratively each with a subset of the extreme points $\{x^{d,t}\}_{t=1}^{|T_d|}$ or equivalently a subset of columns $\begin{pmatrix} c^d x \\ A^d x \\ e_d \end{pmatrix}$. The full master problem (a.k.a. extremal problem, LPM) with only a subset of columns is called *Restricted Master Program* [21]. Alternatively, it can also be called the *Restricted Linear Programming Master Problem* (RLPM) [70].

$$
\text{(RLPM)} \qquad
\begin{aligned}
& \min \sum_{d=1}^{|D|} \sum_{t=1}^{|\tilde{T}_d|} (c^d x^{d,t}) \lambda_{d,t} \\
& \sum_{d=1}^{|D|} \sum_{t=1}^{|\tilde{T}_d|} (A^d x^{d,t}) \lambda_{d,t} = b && ;\pi_i \\
& \sum_{t=1}^{|\tilde{T}_d|} \lambda_{d,t} = 1;\ \forall d \in D && ;\mu_d \\
& \lambda_{d,t} \geq 0;\ \forall t \in \tilde{T}_d, d \in D \\
& 1 \leq |\tilde{T}_d| \ll |T_d|
\end{aligned}
\qquad (2.12)
$$

Subsequently, an efficient method to choose only good extreme points or columns as to improve the LPM objective function value at each iteration is needed. This is the point where the pricing subproblems we mentioned above come in.

The pricing subproblem finds a column with the smallest reduced cost (or largest reduced cost for maximizing problem). In other words, it finds the columns or variables, when added into the master problem for which the master problem objective function will reduce the most (and vice versa for a maximizing problem). According to [3, 68, 22], the reduced cost associated with each column $x \in X^d$ can be determined as $c^d x - \pi A^d x - \mu_d$. Since the number of columns are numerous and we are only interested in the column yielding the smallest reduced cost, the following pricing subproblem can be set up for each X^d.

$$\text{(Pricing Subproblem)} \qquad \min \left(c^d - \pi A^d\right) x - \mu_d : x \in X^d \tag{2.13}$$

Therefore, there are $|D|$ pricing subproblems altogether to be solved. As we mentioned earlier that the master problem and pricing subproblems will be solved iteratively, next we discuss the stopping criteria. The solution algorithm terminates when there is no column (variable), when added into the master problem that can reduce the master problem's objective function value. In other words, the maximum reduced costs from all pricing subproblems are greater than or equal to zero. Having discussed the mathematical formulations for both master problem and pricing subproblems as well as the stopping criteria, next we introduce the algorithm implementing the Dantzig-Wolfe decomposition technique presented above.

2.2.2 Dantzig-Wolfe Decomposition: The Overall Algorithm

This algorithm is adapted from the Dantzig-Wolfe decomposition method for linear programming presented in [19], as well as [70] when applied to an integer program, and the MOSEL code developed for combining sequential and parallel solving [18]. Our algorithm is presented in four phases; Phase 1: Initialization, Phase 2: Master problem feasibility check, Phase 3: Optimization of the RLPM, and Phase 4: Calculation of the solution to the original problem. Phase 1 is simple and only requires optimizing each pricing subproblem only once to check for its feasibility and obtain initial feasible solutions. Phases 2 and 3 involve the optimization of the master problem and the pricing subproblems repeatedly until a stopping criteria is met. Finally, Phase 4 determines the solutions to the original problem. What follow are detailed explanation for each phase as well as the algorithm flowcharts.

2.2.2.1 Phase 1: Initialization

Phase 1 serves two purposes: check for feasibility of each pricing subproblem d and obtain initial feasible solutions. During this phase, we do not have information on dual variables from the master problem yet. Therefore, each pricing subproblem d is solved with its original objective function (i.e., $c^d x^d$). The infeasibility of at least one pricing subproblem indicates the infeasibility of the original problem and the algorithm terminates. On the other hand, if all pricing subproblems are feasible, at lease one feasible solution from each pricing subproblem d is needed. These $|D|$ solutions are input to the master problem at the start of Phase 2. Since any feasible solution will do and in order to save computing time, solving each pricing subproblem to optimality at this phase is unnecessary unless the optimal solution is available easily. With that in mind, we seek the best solutions obtained within a given preset computing time limit depending on the application. If no integer solution is found within the preset time limit, the optimizer continues until the first integer solution is found. Figure 2.6 displays a flowchart for Phase 1 and a detailed explanation of each step follows.

1. Start Phase 1. Initialize $d = 1$.

2. Set the proposal counter for pricing subproblem d: $|\tilde{T_d}| = 1$. Solve pricing subproblem d, with its original objective function and constraints: min $c^d x^d$: $E^d x^d \leq f^d$.

3. If pricing subproblem d is feasible, continue to next step. Otherwise, algorithm terminates. Report original problem as infeasible.

4. Record solution for pricing subproblem d, denoted by $\tilde{x^d}$. $x^{d,|\tilde{T_d}|} = \tilde{x^d}$.

5. Update $d \leftarrow d + 1$.

6. Check if all pricing subproblems have been solved. In other words, Check if $d = |D|$. If yes, Phase 1 finishes. Continue to Phase 2. If no, go to Step 2.

2.2.2.2 Phase 2: Master Problem Feasibility

The feasible solutions from each pricing subproblem d obtained during Phase 1 are used to generate and initialize the solutions for the master problem. However,

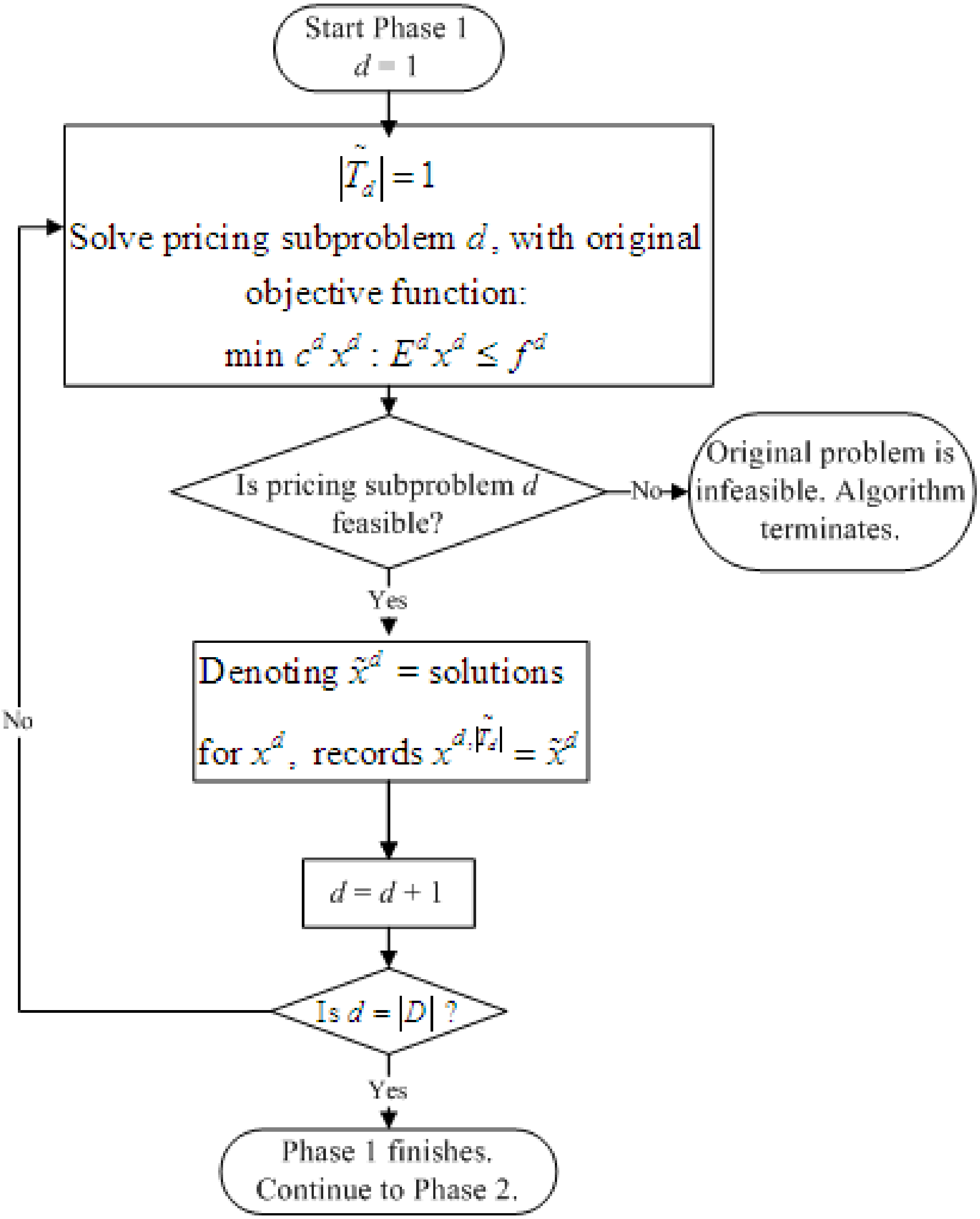


Figure 2.6: Flowchart for Phase 1

the solutions from these pricing subproblems may not always be feasible in the master problem. In other words, these solutions can not satisfy the joint constraints. Therefore, practically during this phase, slack variables are added to constraints in the master problem in such a way that this modified master problem is always feasible. Subsequently, the master problem objective function is also modified to minimize the sum of all slack variables. As for each pricing subproblem, its objective function is modified such that the original objective function coefficients are removed. In particular, the pricing subproblem objective function is reduced to $\min\left(-\pi A^d\right)x - \mu_d$. This modified objective function helps in identifying pricing subproblem solutions so as to reduce the infeasibility of the master problem and/or prove that feasibility cannot be achieved [54]. Figure 2.7 illustrates a flowchart for Phase 2. The goal here is to iterate through the master problem and pricing subproblems until the modified master problem optimal objective function value goes to zero, meaning that the slack variables are no longer needed to maintain problem feasibility. In other words, the proposed solutions from pricing subproblems can satisfy the constraints (joint constraints) in the master problem. In contrast, if all proposals from pricing subproblems are exhausted and the master problem optimal objective function is not zero, then the original problem is infeasible and the algorithm terminates. Next, we explain Phase 2 algorithm step by step.

1. Start Phase 2. Add variable $\lambda_{d,|\tilde{T}_d|}$. Note that the algorithm continues to Phase 2 if and only if every pricing subproblem is feasible in Phase 1. Therefore, $|\tilde{T}_d| = 1 \forall d \in D$.

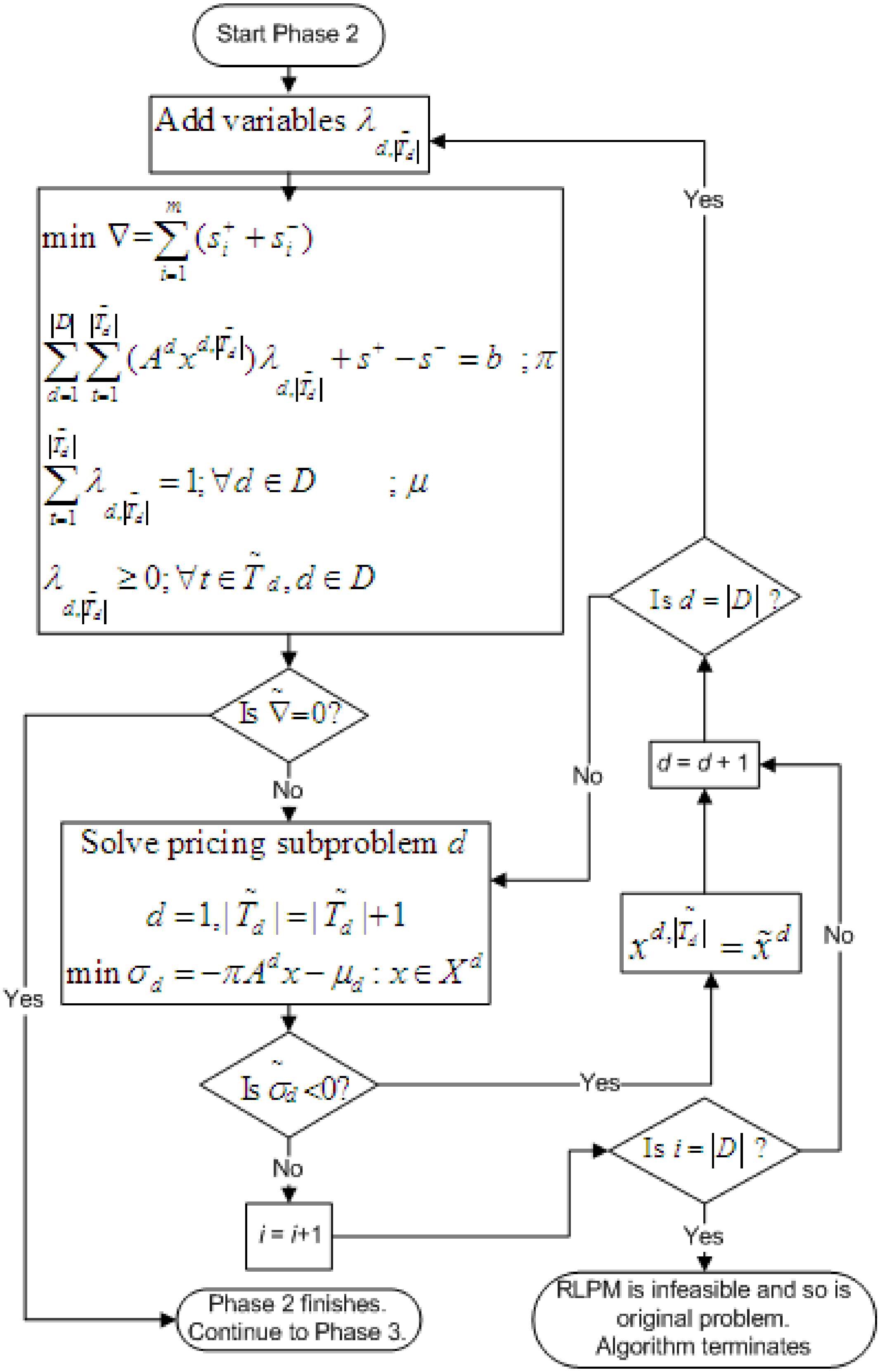

Start Phase 2
Add variables $\lambda_{d,|\tilde{T}_d|}$
$\min \nabla = \sum_{i=1}^{m} (s_i^+ + s_i^-)$
$\sum_{d=1}^{|D|} \sum_{t=1}^{|\tilde{T}_d|} (A^d x^{d,|\tilde{T}_d|}) \lambda_{d,|\tilde{T}_d|} + s^+ - s^- = b \; ; \pi$
$\sum_{t=1}^{|\tilde{T}_d|} \lambda_{d,|\tilde{T}_d|} = 1; \forall d \in D \quad ; \mu$
$\lambda_{d,|\tilde{T}_d|} \geq 0; \forall t \in \tilde{T}_d, d \in D$
Is $\tilde{\nabla}=0$?
No
Yes
Solve pricing subproblem d
$d = 1, |\tilde{T}_d| = |\tilde{T}_d| + 1$
$\min \sigma_d = -\pi A^d x - \mu_d : x \in X^d$
Is $\tilde{\sigma}_d < 0$?
Yes
No
$i = i+1$
Phase 2 finishes. Continue to Phase 3.
$x^{d,|\tilde{T}_d|} = \tilde{x}^d$
$d = d + 1$
Is $d = |D|$?
Yes
No
Is $i = |D|$?
No
Yes
RLPM is infeasible and so is original problem. Algorithm terminates

Figure 2.7: Flowchart for Phase 2

2. Solve the RLPM with the modified objective function. Assuming there are m joint constraints, the slack variables: $\{s_i^+, s_i^-\}_{i=1}^m$ are added to their m joint constraints to make the RLPM always feasible. Consequently, the objective function is min $\nabla = \sum_{i=1}^{m}(s_i^+ + s_i^-)$. The dual variables associated with the joint constraints and convexity constraint, denoted by π and μ respectively, are recorded.

3. If the optimal objective function value of the RLPM (denoted ∇) $= 0$, solutions from the pricing subproblems satisfy master problem joint constraints. Go to Step 10. Otherwise, continue to next step.

4. Initialize $d = 1$. Update the proposal counter for pricing subproblem d: $|\tilde{T_d}| = |\tilde{T_d}| + 1$. Solve pricing subproblem d, with modified objective function. In particular, the objective function is reduced to min $\sigma_d = -\pi A^d x - \mu_d$.

5. Check if pricing subproblem objective function value, denoted $\tilde{\sigma_d} < 0$. If yes, go to Step 7. If no, continue to next step.

6. Count the number of pricing subproblem whose objective function value ≥ 0 ($\tilde{\sigma_d} \geq 0$). Set $i \leftarrow i + 1$. If $i = |D|$, RLPM is infeasible and so is the original problem. Algorithm terminates. Otherwise, go to Step 8.

7. Record solution for pricing subproblem d: $x^{d,|\tilde{T_d}|} = \tilde{x^d}$. Go to next step.

8. Update $d \leftarrow d + 1$.

9. Check if $d = |D|$. If yes, go to Step 1. If no, go to Step 4.

10. Phase 2 finishes. Continue to Phase 3.

2.2.2.3 Phase 3: Optimization of the RLPM

Phase 3 iterates through the RLPM and pricing subproblem until no proposal from the pricing subproblem can improve the objective function value of RLPM. At that point, the objective function values for all pricing subproblems are greater than or equal to zero. In other words, all reduced costs or $\tilde{\sigma_d} \geq 0$. Figure 2.8 shows a flowchart for Phase 3. The explanation of each step follows.

1. Start Phase 3, with all $\lambda_{d,\tilde{T_d}}$ added from Phase 2.

2. Solve RLPM. Record dual variable values: π and μ.

3. Initialize $d = 1$. Update $|\tilde{T_d}| \leftarrow |\tilde{T_d}| + 1$. Retrieve π and μ values from previous step. Solve the pricing subproblem d.

4. Check pricing subproblem objective function value, $\tilde{\sigma_d}$. If $\tilde{\sigma_d} < 0$. Go to Step 7. Otherwise, go to next step.

5. Count the number of pricing subproblem whose solution will not improve the RLPM objective function value ($\tilde{\sigma_d} \geq 0$ for minimization problem, $\tilde{\sigma_d} \leq 0$ for maximization problem). Update $i \leftarrow i + 1$.

6. Check if $i = |D|$. If yes, go to 12. Otherwise, go to 8.

7. Record solution for pricing subproblem d: $x^{d,|\tilde{T_d}|} = \tilde{x^d}$. Go to next step.

8. Update $d \leftarrow d + 1$.

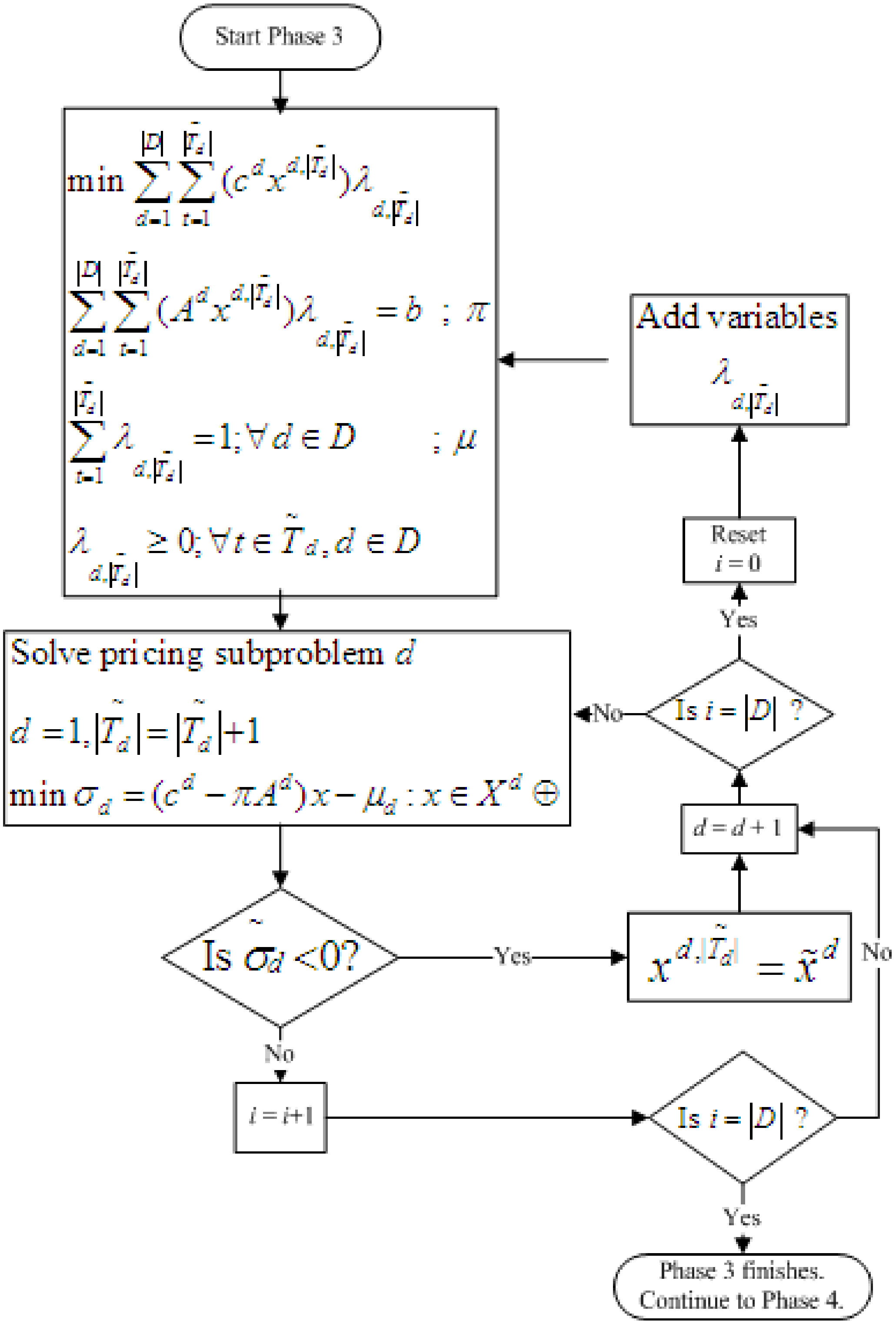

Start Phase 3
min $\sum_{d=1}^{|D|}\sum_{t=1}^{|\tilde{T}_d|}(c^d x^{d,|\tilde{T}_d|})\lambda_{d,|\tilde{T}_d|}$
$\sum_{d=1}^{|D|}\sum_{t=1}^{|\tilde{T}_d|}(A^d x^{d,|\tilde{T}_d|})\lambda_{d,|\tilde{T}_d|} = b \;;\; \pi$
$\sum_{t=1}^{|\tilde{T}_d|}\lambda_{d,|\tilde{T}_d|} = 1; \forall d \in D \;;\; \mu$
$\lambda_{d,|\tilde{T}_d|} \ge 0; \forall t \in \tilde{T}_d, d \in D$
Add variables $\lambda_{d,|\tilde{T}_d|}$
Reset $i = 0$
Yes
No
Is $i = |D|$?
Solve pricing subproblem d
$d = 1, |\tilde{T}_d| = |\tilde{T}_d| + 1$
$\min \sigma_d = (c^d - \pi A^d)x - \mu_d : x \in X^d \oplus$
$d = d + 1$
Is $\tilde{\sigma}_d < 0$?
Yes
$x^{d,|\tilde{T}_d|} = \tilde{x}^d$
No
No
$i = i+1$
Is $i = |D|$?
Yes
Phase 3 finishes. Continue to Phase 4.

Figure 2.8: Flowchart for Phase 3

9. Check if $d = |D|$. If no, go to Step 3. Otherwise, continue to next step.

10. Reset the counter $i = 0$.

11. Add variable $\lambda_{d,|\tilde{T_d}|}$. Go to Step 2.

12. Phase 3 finishes. Continue to Phase 4

2.2.2.4 Phase 4: Calculate the solution to the original problem

The solution to the original problem is the convex combination of all proposals from each pricing subproblem. Denoting x^{d^*} as the optimal solution for x^d, $\lambda^*_{d,|\tilde{T_d}|}$ as the optimal solution for $\lambda_{d,|\tilde{T_d}|}$, the optimal solution to the original problem can be calculated as follows. $x^{d^*} = \sum\limits_{t=1}^{|\tilde{T_d}|} x^{d,|\tilde{T_d}|}\lambda^*_{d,|\tilde{T_d}|}$. Figure 2.9 depicts a flowchart for Phase 4. The algorithm details follow.

1. Start Phase 4. Initialize $d = 1$.

2. Retrieve all proposals $x^{d,|\tilde{T_d}|}$ and $\lambda^*_{d,|\tilde{T_d}|}$.

3. Calculate $x^{d^*} = \sum\limits_{t=1}^{|\tilde{T_d}|} x^{d,|\tilde{T_d}|}\lambda^*_{d,|\tilde{T_d}|}$.

4. Update $d \leftarrow d + 1$.

5. Check if $d = |D|$. If yes, go to Step 6. Otherwise, go to Step 3.

6. Phase 4 finishes. Algorithm terminates.

Next we discuss the Dantzig-Wolfe decomposition technique for integer programs (DWIP) presented by Wolsey [70].

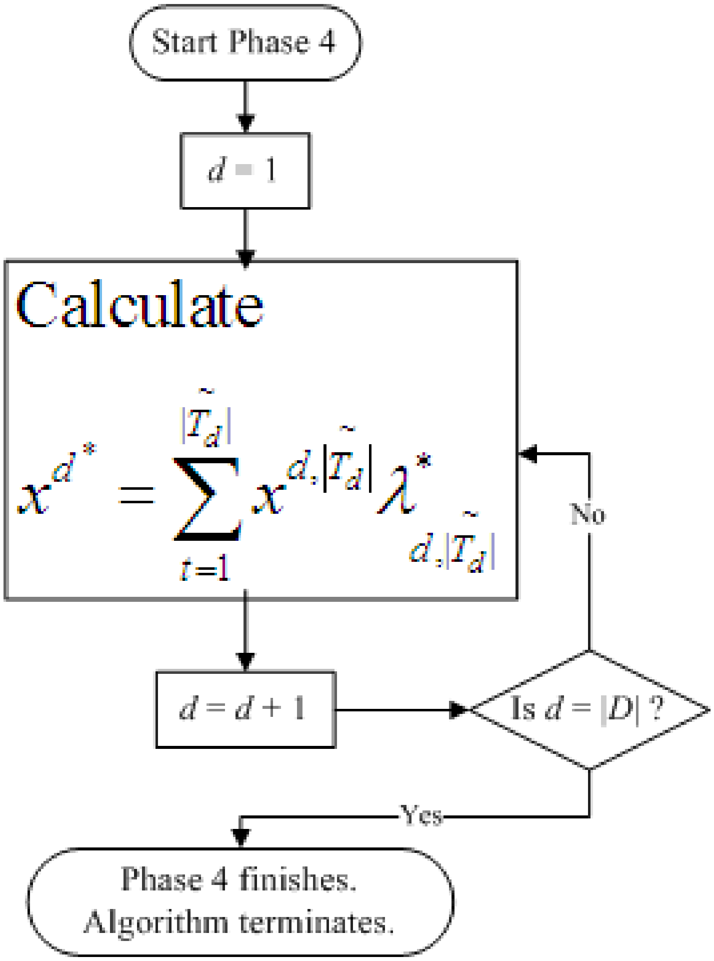


Figure 2.9: Flowchart for Phase 4

2.2.3 Dantzig-Wolfe Decomposition of an Integer Program

Consider the integer program below similar to Problem (2.9), except that the set X^d now also has integer restrictions on the variables x^d: $X^d = \{x^d \in Z_+^{n_d} : E^d x^d \leq f_d\}$.

$$\min \sum_{d=1}^{|D|} c^d x^d$$

subject to

$$\begin{array}{cccccc} A^1x^1 & +A^2x^2 & +\dots & +A^dx^d & \leq & b \\ E^1x^1 & & & & \leq & f_1 \\ & \dots & & & \leq & . \\ & & \dots & & \leq & . \\ & & & E^{|D|}x^{|D|} & \leq & f_{|D|} \\ x^1 \in Z_+^{n_1}, & \dots & \dots & ,x^{|D|} \in Z_+^{n_{|D|}} & & \end{array} \tag{2.14}$$

It is shown that Problem (2.14) still has the same decomposable structure as Problem (2.9) we presented earlier and may be rewritten as follows.

$$\text{(IP)} \qquad \begin{array}{l} \min \sum_{d=1}^{|D|} c^d x^d : \sum_{d=1}^{|D|} A^d x^d = b : x^d \in X^d; \forall d \in D \\ X^d = \{x^d \in Z_+^{n_d} : E^d x^d <= f_d\}; \forall d \in D \end{array} \tag{2.15}$$

Now assuming that each set X^d contains a large but finite set of points $\{x^{d,t}\}_{t=1}^{|T_d|}$, we may write $X^d = \{x^d \in Z^{n_d} : x^d = \sum_{t=1}^{|T_d|} \lambda_{d,t} x^{d,t}, \sum_{t=1}^{|T_d|} \lambda_{d,t} = 1, \lambda_{d,t} \in \{0,1\}; \forall t \in T_d$. Now substituting for x^d leads to an equivalent Integer Programming Master Problem

(IPM) [70].

$$
\text{(IPM)} \qquad \begin{aligned}
&\min \sum_{d=1}^{|D|} \sum_{t=1}^{|T_d|} (c^d x^{d,t}) \lambda_{d,t} \\
&\sum_{d=1}^{|D|} \sum_{t=1}^{|T_d|} (A^d x^{d,t}) \lambda_{d,t} = b \qquad && ; \pi_i \\
&\sum_{t=1}^{|T_d|} \lambda_{d,t} = 1 \forall d \in D \qquad && ; \mu_d \\
&\lambda_{d,t} \in \{0,1\}; \ \forall t \in T_d, d \in D
\end{aligned} \tag{2.16}
$$

Next, the linear programming relaxation of the IPM is developed and has the same form as the LPM when we discussed the Dantzig-Wolfe decomposition for linear programs (will be called DWLP, for the sake of discussion). In fact, from this point on the formulations of the master problem and pricing subproblems for DWIP are very similar to those of DWLP. The only difference is the definition of the set X^d in the pricing subproblems, where $X^d \in R_+^{n_d}$ for DWLP and $X^d \in Z_+^{n_d}$ for DWIP. To this end, the Dantzig-Wolfe decomposition algorithm we have presented above can be directly applied to the integer programming, however, with only slight changes for the definition of the set X^d we discussed above.

2.3 Benders Decomposition

In this section, we discusses the problem where if some variables are fixed to certain values, the resulting problem becomes either decomposable into several smaller problems or simpler to solve. One example of practical problem is the long-term multi-period investment planning problem [19], where the operation decisions are continuous variables and often decomposable by time period when the integer investment decisions are fixed to given values. Next, let's consider the optimization of the following form:

$$\min cX + dY : AX + BY \geq 0; X, Y \geq 0, \tag{2.17}$$

where $A \in R^{m\times n_1}, B \in R^{m\times n_2}$, c, d are row vectors of dimension R^{n_1} and R^{n_2}, X, Y are column vectors of dimension R^{n_1} and R^{n_2}, respectively. The Benders decomposition applies to (2.17) when [34]:

1. for fixed Y, (2.17) decomposes into several independent optimization problems, each containing a different subvector of X;

2. for fixed Y, (2.17) takes a well-known special structure (e.g., transportation problem), where efficient solution algorithms are are available, and

3. Problem (2.17) is not a convex program in X and Y jointly, but fixing Y yields it so in X.

Benders decomposition is proved to be useful in the problem structures we mentioned above. Nevertheless, we have implemented the solution algorithm presented in [19] to our optimization problem and found that the algorithm convergence is not promising after considerable amount of run time. The non-convergence is presumably due to the structure and/or size of our optimization problem. The detailed solution algorithm can be found at [19, 34, 53].

In this chapter, we have discussed the multiobjective optimization program theory and solution techniques. Later, we discuss the Dantzig-Wolfe decomposition technique used to decompose a big problem into several smaller pricing subproblems and one master problem. In addition, we also provided a solution algorithm in detail. The decomposition technique is shown to be useful in Chapter 5, where it is employed to significantly improve the problem's lower bounds on the objective function. Consequently, we can numerically prove that several integer solutions obtained are already close to optimality. Finally, we briefly discuss the Benders decomposition and show that what type of problem can take advantages of this decomposition method.

In the next chapter, we discuss the wastewater treatment processes at the Blue Plains advance wastewater treatment facility located in Washington, DC, for which served as our case study. Later, we discuss the odor statistical model developed by Gabriel et al. [32] used to predict biosolids odors produced at the Blue Plains facility. Finally, data and sources are discussed followed by odor threshold index calculations.

Chapter 3

Wastewater Treatment Processes, Odor Prediction Statistical Models, Data and Sources, and Odor Threshold Calculations

In this chapter we will briefly discuss the wastewater treatment process at DCWASA, which is the case study for this dissertation. Then, we will discuss odor prediction statistical models developed by Gabriel et al. [32]. Lastly, we will examine the data input for our optimization models described in Chapter 4 and their sources followed by odor threshold calculations.

3.1 Wastewater Treatment Processes

DCWASA currently handles more than 330 million gallons a day of raw sewage from area jurisdictions and this figure is anticipated to increase to 370 million gallons a day by 2010 [25]. The initial treatment stage starts as a discarded refuse where debris and grit are separated and transported to a landfill. The remaining sewage then effuses into the primary sedimentation tanks where approximately half of the floating solids are removed from the liquid. The liquid empties into the secondary treatment tanks where oxygen is infused so bacteria can break down the organic substance. During the next phase of treatment, the bacteria transforms ammonia into innocuous nitrogen gas. Residual solids settle to the bottom and the water is drained down through sand filters that purge the remaining floating solids and phosphorus. The water is sanitized and expelled into the Potomac River. Figure 3.1 shows the liquid flow diagram [25].

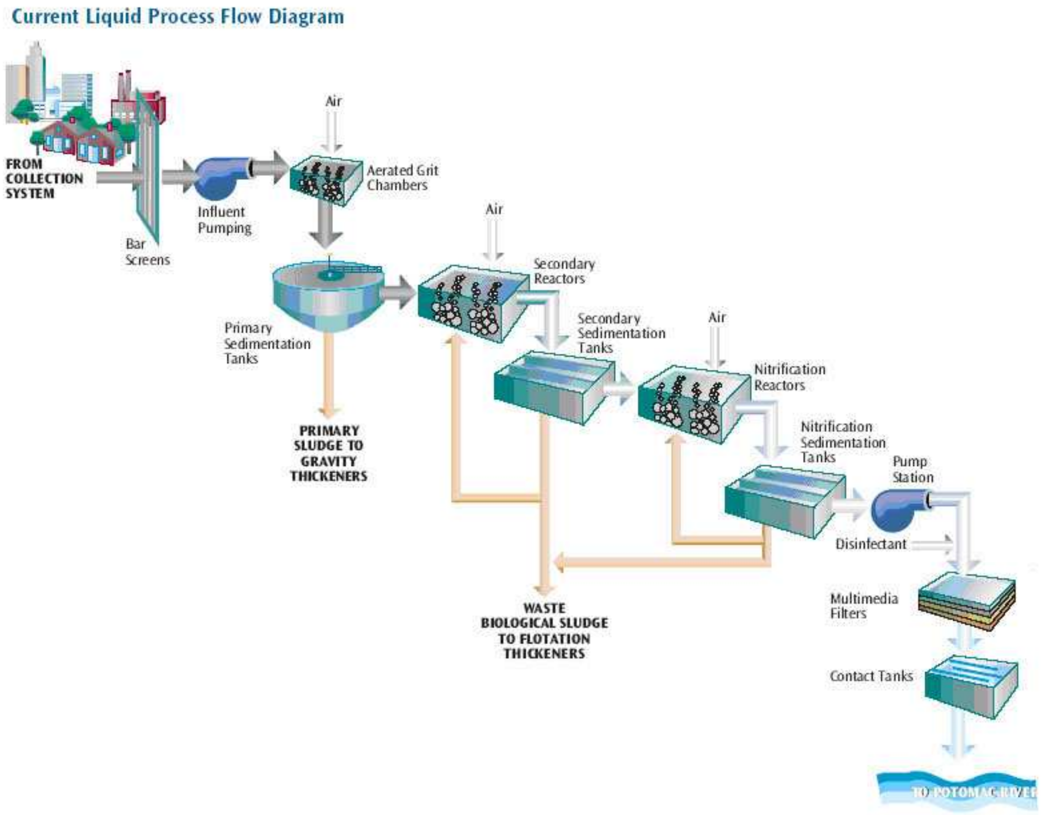

Current Liquid Process Flow Diagram
FROM COLLECTION SYSTEM
Bar Screens
Influent Pumping
Air
Aerated Grit Chambers
Air
Primary Sedimentation Tanks
Secondary Reactors
Secondary Sedimentation Tanks
Air
Nitrification Reactors
Nitrification Sedimentation Tanks
Pump Station
Disinfectant
PRIMARY SLUDGE TO GRAVITY THICKENERS
WASTE BIOLOGICAL SLUDGE TO FLOTATION THICKENERS
Multimedia Filters
Contact Tanks
TO POTOMAC RIVE

Figure 3.1: Liquid Process Flow Diagram [25]

The sludge or solid residue, which is a semisolid precipitated mass or deposit, in the primary sedimentation tanks proceeds to large vessels where gravity causes the concentrated sludge to settle to the bottom and thicken [25]. Sludge from the secondary sedimentation tanks and nitrifications reactors are thickened separately in dissolved air flotation (DAF) thickeners. The thickened sludge from the gravity and DAF thickeners is then blended in the blend tanks. The blended sludge is partially dewatered by an on-site contractor providing additional dewatering equipment (i.e., belt filter presses and centrifuges) and the remaining by DCWASA. Pathogens are eliminated by a lime-stabilization process. The on-site contractor adds lime both prior to (pre-lime) and after (post-lime) dewatering while DCWASA only does post-lime. The organic biosolids are then applied to farmland in Maryland and Virginia. Figure 3.2 displays the solids-handling process diagram. In the next section, we discuss two odor statistical models developed by Gabriel et al. [32], where one of them was used as a linear constraint in our optimization models.

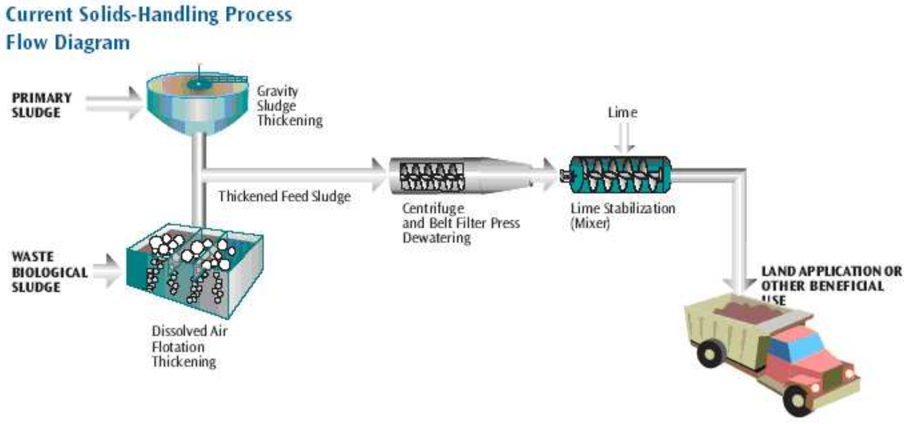

Current Solids-Handling Process
Flow Diagram
PRIMARY SLUDGE
Gravity Sludge Thickening
Lime
Thickened Feed Sludge
Centrifuge and Belt Filter Press Dewatering
Lime Stabilization (Mixer)
WASTE BIOLOGICAL SLUDGE
LAND APPLICATION OR OTHER BENEFICIAL USE
Dissolved Air Flotation Thickening

Figure 3.2: Solids-Handling Process [25]

3.2 Odor Prediction Statistical Models

Gabriel et al. [32] have demonstrated that biosolids odor may be predicted from some variables in the treatment process as well as ambient conditions. Before delving into the developed odor prediction models, we will briefly describe each independent variable as well as its effects on biosolids odor levels.

3.2.1 Polymer Addition

In wastewater treatment, polymer is used to condition biosolids in thickening or dewatering processes and eases the breakup of solids by combining small particles into larger ones or “flocs” [58]. Murthy et al. [45], Kim et al. [41], as well as Gabriel et al. [31, 32] have reported that polymer addition may result in an increase of biosolids odor level, especially the production of trimethyl amine (TMA). This finding is confirmed in equations (3.1) and (3.2), where the more polymer added the higher the biosolids odor level [32].

3.2.2 Number of Centrifuges or Belt Filter Presses in Service

Centrifuges and belt filter presses are used in a dewatering process. By cutting down water content of biosolids and increasing the solids concentration, biosolids volume is significantly decreased during this process [58]. Besides reducing volume and therefore saving cost of biosolids storage and transportation, dewatering may provide other benefits such as getting rid of liquids before landfill disposition and cutting down fuel needs if residuals are to be incinerated or heat dried [60, 59].

In centrifugal dewatering, water is removed from wastewater solids by the force generated from a rapid rotation of a cylindrical bowl [60]. Whereas, a belt filter press squeezes out water by putting on pressure to the biosolids [59]. Figures 3.3, 3.4, and 3.5 display a typical centrifuge thickening and dewatering system, a schematic of a belt filter press, and a dewatered solids cake dropping from belt filter press, respectively. At DCWASA, at the time the odor prediction statistical models

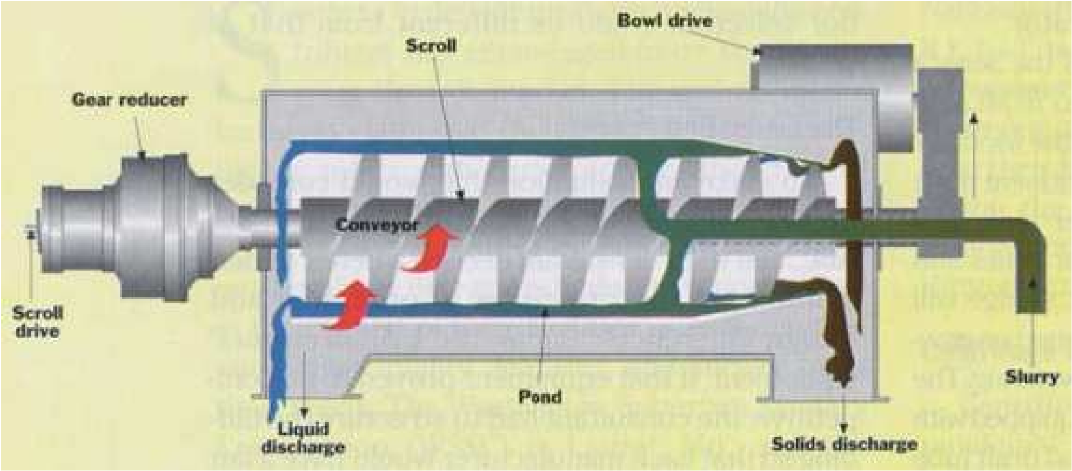


Figure 3.3: Typical centrifuge thickening and dewatering system [60]

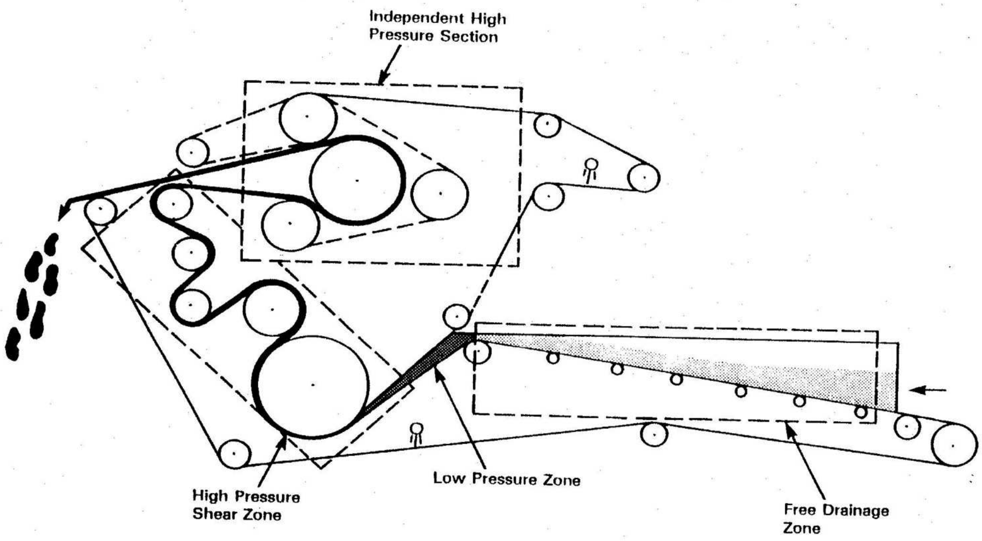


Figure 3.4: Schematic of a belt filter press [59]

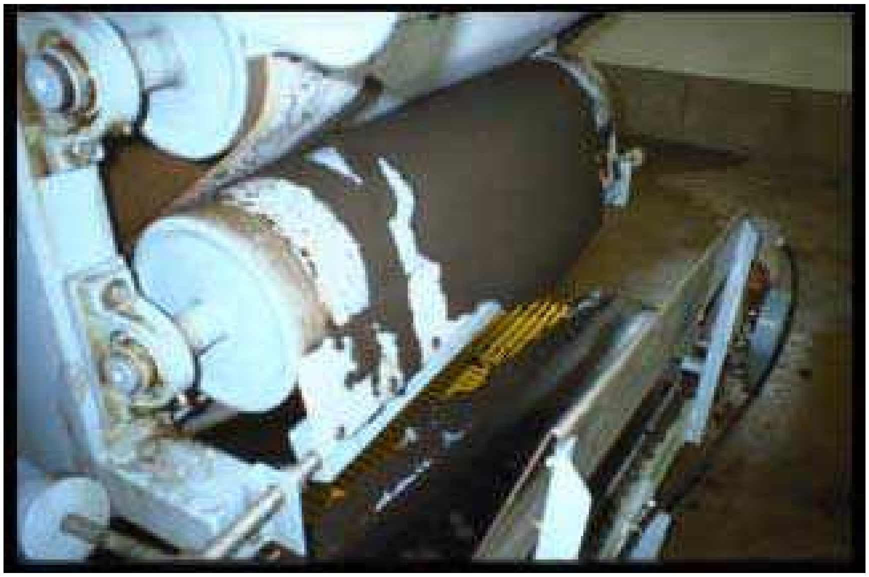


Figure 3.5: Dewatered solids cake dropping from belt filter press [59]

that we refer to later were developed, there were seven centrifuges in operation. In addition, there were two centrifuges as well as seven belt filter presses operated by a contractor. An important point is that when there is an insufficient number of centrifuges, there is more retention time of sludge in the line of wastewater treatment. Recently, Gabriel et al. [31, 32] have suggested the higher number of centrifuges in service the lower the biosolids odor level.

3.2.3 Temperature

Arispe [1] has reported that the higher ambient temperatures may result in a more septic condition of the sewer lines entering the plant as well as create more anaerobic conditions in the treatment process. Consequently, the amount of reduced sulfur compounds, resulting in over all biosolids odor, in the wastewater and sludge may increase [1]. Another study by Kim et al. [40] also indicated that high temperatures may promote higher microbial activity and result in lower oxidation-reduction potential (ORP) conditions of sludge from DAF and gravity thickener. More reduced sulfur compounds could then be observed due to this lower ORP [1, 40]. Recently, Gabriel et al. [31, 32] have demonstrated in odor prediction statistical models that the higher temperatures the lower biosolids odor level could be detected, given all things being equal.

3.2.4 Sludge Blanket Depth

The sludge blanket depth was identified as the depth of suspended solids at bottoms of settling tanks [31, 32]. Gabriel et al. [31, 32] proposed that the sludge blanket depth should be positively correlated with biosolids odor level due to the longer retention time for activated sludge in the settling tanks when the blanket depth was high. Sludge blanket depths were collected at three different locations at the DCWASA facility: secondary west even tanks, secondary west odd tanks, and secondary east tanks.

3.2.5 Lime Addition

During the last stage of the solids process, biosolids are stabilized with lime to raise the pH level as well as the temperature. Subsequently, some pathogens are eliminated before biosolids are hauled to the fields for beneficial reuse. Murthy et al. [45] suggested that an increase in a lime dose may decrease the biological activities and thus less reduced sulfur compounds were observed. Also, Gabriel et al. [32] have demonstrated in their odor prediction statistical model (see equation (3.2), page 55) that the more lime added the less biosolids odor detected.

Having introduced all relevant independent variables, we next show two odor prediction statistical equations [32] used as linear constraints in our optimization models.

3.2.6 First Odor Prediction Statistical Model

The first model [32] to predict biosolids odors is the following:

$$Y_i = -1.03 + 0.14X_1 - 0.32X_2 + 0.07X_3 + 0.47X_4 + 0.58X_5 - 0.96X_6 \tag{3.1}$$

where,

Y_i = Inspector P's biosolids odor score on day i in the winter period,

$X_j = j^{th}$ independent variable.

The independent variables in order were as follows.

1. The amount of dewatered polymer on day d (lbs/dry ton solids (DTS))
2. The number of DCWASA centrifuges in service on day d
3. The minimum temperature on day $d - 1$ (°F)
4. The blanket depth for the secondary east tank on day $d - 1$ (feet)
5. A dummy variable for when the sum of blanket depth from all locations on day $d - 1$ was greater than 10.3 feet (0 or 1)
6. A dummy variable for when the average temperature on day $d - 1$ was less than 43 °F (0 or 1)

Besides the independent variables we have previously discussed, there were two additional independent dummy variables, X_5 and X_6, as it can be seen in Equation (3.1). The value of each dummy variable equals one when its associated condition is met. The value of 10.3 feet appearing in the associated condition to X_5, represents the 80% fractile determined from the observed data [32]. Whereas, the value of 43 °F in X_6 represents the 20% fractile. Gabriel et al. [32] indicated an adjusted R^2 of approximately 0.60 was achieved for this odor prediction model. In other words, this model explained about 60% of the variation in the biosolids odor levels accounted for by the explanatory variables. In addition, with the exception of the intercept, most coefficient values were statistically distinct from zero at the 10% level [32]. Next, we present the second model, which had a higher adjusted R^2.

3.2.7 Second Odor Prediction Statistical Model

The second model from [32] is:

$$Y_i = 3.89 + 0.03X_1 + 0.98X_2 - 0.47X_3 - 1.91X_4 - 0.01X_5 + 0.56X_6 + 1.13X_7 \quad (3.2)$$

where,

Y_i = Inspector C's biosolids odor score on day i in the winter period,

$X_j = j^{th}$ independent variable.

The independent variables in order were as follows.

1. The minimum temperature on day $d-1$ (°F)

2. Blanket depth, secondary east tank on day $d-1$ (feet)

3. The number of contractor belt filter presses in service on day $d - 1$

4. The number of contractor centrifuges in service on day $d - 1$

5. The amount of lime used on day $d - 1$ (lbs/DTS)

6. A dummy variable for when the sum of polymer amount at Dissolved Air Floatation (DAF) and dewatered polymer on day $d - 1$ was greater than 200.05 lbs/DTS (0 or 1)

7. A dummy variable for when the lime amount on day $d - 1$ was less than 308 lbs/DTS (0 or 1)

Analogous to the first model, there were two additional independent dummy variables: X_6 and X_7. Moreover, the values of 200.05 lbs/DTS and 308 lbs/DTS represent the 80% and 20% fractiles, respectively. The adjusted R^2 for this model was 80.28% and most of the coefficient values were statistically distinct from zero at the 10% level except the intercept [32].

3.3 Data and Sources

In this section, we examine all data used in our optimization model and their sources. Since the statistical equations were developed during 2002 and some treatment processes have been changed since then, in order for validity of these equations, wastewater treatment process data as well as others were taken from the year 2002. It is crucial to note that some data could not be used in its original form but needed to be modified first. The necessary computational modifications are also described in what follows.

3.3.1 Centrifuge and Belt Filter Press' Operation and Maintenance Cost Data

The costs of operating and maintaining one centrifuge was approximated to be between \$65/DTS to \$209/DTS [60]. Generally less efficient, operation and maintenance costs of one belt filter press were approximately between \$80/DTS to \$200/DTS [59]. At DCWASA, the operation and maintenance costs of one centrifuge were estimated at \$196/DTS (C. Peot, personal communication, December 2, 2004). Without access to the on-site contractor's belt filter press data, we assumed that the operation and maintenance costs of one belt filter press were \$200/DTS.

3.3.2 On-Site Contractor's Dewatering and Lime Stabilization Costs

The following costs were obtained from C. Peot (personal communication, December 16, 2004). The dewatering costs were \$71.75 per DTS for the first 150 DTS and then \$50.25 per DTS thereafter. Not only did the on-site contractor provide lime-stabilized biosolids after dewatering (post-lime) but they also applied lime before dewatering (pre-lime). The costs for pre-lime and post-lime were \$10.10 per DTS and \$7.40 per DTS, respectively. It is also important to note that the pre-lime process only applied to 67% of the flow to the contractor.

3.3.3 DCWASA's dewatering Cost

Despite great efforts to obtain DCWASA's dewatering costs, we were not able to obtain these data, due no doubt to the large and decentralized nature of the facility. This cost had an important role in our optimization model notwithstanding and was generally higher than the on-site contractor's dewatering cost (C. Peot, personal communication, December 16, 2004). Thus, the compromise solution was to vary DCWASA's dewatering cost between \$70 and \$90 per DTS as part of a sensitivity analysis.

3.3.4 Chemical Costs

Our optimization models included two types of chemicals: polymer and lime. Their costs were \$1.26 per pound and \$0.06 per pound for polymer and lime, respectively (C. Peot, personal communication, December 16, 2004).

3.3.5 Biosolids' Hauling Cost

DCWASA arranged with three contractors to haul biosolids to reuse fields. Hauling costs were charged per wet ton of biosolids. In other words, no distances were taken into account. For proprietary purposes, we named these three contractors contractor 1, contractor 2, and contractor 3. The hauling costs for contractors 1, 2, and 3 were \$42.50, \$46.00, and \$25.14 per wet ton, respectively (C. Peot, personal communication, December 16, 2004).

3.3.6 Temperature Data

Temperature data were collected from the National Climate Data Center (NCDC) website (http://www.ncdc.noaa.gov/oa/ncdc.html.). To represent the temperature at DCWASA, temperatures at the National Airport Station in Washington, DC, the closest station to DCWASA, were collected. Three types of daily temperatures were used: minimum, maximum, and average. A sample of the temperature data from January 1, 2002 to January 8, 2002 is shown in Table 3.1.

Table 3.1: Temperature in degrees Fahrenheit on selected days

Date	Day	Maximum Temperature	Minimum Temperature	Average Temperature
1/1/2002	Tuesday	34	20	27
1/2/2002	Wednesday	37	21	29
1/3/2002	Thursday	36	30	33
1/4/2002	Friday	41	27	34
1/5/2002	Saturday	51	26	38.5
1/6/2002	Sunday	39	29	34
1/7/2002	Monday	39	30	34.5
1/8/2002	Tuesday	40	28	34

3.3.7 Wastewater Treatment Processing Data

Wastewater treatment processing data were obtained from DCWASA's process control historical (PCH) database. The data required in our optimization model included sludge blanket depths at three different locations, DAF polymer, dewatered polymer, and lime additions, numbers of DCWASA and contractor's centrifuges in service, and lastly the number of belt filter presses in service. Sample data from January 1, 2002 to January 8, 2002 are given in Tables 3.2, 3.3, and 3.4.

Table 3.2: Blanket depth in feet on selected days

Date	Blanket depth secondary east	Blanket depth secondary west odd	Blanket depth secondary west even
1/1/2002	2.3	2.6	3
1/2/2002	2.6	2.6	3
1/3/2002	3.3	3.3	3.4
1/4/2002	3.1	3.1	3.5
1/5/2002	3.1	3.1	2.9
1/6/2002	3.3	3.3	3.5
1/7/2002	3.7	3.9	4.7
1/8/2002	4.1	3.3	3.4

Table 3.3: Polymer and lime additions in lbs/DTS on selected days

Date	DAF polymer addition	Dewatering polymer addition	Lime addition
1/1/2002	190.55	0.78	269.91
1/2/2002	229.85	3.13	275.73
1/3/2002	146.95	3.31	277.03
1/4/2002	182.11	7.3	273.49
1/5/2002	131.41	3.64	259.92
1/6/2002	87.94	18.28	262.82
1/7/2002	208.86	14.84	265.77
1/8/2002	144.23	19.14	264.88

Table 3.4: Numbers of centrifuges and belt filter presses in service on selected days

Date	#of DCWASA's centrifuges in service	# of contractor's centrifuges in service	# of contractor's belt filter press in service
1/1/2002	4	1	4
1/2/2002	2	1	4
1/3/2002	3	1	4
1/4/2002	4	1	4
1/5/2002	4	1	4
1/6/2002	4	1	4
1/7/2002	4	1	4
1/8/2002	3	2	4

3.3.8 Wind Direction

Wind direction for each reuse field was recorded by a contractor when biosolids were applied at that particular field and was used along with other variables to calculate an odor threshold for each reuse field. In particular, the numbers of people and key institutions affected by biosolids odors in the downwind areas were determined according to a wind direction for each reuse field. Then, these numbers would then be used to calculate an odor threshold for each reuse field as discussed in Section 3.4. It was very difficult to actually assign prevailing wind direction to each field as wind direction may vary for each time of visit. In addition, there were only one or two visits for most of the fields during the time period in question. Therefore, the wind direction for each reuse field used in the optimization model was obtained by arbitrarily picking one of the wind directions that were recorded. Samples wind direction assigned to reuse fields are shown in Table 3.5. It is noted here that if more historical wind directions for each reuse field were available, statistical analysis on wind directions could be performed to determine either a dominant wind direction or several, with appropriate weights.

3.3.9 Reuse Field Data

Starting from October 2001 to the end of October 2006, there were approximately 5900 reuse fields receiving biosolids from DCWASA. This number was obtained from the Maryland Environmental Service (MES) database. In choosing candidates or feasible reuse fields for our optimization model runs, we queried visited reuse fields during a time period in question. For example, when running the optimization model that covered the time period during January of 2002, we can query all visited reuse fields during a year 2002. These fields would then be candidates or feasible fields for the run. The choice of the number of fields to use is a complicated one. Using all 5900 fields as input for a multiobjective optimization was computationally prohibitive in terms of obtaining Pareto optimal solutions. Furthermore, there were also some constraints not allowing all fields to be candidates for a biosolids application at the same time. One example of these constraints was the nutrient depository limit of each field not allowing the application of biosolids at the same farm in a certain time period. In particular, we used one year worth of visited fields.

Besides the geographic coordinates of each field, the MES database also provided tonnage capacity and wind direction at the time the field was inspected. After we plotted each field location in Virginia and Maryland, and by running an Avenue script in ArcView 3.2a (see the script in Appendix A), we then obtained more information for each field such as the total number of people, number of schools and hospitals, and total length of streets within a three-mile radius of the reuse field. The procedures used to obtain this information will be discussed in the following subsections. This information was then used to determine the odor threshold for each reuse field (see Section 3.4). Figure 3.6 shows a map of five selected reuse fields and surrounding schools, hospitals, streets and block group population density in St. Mary's County, Maryland. All map layers were originally in North American 1983 geographic coordinated system (GCS). Later, they were projected onto the

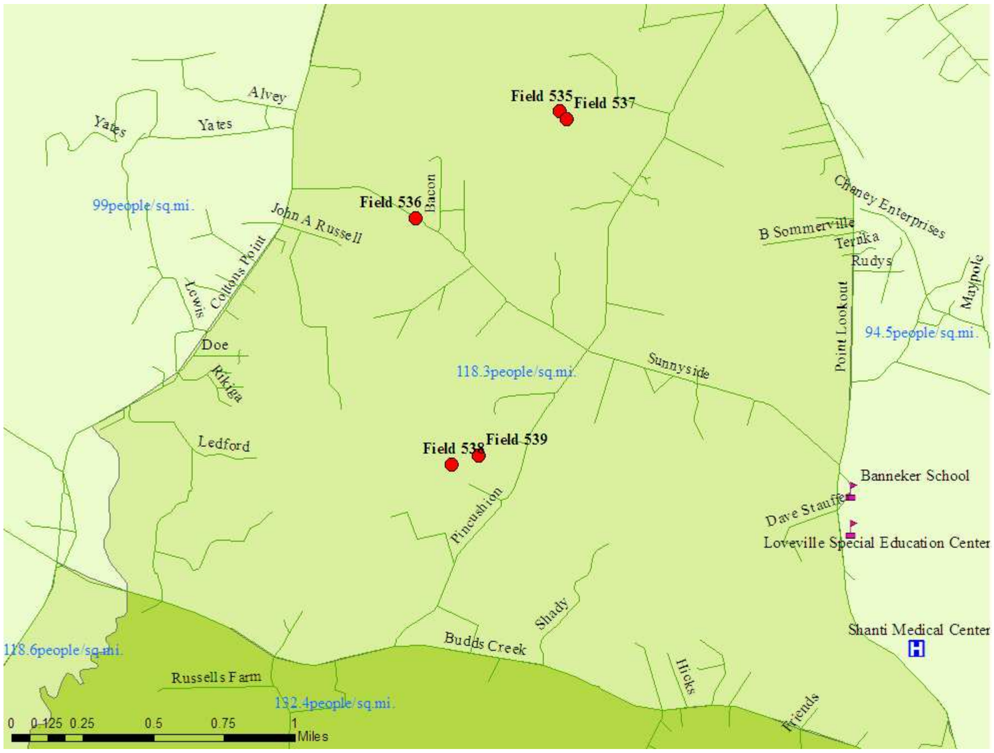


Figure 3.6: Reuse fields, schools, hospitals, streets and block group population densities, St. Mary's County, Maryland

Maryland State Plane Coordinate System, North American Datum (NAD) 1983, in horizontal units of feet. All calculations for obtaining GIS data were done within the specified projected coordinate system. Locations of all schools, hospitals and streets and block group population densities were obtained from Environmental Systems Research Institute, Inc. (ESRI)'s Data & Maps 2003 DVD ROM. Table 3.5 displays five selected reuse fields with their latitudes, longitudes, biosolids tonnage capacities, wind directions at the time of visit, population densities, total numbers of people,

Table 3.5: Five selected reuse fields with their latitudes, longitudes, biosolids tonnage capacities, wind directions at the time of visit, population densities, total number of people, schools, hospitals, length of streets within a three-mile radius from each reuse field

Field ID	Latitude of field centroid	Longitude of field centroid	Capacity (tons)	Wind Direction	Population density (people/sq.mi.)	# of people (0–1 mile)	# of people (1–2 miles)	# of People (2–3 miles)	# School (0–1 mile)	# School (1–2 miles)	# School (2–3 miles)	# Hospital (0–1 mile)	# Hospital (1–2 miles)	# Hospital (2–3 miles)	Length of Street (miles) (0–1 mile)	Length of Street (miles) (1–2 miles)	Length of Street (miles) (2–3 miles)
535	38.34928	−76.69468	89.82	SSW	118.30	367.45	1021.82	2053.36	1	4	2	0	0	1	13.14	44.19	57.83
536	38.34384	−76.70407	87.45	SW	118.30	360.57	1031.35	1857.14	0	3	4	0	0	1	13.63	39.71	60.03
537	38.34888	−76.69426	39.19	SSW	118.30	367.68	1018.76	2046.00	1	4	2	0	0	1	13.37	43.86	57.49
538	38.33125	−76.70180	38.93	N	118.30	375.86	1074.05	1613.28	0	3	1	0	1	0	13.06	37.03	56.20
539	38.33169	−76.70005	75.12	N	118.30	375.12	1077.68	1623.02	0	3	2	0	1	0	11.62	38.47	56.78

schools, and hospitals and total lengths of streets within each mile radius up to a distance of three miles. Similar to this table, Table 3.6 displays those numbers that were in downwind areas for the same reuse fields.

Table 3.6: Total number of people, schools, and hospitals and total length of streets in downwind areas within a three-mile radius of each reuse field

# of people (downwind, 0–1 mile)	# of people (downwind, 1–2 miles)	# of people (downwind, 2–3 miles)	# of school (downwind, 0–1) mile	# of school (downwind, 1–2 miles)	# of school (downwind, 2–3 miles)	# Hospital (downwind, 0–1 mile)	# Hospital (downwind, 1–2 miles)	# Hospital (downwind, 2–3 miles)	Length of Street (miles) (downwind, 0–1 mile)	Length of Street (miles) (downwind, 1–2 miles)	Length of Street (miles) (downwind 2–3 miles)
38.03	125.82	330.68	0	1	0	0	0	0	1.23	6.9	7.5
38.06	105.9	214.35	0	0	0	0	0	0	0.55	6.89	7.94
37.97	122.87	329.9	0	1	0	0	0	0	1.43	6.8	7.26
40.56	127.8	172.99	0	0	0	0	0	0	1.48	3.59	5.16
40.29	127.79	184.7	0	0	0	0	0	0	1.53	4.03	5.27

Next, the procedures used to obtain data in Tables 3.5 and 3.6 are discussed.

3.3.10 Field Tonnage Capacity

Field tonnage capacity was obtained by multiplying available acreage (acres) by application rate (tons/acre). The available acreage and application rate for each reuse field were obtained from MES's database CD ROM.

3.3.11 Population Density

Population densities (people/square mile) were also obtained from ESRI's Data & Maps 2003 DVD ROM. The DVD ROM provides population densities at three levels, i.e., county, tract, and block group. Tracts are small, relatively permanent statistical subdivisions of a county or statistically equivalent entity in accordance with Census Bureau guidelines. A block group contains several blocks, where each block is bounded on all sides by visible features such as streets, streams, and railroad tracks, and by invisible boundaries such as city, town, and county limits. The block group population densities are considered as the lowest level of population densities available and provide most detailed data. Therefore, we used the block group population densities to calculate the total number of people in a specified area. The calculation can be found in the next section.

3.3.12 Total Number of People

The total number of people was determined by the inner product of a block group population density and its area in question. The areas in question for each reuse field may be composed of several block groups (whole or part of each block group). Figure 3.7 shows a particular field (# 536) with the circles in one, two, and a three-mile radius. To demonstrate how we calculated a total population within the area in question, we next discuss the calculation of total population within a one-mile radius of Field 356. It can be seen that areas within the 0–1-mile radius from Field 536 were composed of two pieces, each from different block groups. Note

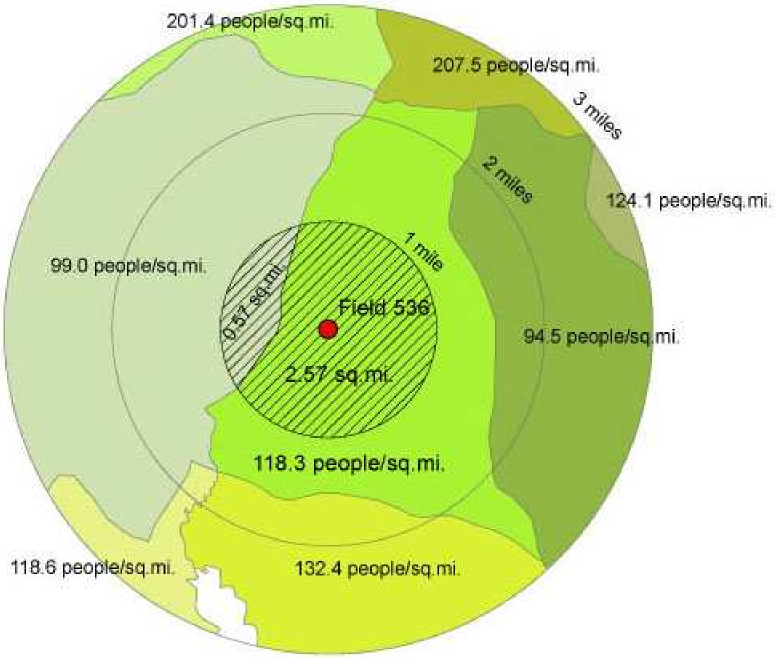


Figure 3.7: Block group layer within a three-mile radius of Field 536

that each block group is distinguished by different colors. In particular, the smaller piece with an area of 0.57339 sq.mi. has a block group population density of 99 people/sq.mi. and the larger piece with the area of 2.56805 sq.mi. has a block group population density of 118.3 people/sq.mi. Therefore, total population within a 0–1-mile radius of Field 536 is calculated as:

$$0.57339 \text{ sq.mi.} \times 99 \ \frac{\text{people}}{\text{sq.mi}} + 2.56805 \text{ sq.mi} \times 118.3 \ \frac{\text{people}}{\text{sq.mi}} = 360.57 \text{ people}$$

Using similar calculations, we determined the total number of people within two and three-mile radii. Then the total number of people within the 1–2-mile and 2–3-mile radii were obtained by subtraction of appropriate population totals.

In addition to calculating the total number of people in a specified radius, it is interesting to see how the optimization results would change, when the wind direction was taken into account for the affected areas. Hence, we calculated the total number of people located in the downwind directions. This total population figure would then be used in calculations of the second set of odor threshold in Section 3.4.2. Figure 3.8 demonstrates how we took wind direction into account. First, we assumed that the odor would gradually disperse laterally on each side of the wind direction up to one mile at the end of the three-mile radius. For the sake of discussion, let's consider the right triangle D with sides a, h, and o displayed in Figure 3.9. An adjacent side of angle b, denoted a, represents a three-mile radius in a downwind direction. An opposite side of angle b , denoted o, represents the dispersion distance of one mile at the end of the three-mile radius and forms a right angle with a. Point E represents the center of a reuse field. Now, we have that $\tan b = \frac{o}{a} = \frac{1}{3}$. Subsequently, we also have that $b = \arctan \frac{1}{3} \cong 32°$. Therefore, this assumption yields an angle of $\arctan \frac{1}{3}$ on each side of the wind direction. Next, we drew a line with an angle of $\arctan \frac{1}{3}$ on each side of a line representing the

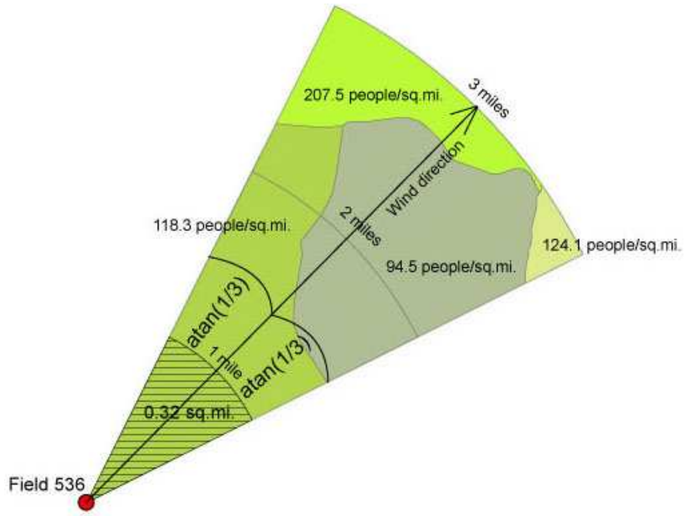


Figure 3.8: Block group in downwind areas within a three-mile radius of Field 536

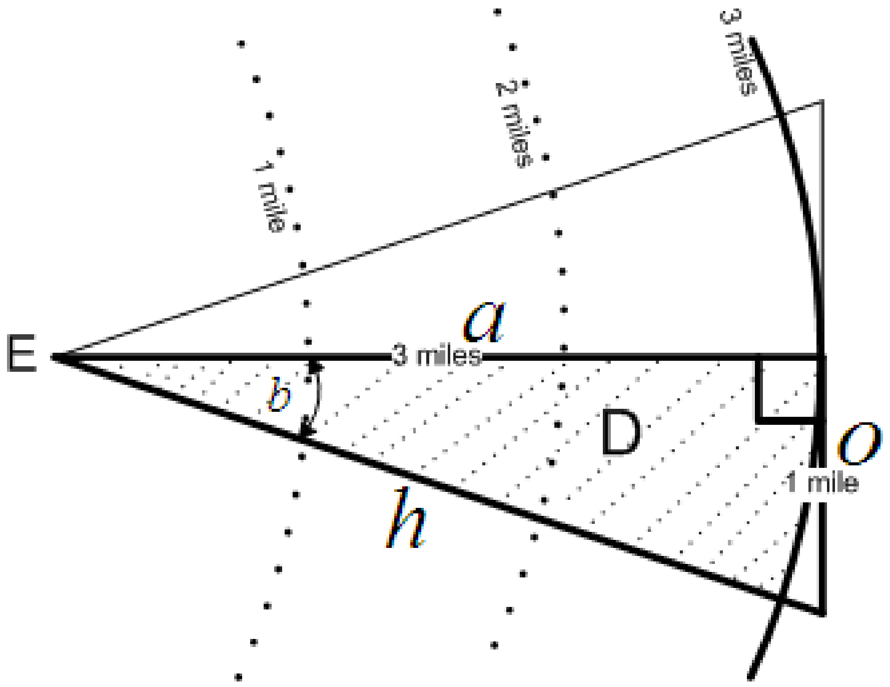


Figure 3.9: A right triangle with a hypotenuse side = 1 mile and an adjacent side = 3 miles

downwind direction. These two lines formed an infinite cone running outward from the center of the reuse field. We then found the intersections of this cone with the circle area in 1, 2, and a 3-mile radius, respectively. The hatched shaded area shown in Figure 3.8 exhibits the intersection of the cone with the area in a 0–1-mile radius. It represented the affected area according to a wind direction in a 0–1-mile radius. Using a similar method to calculate a total population discussed earlier in this subsection, the total affected population according to the wind direction within a 0–1-mile radius could be calculated as:

$$0.32173 \text{ sq.mi.} \times 118.3 \, \frac{\text{people}}{\text{sq.mi}} = 38.06 \text{ people}$$

The total affected population according to the wind direction within a 1–2-mile radius and a 2–3-mile radius was calculated in a similar manner when we calculated total affected population within 1–2-mile radius and 2–3-mile radius, without wind direction. We will discuss how we use different versions of a total population calculations to derive several sets of odor thresholds in Section 3.4. Next, we discuss a total length of streets calculations as it relates to odor threshold calculations.

3.3.13 Total Length of Streets

Analogous to a total population calculation, we also calculated the total length of streets within 0–1, 1–2, and a 2–3 mile radii as well as those in downwind areas. The circles and cone for each reuse field were identical to those used in a total population calculation of a same reuse field. Figure 3.10 displays the streets within a three-mile radius of a particular field (# 536).

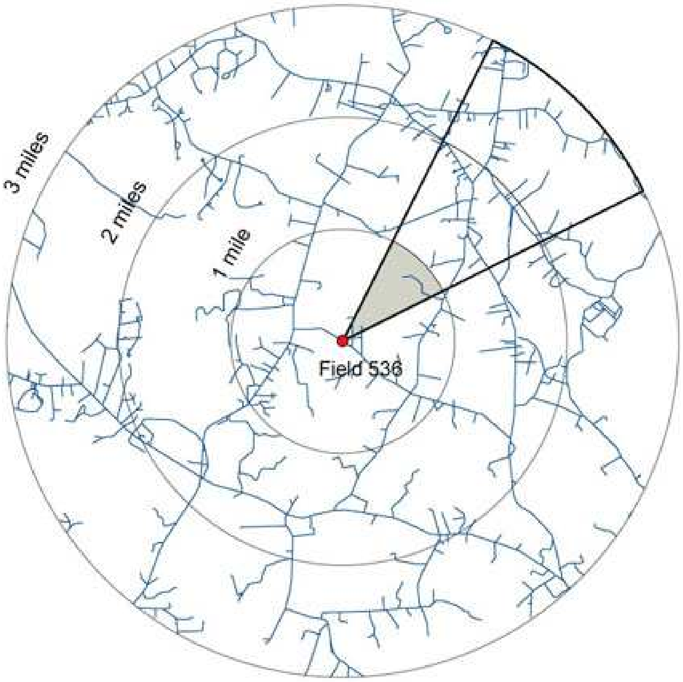


Figure 3.10: Streets within a three-mile radius of Field 536

3.3.14 Total Number of Schools

The calculations of a total number of schools within each mile radius and each downwind area were more straightforward than the calculations of a total number of people and a total length of streets for each reuse field. We simply drew one, two, and a three-mile-radius circle and a line with an angle of arctan $\frac{1}{3}$ on each side of a line representing the downwind direction. Then, the number of schools within each area in question was counted. Figure 3.11 displays the schools within a three-mile radius of a particular field (# 536).

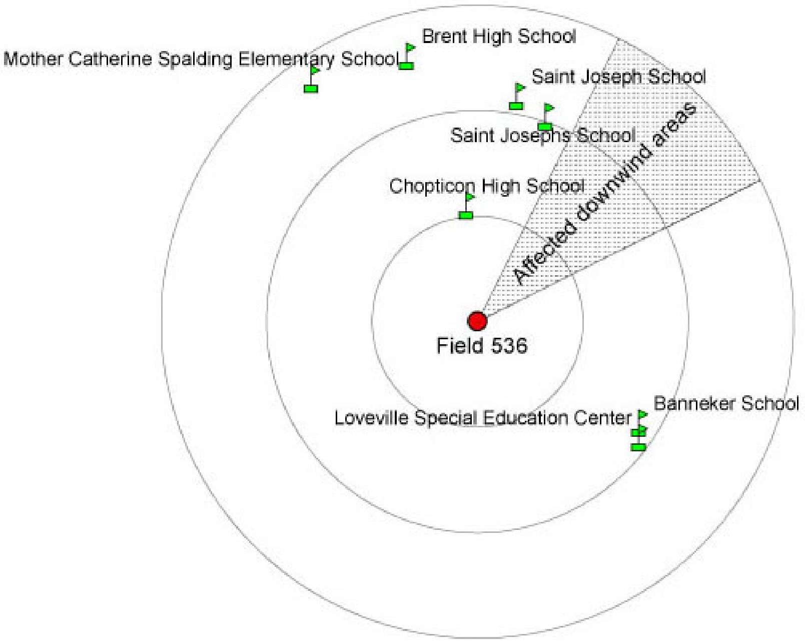


Figure 3.11: Schools within a three-mile radius of Field 536

3.3.15 Total Number of Hospitals

The calculations of a total number of hospitals within each mile radius and each downwind area were analogous to the calculations of the number of schools.

Figure 3.12 displays the hospitals within a three-mile radius of Field 536.

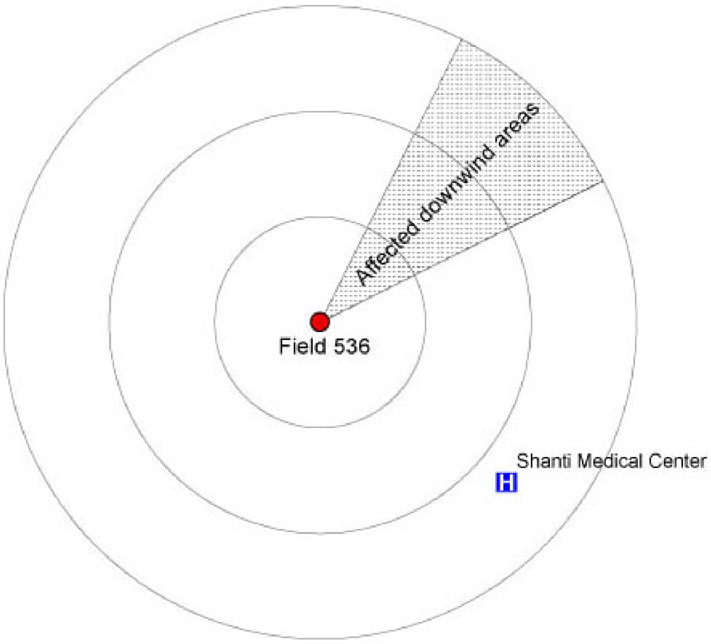


Figure 3.12: Hospitals within a three-mile radius of Field 536

Next, we discuss how we derived the odor threshold for each reuse field from the total number of population, number of schools, number of hospitals, and length of streets.

3.4 Odor Thresholds for Each Reuse Field

To aid in the decision of where to deliver biosolids as previously described, data were collected on the local population, length of streets, and number of key local institutions (i.e., schools, hospitals). All things being equal, odor thresholds should be lower for more densely populated areas, and locations with close proximity to schools, hospitals, and streets reflecting the sensitivity to the affected areas.

The biosolids odor scores were determined subjectively by field inspectors with the range of scores from 0 (least malodorous) to 9 (most malodorous). The odor thresholds for each reuse field f were not available from the inspectors and were thus computed from scratch. In particular, many choices for these values are possible. However, since the malodorous aspects usually affect people, factors that involved the local population should be used. Also, all things being equal, the more people potentially affected by the odors (within a given radius), the lower the threshold should be. Using these guiding principles, we established the following criteria in order to determine an odor threshold index for each reuse field. When possible, it is preferable to always apply biosolids to a field that would affect less people, less schools, less hospitals, and less total length of streets in a three-mile radius. In particular, these key parameters would be more affected by biosolids' odor when they were located near a reuse field. Therefore, besides quantities of these key parameters, distances to a reuse field would be taken into account when the odor thresholds were calculated. In addition, the three-mile radius figure representing the traveling distance of odor through air was obtained from A. Razik (personal communication, November 5, 2004), a specialist in biosolids land application.

We developed two sets of odor thresholds to vary inputs to the optimization models. The difference between the two sets was that the first one did not take wind direction into account for the calculation, while the second one did. It is also important to note that these two sets of odor thresholds were different than the odor thresholds developed in an earlier work by Gabriel et al. [30]. In this dissertation, we added more data (e.g., total number of hospitals, total length of streets in proximity area) into the calculations as an attempt to produce odor thresholds that more closely represented the relative odor thresholds of all reuse fields in question. Also, we captured as many factors as possible. Another major difference was how the population index was computed. In [30], the population index calculation simply used the reciprocal of the block group population density assigned to each reuse field based on its location. While, in this dissertation, the population index for each reuse field was calculated from the total population in proximity areas, which in some cases the total population calculations were involved more than just one block group density as can be seen in Section 3.3.12. Therefore, the population index calculated by the current approach better represents the number of people affected by biosolids odor. Consequently, the resulting odor indices from all reuse fields will be more accurately represented. Next, we discuss how we computed each set of odor threshold for each reuse field.

3.4.1 First Set of Odor Thresholds

The data used to calculate the first set of odor threshold values were total population, total number of schools, total number of hospitals, and total length of streets within a 0–1, 1–2, and a 2–3 mile radius. The calculations resulting a school index, population index, hospital index, street index, and an odor threshold index.

1. School index calculation

   The school index of each reuse field was calculated as follows:

$$SI_j = \begin{cases} 9 & \text{if } s^{0-3} = 0 \\ 8 \times \frac{s_j - \min\{s_j\}}{\max\{s_j\} - \min\{s_j\}} & \text{if } s^{0-3} \neq 0 \end{cases} \tag{3.3}$$

where

$$SI_j = \text{school index of field } j$$

$$s_j = \frac{1}{2.5s_j^{0\text{–}1} + 1.5s_j^{1\text{–}2} + 0.5s_j^{2\text{–}3}}$$

$$s_j^{0\text{–}3} = \text{total number of schools within 0–3 mile radius of field } j$$

$$s_j^{0\text{–}1} = \text{total number of schools within 0–1 mile radius of field } j$$

$$s_j^{1\text{–}2} = \text{total number of schools within 1–2 mile radius of field } j$$

$$s_j^{2\text{–}3} = \text{total number of schools within 2–3 mile radius of field } j$$

The calculation of a school index where there is no school within a three-mile radius was treated differently to prevent the reciprocal of zero. More specifically, a school index of 9, corresponded to a highest odor score examined by a field inspector, would be assigned to a reuse field with no school within a three-mile radius. Otherwise, a school index would be varied between 0 and 8. The min $\{s_j\}$ and max $\{s_j\}$ were used to normalize indices to be between these values. In addition, more weight should be given to parameters within a shorter mile radius. Thus, the coefficients 2.5, 1.5, and 0.5 were used as multipliers to represent the relative importance of the associated number of school within a 0–1, 1–2, and a 2–3 mile radius, respectively. The reciprocal shown in the calculation of s_j indicates that a higher number of schools in the proximity area would produce a lower school index score. Next, we discuss the population index calculation.

2. Population index calculation

   Defining PI_j as the population index of field j, we calculate the population index of each reuse field as follows.

$$PI_j = 9 \times \frac{p_j - \min\{p_j\}}{\max\{p_j\} - \min\{p_j\}} \tag{3.4}$$

   where

$$p_j = \frac{1}{2.5p_j^{0\text{–}1} + 1.5p_j^{1\text{–}2} + 0.5p_j^{2\text{–}3}}$$

   $p_j^{0\text{–}1}$ =total population within 0–1 mile radius of field j

   $p_j^{1\text{–}2}$ =total population within 1–2 mile radius of field j

   $p_j^{2\text{–}3}$ =total population within 2–3 mile radius of field j

   Next, we discuss the hospital index calculation.

3. Hospital index calculation

   The same terminology used to calculated a school index was used to calculate a hospital index. Defining HI_j as the hospital index of field j, we calculate the hospital index for each reuse field as follows.

$$HI_j = \begin{cases} 9 & \text{if } h^{0\text{–}3} = 0 \\ 8 \times \frac{h_j - \min\{h_j\}}{\max\{h_j\} - \min\{h_j\}} & \text{if } h^{0\text{–}3} \neq 0 \end{cases} \tag{3.5}$$

   where

$$h_j = \frac{1}{2.5h_j^{0\text{–}1} + 1.5h_j^{1\text{–}2} + 0.5h_j^{2\text{–}3}}$$

   $h_j^{0\text{–}3}$ =total number of hospitals within 0–3 mile radius of field j

   $h_j^{0\text{–}1}$ =total number of hospitals within 0–1 mile radius of field j

$h_j^{1\text{–}2}$ =total number of hospitals within 1–2 mile radius of field j

$h_j^{2\text{–}3}$ =total number of hospitals within 2–3 mile radius of field j

Next, we discuss the street index calculation.

4. Street index calculation

   Again, with the similar line of reasoning in mind, a street index calculation is similar to a population index calculation. Denoting RI_j as the street index of field j, we calculate the street index for each reuse field as follows.

$$RI_j = 9 \times \frac{r_j - \min\{r_j\}}{\max\{r_j\} - \min\{r_j\}} \tag{3.6}$$

   where

$$r_j = \frac{1}{2.5r_j^{0\text{–}1} + 1.5r_j^{1\text{–}2} + 0.5r_j^{2\text{–}3}}$$

   $r_j^{0\text{–}1}$ =total length of streets within 0–1 mile radius of field j

   $r_j^{1\text{–}2}$ =total length of streets within 1–2 mile radius of field j

   $r_j^{2\text{–}3}$ =total length of streets within 2–3 mile radius of field j

   Next, we combine population, school, hospital, and street indices to obtain odor thresholds.

5. Odor threshold calculation

   Denoting O_j^{up} as the odor threshold of reuse field j, we calculate an odor threshold for each reuse field as follows.

$$O_j^{\text{up}} = \frac{PI_j + SI_j + HI_j + RI_j}{4} \tag{3.7}$$

> We note here that taking an average of PI, SI, HI, and RI is just one way to arrive at an odor threshold. Other possible approaches would include taking the minimum or maximum of these four indices.

Table 3.7 displays the five selected fields[1] and their corresponding odor thresholds calculated by the five steps above.

Table 3.7: Five selected fields with associated odor thresholds set 1

Field ID	Odor Threshold
535	2.27964
536	2.36103
537	2.28051
538	0.45708
539	0.44523

3.4.2 Second Set of Odor Thresholds

The second set of odor thresholds were derived to take into account wind directions. These odor thresholds were computed similarly to the previous odor thresholds, the differences being that we assumed that people and/or key institutions located in the downwind directions would be more affected by the biosolids odor than others not located in the downwind directions. Consequently, greater multipliers were applied to people or key institutions located in the downwind directions, comparing to multipliers applied to people or key institutions located within

[1] For this data set of reuse fields: $\min\{p_j\} = 3.45 \times 10^{-5}$ people^{-1}, $\max\{p_j\} = 3.92 \times 10^{-3}$ people^{-1}, $\min\{s_j\} = 0.11$ people^{-1}, $\max\{s_j\} = 2.00$ people^{-1}, $\min\{h_j\} = 0.67$ people^{-1},$\max\{h_j\} = 2.00$ people^{-1}, $\min\{r_j\} = 6.34 \times 10^{-4}$ people^{-1},$\max\{r_j\} = 3.12 \times 10^{-2}$ people^{-1}.

the same mile radius, but not in the downwind directions. In particular, multipliers applied to people or key institutions within each mile radius would be increased by one if they were applied to numbers of people or key institutions in the downwind directions.

There were five steps used to compute this last set of odor thresholds. They were similar to the five steps we used to compute the first set of odor thresholds except that we took the wind directions into account as mentioned above. The data used to calculate this set of odor thresholds were total population, total number of schools, total number of hospitals, and total length of streets within a 0–1, 1–2, and a 2–3 mile radius, that were not in areas affected by downwind, and those that were in areas affected by downwind. Next, we present the five steps used to calculate the odor thresholds.

1. Population index calculation

   Denoting PI_j as the population index of field j, we calculate the population index of each reuse field as follows.

$$PI_j = 9 \times \frac{p_j - \min\{p_j\}}{\max\{p_j\} - \min\{p_j\}} \tag{3.8}$$

   where

$$p_j = \frac{1}{2.5p_j^{0\text{–}1} + 1.5p_j^{1\text{–}2} + 0.5p_j^{2\text{–}3} + 3.5pw_j^{0\text{–}1} + 2.5pw_j^{1\text{–}2} + 1.5pw_j^{2\text{–}3}}$$

   $p_j^{0\text{–}1}$ =total population within a 0–1 mile radius of field j

   less those in downwind areas

   $p_j^{1\text{–}2}$ =total population within a 1–2 mile radius of field j

less those in downwind areas

$p_j^{2\text{–}3}$ =total population within a 2–3 mile radius of field j

less those in downwind areas

$pw_j^{0\text{–}1}$ =total population in downwind areas within a 0–1 mile radius of field j

$pw_j^{1\text{–}2}$ =total population in downwind areas within a 1–2 mile radius of field j

$pw_j^{2\text{–}3}$ =total population in downwind areas within a 2–3 mile radius of field j

2. School index calculation

Denoting SI_j as the school index of field j, we calculate the school index of each reuse field as follows.

$$SI_j = \begin{cases} 9 & \text{if } s^{0\text{–}3} = 0 \\ 8 \times \frac{s_j - \min\{s_j\}}{\max\{s_j\} - \min\{s_j\}} & \text{if } s^{0\text{–}3} \neq 0 \end{cases} \tag{3.9}$$

where

$$s_j = \frac{1}{2.5s_j^{0\text{–}1} + 1.5s_j^{1\text{–}2} + 0.5s_j^{2\text{–}3} + 3.5sw_j^{0\text{–}1} + 2.5sw_j^{1\text{–}2} + 1.5sw_j^{2\text{–}3}}$$

$s_j^{0\text{–}3}$ =total number of schools within a 0–3 mile radius of field j

$s_j^{0\text{–}1}$ =total number of schools within a 0–1 mile radius of field j

less those in downwind areas

$s_j^{1\text{–}2}$ =total number of schools within a 1–2 mile radius of field j

less those in downwind areas

$s_j^{2\text{–}3}$ =total number of schools within a 2–3 mile radius of field j

less those in downwind areas

$sw_j^{0\text{–}1}$ =total number of schools in downwind areas

within a 0–1 mile radius of field j

$sw_j^{1\text{–}2}$ =total number of schools in downwind areas

within a 1–2 mile radius of field j

$sw_j^{2\text{–}3}$ =total number of schools in downwind areas

within a 2–3 mile radius of field j

3. Hospital index calculation

   Denoting HI_j as the hospital index of field j, we calculate the hospital index for each reuse field as follows.

$$HI_j = \begin{cases} 9 & \text{if } h^{0\text{–}3} = 0 \\ 8 \times \frac{h_j - \min\{h_j\}}{\max\{h_j\} - \min\{h_j\}} & \text{if } h^{0\text{–}3} \neq 0 \end{cases} \tag{3.10}$$

   where

$$h_j = \frac{1}{2.5h_j^{0\text{–}1} + 1.5h_j^{1\text{–}2} + 0.5h_j^{2\text{–}3} + 3.5hw_j^{0\text{–}1} + 2.5hw_j^{1\text{–}2} + 1.5hw_j^{2\text{–}3}}$$

$h_j^{0\text{–}3}$ =total number of hospitals within a 0–3 mile radius of field j

$h_j^{0\text{–}1}$ =total number of hospitals within a 0–1 mile radius of field j

less those in downwind areas

$h_j^{1\text{–}2}$ =total number of hospitals within a 1–2 mile radius of field j

less those in downwind areas

$h_j^{2\text{–}3}$ =total number of hospitals within a 2–3 mile radius of field j

less those in downwind areas

$hw_j^{0\text{–}1}$ =total number of hospitals in downwind areas

within a 0–1 mile radius of field j

$hw_j^{1\text{–}2}$ =total number of hospitals in downwind areas

within a 1–2 mile radius of field j

$hw_j^{2\text{–}3}$ =total number of hospitals in downwind areas

within a 2–3 mile radius of field j

4. Street index calculation

Denoting RI_j as the street index of field j, we calculate the street index for each reuse field as follows.

$$RI_j = 9 \times \frac{r_j - \min\{r_j\}}{\max\{r_j\} - \min\{r_j\}} \tag{3.11}$$

where

$$r_j = \frac{1}{2.5r_j^{0\text{–}1} + 1.5r_j^{1\text{–}2} + 0.5r_j^{2\text{–}3} + 3.5rw_j^{0\text{–}1} + 2.5rw_j^{1\text{–}2} + 1.5rw_j^{2\text{–}3}}$$

$r_j^{0\text{–}1}$ =total length of streets within a 0–1 mile radius of field j

less those in downwind areas

$r_j^{1\text{–}2}$ =total length of streets within a 1–2 mile radius of field j

less those in downwind areas

$r_j^{2\text{–}3}$ =total length of streets within a 2–3 mile radius of field j

less those in downwind areas

$rw_j^{0\text{–}1}$ =total length of streets in downwind areas

within a 0–1 mile radius of field j

$rw_j^{1\text{–}2}$ =total length of streets in downwind areas

within a 1–2 mile radius of field j

$rw_j^{2\text{–}3}$ =total length of streets in downwind areas

within a 2–3 mile radius of field j

Next, we combined all indices above and derived an odor threshold for each reuse field.

5. Odor threshold calculation

Denoting O_j^{up} as the odor threshold of reuse field j, we calculate an odor threshold for each reuse field as follows.

$$O_j^{\text{up}} = \frac{PI_j + SI_j + HI_j + RI_j}{4} \tag{3.12}$$

It is noted here that if several wind directions were assigned to a particular field with associated weights as discussed in Section 3.3.8, the above calculations for p_j, s_j, h_j, and r_j could be modified. In particular, the total numbers of people, schools, hospitals, and length of streets associated with each wind direction would be included in the calculations with their corresponding weights. One choice of the modified calculations might include:

$$p_j = \frac{1}{2.5p_j^{0-1} + 1.5p_j^{1-2} + 0.5p_j^{2-3} + 3.5\sum_{i=1}^{|I|}\mu_i pw_{ji}^{0-1} + 2.5\sum_{i=1}^{|I|}\mu_i pw_{ji}^{1-2} + 1.5\sum_{i=1}^{|I|}\mu_i pw_{ji}^{2-3}}$$

where μ_i is the appropriate weight associated with wind direction i and $pw_{ji}^{l\text{–}m}$ is the number of people within a l–m-mile radius of reuse field j in downwind direction i.

Table 3.8 displays the five selected fields[2] and their corresponding odor thresholds set 2 calculated by the five steps above. As indicated, odor thresholds in the second set and in the first set are not much different. This is due to the fact that the reuse fields are located in rural areas where there are not many schools, hospitals, and streets. In addition, the population densities in the areas surrounding reuse fields are not much different. Therefore, additional number of people, schools, hospitals, and length of streets in the downwind direction are very small compared to the those not in downwind area. Consequently, the calculated odor thresholds when wind directions were taken into account are not much different than ones that did not take wind directions into account.

Table 3.8: Five selected fields with associated odor thresholds set 2

Field ID	Odor Threshold
535	2.22964
536	2.32265
537	2.23113
538	0.77382
539	0.75717

In this chapter, we have discussed the wastewater treatment process at DCWASA. In addition, we have discussed all the input for our optimization models, and their sources and/or methods used to acquire the input were also provided. Now, that we have everything ready for the model construction, in the next chapter, we discuss

[2] For this data set of reuse fields: $\min\{p_j\} = 3.41 \times 10^{-5}$ people^{-1}, $\max\{p_j\} = 3.90 \times 10^{-3}$ people^{-1},$\min\{s_j\} = 9.52 \times 10^{-2}$ people^{-1}, $\max\{s_j\} = 2.00$ people^{-1},$\min\{h_j\} = 0.40$ people^{-1}, $\max\{h_j\} = 2.00$ people^{-1},$\min\{r_j\} = 5.88 \times 10^{-3}$ people^{-1}, $\max\{r_j\} = 2.95 \times 10^{-2}$ people^{-1}.

the optimization models, formulations, and results.

Chapter 4

Optimization Models, Formulations, and Results

In this chapter, we construct several optimization models for the problem at hand and report and analyze the resulting optimal solutions. Methods (e.g., algorithms), data input, assumptions, and constraints differentiate each model from another. Sensitivity analysis on some parameters and data input (e.g., % flows from the blend tank to DCWASA and the on-site contractor, odor thresholds) are also performed. Such analyses provide valuable information for wastewater treatment managers. For example, given that the flow from the blend tank to the on-site contractor changed from 90% to 80%, how would the tradeoff curve between cost and odor change?

The remaining parts of this chapter are organized as follows. Section 4.1 provides the problem statement and assumptions. In Section 4.2, notation and constraints are discussed. The notation and constraints were applied in several optimization models we developed and to avoid repetition, they are presented before we discuss the optimization models. On the other hand, notation and constraints that appear in just a particular model are discussed in the section for that model. Finally, from Section 4.3 on, we discuss each optimization model and its results. In addition, Pareto optimal solutions and insights are also discussed.

4.1 Problem Statement

DC Water and Sewer Authority generates approximately 1200 wet-tons of biosolids daily (personal communication, C. Peot, December 16, 2004). Biosolids, which are a nutrient-rich organic material, can be used as fertilizer for some crops (e.g., corn, hay). During each day, generated biosolids require removal from the wastewater treatment plant located at DCWASA and distribution to reuse fields located in Maryland and Virginia. Despite their beneficial reuse characteristics, biosolids derived from the end-product of the wastewater treatment process generally are considered as malodorous and DCWASA has received complaints from the communities surrounding some reuse fields. Since the problem has arisen, DCWASA has needed to design a treatment process that minimized the biosolids' odor. However, trying to minimize biosolids' odor could require costly treatment processes. Therefore, if biosolids were distributed to a less restricted reuse field, for example further away from populated areas, some treatment costs could be saved in the sense that they might require less treatment. These cost savings may come in part from chemical doses and/or number of centrifuges in service, to name a few areas. This situation is adequately modeled by a multiobjective optimization program to find *Pareto* optimal solutions which balance both cost and biosolids odor minimization.

It is also important to note that under certain circumstances (e.g., inaccessibility of reuse fields and frozen soil) when biosolids are unable to be land applied immediately, contingency planning usually involves storage along with alternate options of landfill disposal, incineration or alternative treatment and use, such as composting, heat drying, and advanced alkaline stabilization [62]. However, for this dissertation, we excluded these options to facilitate model construction. In other words, we assumed no storage of biosolids was allowed either at the treatment facility or at the fields. Along with this assumption, other assumptions we used throughout our models are the following:

1. Biosolids produced by the on-site contractor and DCWASA had the same biosolids' odor levels. This assumption was due to the fact that the statistical models for odor prediction were developed using observed biosolids' odor levels from combined biosolids from both the on-site contractor as well as DCWASA.

2. The dewatering capacities of one centrifuge and one belt filter press were approximately 50 DTS per day and 25 DTS per day, respectively (D. Tolbert, personal communication, November 18, 2003).

In summary, the optimal solutions that our optimization models provide are as follows for a given time horizon:

1. Optimal lime addition's dose on each day

2. Optimal numbers of on-site contractor centrifuges and belt filter presses in service

3. Optimal or preferred fraction of the flow handled by the on-site contractor and by DCWASA

4. Optimal biosolids shipment patterns including which hauling contractors to use and to which reuse fields biosolids should applied on each particular day

5. Trade off curve between cost and odor

Next, we discuss notations and constraints.

4.2 Notation and Constraints

4.2.1 Notation

The multiobjective optimization model corresponds to a given time horizon in which each day is denoted by $d \in D$. For example, if one week is chosen, $D = 1, 2, 3, 4, 5, 6, 7$. During this time frame, the biosolids are distributed to a set of reuse fields F with each field denoted as $f \in F$. The set of distribution contractors hauling the biosolids to these fields is given by C with each contractor denoted as $c \in C$. Next, the main notation used in the models is listed in alphabetical order.

B_d =Dry ton of biosolids on day d (tons)

B_d^{daf} =Dry ton of biosolids at DAF on day d (tons)

B_d^e =The blanket depth for secondary east tank on day d (feet)

C_d^{dc} =Number of DCWASA's centrifuges in service on day d

C_d^k =Number of on-site contractor centrifuges in service on day d

F_{dcf} =Amount of biosolids applied to field f by contractor c on day d (tons)

F_d^{dc} =Percent flow from blend tank to DCWASA on day d

F_d^k =Percent flow from blend tank to the on-site contractor on day d

F_f^{up} =Biosolids capacity of reuse field f (tons)

G_d^k =Number of on-site contractor belt filter presses in service on day d

H_{cf} =Hauling cost to field f by contractor c (\$/wet ton)

L_d^{dc} =The amount of lime DCWASA added on day d (lbs/DTS)

LD_d =A binary dummy variable for when the lime amount on day d was less than 308 lbs/DTS (0 or 1)

O_d =Biosolids' odor level on day d (real-valued and varies with data set)

O_f^{up} =Odor threshold for reuse field f (real-valued and varies with data set)

P_d^{daf} =Amount of polymer DCWASA added at the DAF on day d (lbs/DTS)

P_d^{de} =Amount of polymer DCWASA added at the dewatering process on day d (lbs/DTS)

PD_d =A binary dummy variable for when the sum of polymer amount at DAF and dewatered polymer on day d was greater than 200.05 lbs/DTS

T_d^{min} =Minimum temperature on day d (°F)

4.2.2 Constraints

First, there is a conservation of flow from the blend tank (%) between DCWASA and the on-site contractor.

$$\forall d \in D;\ F_d^{dc} + F_d^k = 1 \tag{4.1}$$

Since no storage of biosolids at the plant is modeled, conservation of biosolids means that

$$\forall d \in D;\ \frac{B_d F_d^{dc}}{0.3107} + \frac{B_d F_d^k}{0.2580} - \sum_{c=1}^{|C|} \sum_{f=1}^{|F|} F_{dcf} = 0 \tag{4.2}$$

The first and second left-most terms in (4.2) represent the daily wet tons of biosolids produced by DCWASA and the on-site contractor, respectively. The number 0.3107 and 0.2580 (personal communication, C. Peot, December 16, 2004) represent the percent solids of biosolids and were used to determine the final wet tons of biosolids produced by DCWASA and the on-site contractor, respectively. For example, if the flow from the blend tank is 10 dry-tons of solids and was handled by DCWASA, the final wet tons of biosolids produced by DCWASA would be $\frac{10}{0.3107} = 32.19$ tons. Moreover, the final wet tons of biosolids is the actual tonnage considered for the hauling weight. The next constraint states that the amount of biosolids distributed over the time horizon to each field cannot exceed the field's given capacity (in tons).

$$\forall f \in F;\ \sum_{d=1}^{|D|} \sum_{c=1}^{|C|} F_{dcf} \leq F_f^{\mathrm{up}} \tag{4.3}$$

The daily amount of biosolids hauled to reuse fields by each contractor c on each day d is limited by the following constraint based on actual considerations.

$$\forall d \in D, c \in C; \sum_{f=1}^{|F|} F_{dcf} \leq \begin{cases} 1200 & \text{if } c = 1 \\ 1200 & \text{if } c = 2 \\ 600 & \text{if } c = 3 \end{cases} \tag{4.4}$$

It is important to note that the right hand sides of constraint (4.4) represent the maximum tonnage of biosolids that can be assigned to each hauling-contractor. These maximum tonnages were 1200 wet tons for Contractors 1 and 2, and 600 wet tons for Contractor 3. The number 1200 was roughly the maximum tonnage of biosolids production on each day. The number 600 was roughly half of biosolids production and intended to prevent all biosolids being assigned to Contractor 3 due to its considerably cheaper contract's price (personal communication, C. Peot, December 16, 2004). Additional logic is needed to show that for each reuse field f , if any distribution is made, i.e., $F_{dcf} > 0$, then the odor threshold for that field should not be exceeded. In addition, for hauling efficiency, the minimum tonnage of each distribution should be at least 23 tons (personal communication, C. Peot, December 16, 2004). These restrictions are achieved by the following four constraints.

$$\forall d \in D, c \in C, f \in F;$$

$$F_{dcf} \geq 23\delta_{dcf} \tag{4.5}$$

$$F_{dcf} \leq F_f^{\text{up}}\delta_{dcf} \tag{4.6}$$

$$O_d \leq O_f^{\text{up}} + M_f(1 - \delta_{dcf}) \tag{4.7}$$

$$\delta_{dcf} \in \{0, 1\} \tag{4.8}$$

Constraint (4.5) clearly represents the restriction on minimum tonnage of each actual distribution, which is 23 tons. The tonnage capacity of each field, F_f^{up}, in constraint (4.6) implicitly bounds the maximum tonnage of each distribution F_{dcf}. More specifically, constraints (4.5), (4.6), and (4.8) work together as follows. When δ_{dcf} equals zero, F_{dcf} must equal zero and vice versa. On the other hand, when δ_{dcf} equals one, F_{dcf} must be greater than or equal to 23 but less than or equal to F_f^{up} and vice versa. In addition, when the tonnage capacity of any particular field f is less than 23 tons, constraints (4.5), (4.6) and (4.8) together imply $F_{dcf} = 0$. Constraints (4.7) and (4.8) represent the restriction on the odor threshold limit when there is a shipment ($F_{dcf} \geq 23$) or, identically, $\delta_{dcf} = 1$. When there is no shipment ($F_{dcf} = 0$) and subsequently $\delta_{dcf} = 0$, the odor score is bounded by the sum of O_f^{up} and M_f. The value of the parameter "M_f" should be picked such that the sum of itself and O_f^{up} equals the possible maximum odor level according to the data set and odor prediction statistical model used in each optimization model run. Having mentioned that, it is important to note that O_d will be substituted by the corresponding equation and variables according to the odor prediction statistical model (see Sections 3.2.6 and 3.2.7) chosen to be implemented in each optimization model. For example, if the odor prediction model from Section 3.2.7 (Equation (3.2)) is chosen, constraint (4.7) would be:

$$\begin{aligned}\forall d \in D;\ 3.89 + .03T_d^{min} + .98B_d^e - .47G_d^k - 1.91C_d^k - \\ .01L_d^{dc} + .56PD_d + 1.13LD_d \ &\leq\ O_f^{\text{up}} + M(1 - \delta_{dcf}).\end{aligned}$$

It is important to note that we apply index d to the variables in the odor prediction equation instead of imposing "$d-1$" index as already seen in (3.2). This is to simplify our model construction by assuming that F_{dcf}, to which δ_{dcf} corresponds, represents biosolids shipment on the following day although it has the index d. Our next three constraints, corresponding to the definition of LD_d, force LD_d to equal one whenever lime addition or L_d^{dc} is less than 308 lbs/DTS and zero, otherwise.

$$\begin{aligned}
&\forall d \in D; \\
-L_d^{dc} - 93LD_d &\leq -308 + \epsilon; \epsilon > 0 \text{ and small} && (4.9) \\
L_d^{dc} - 93(1 - LD_d) &\leq 308 - \epsilon && (4.10) \\
LD_d &\in \{0, 1\} && (4.11)
\end{aligned}$$

It is important to note that the applicability of the above three constraints is based on the assumption that the amount of lime addition was limited by 400 lbs/DTS. Our next constraint states that, on each particular day, there should be enough on-site contractor belt filter presses and centrifuges in service to process the flow assigned to them.

$$\forall d \in D; \ 50C_d^k + 25G_d^k \geq F_d^k B_d \qquad (4.12)$$

The figures 50 and 25 in (4.12) are the operational capacities in DTS of one centrifuge and one belt filter press, respectively (see assumptions in Section 4.3 Problem Statement). Thus, the left hand side of (4.12) represents the maximum flow in DTS that the on-site contractor can handle given the numbers of centrifuges and belt filter presses in service and it must be greater than or equal to the assigned flow from the blend tank on the right-hand side of this equation.

Bounds on variables are given below.

$$\forall d \in D;$$

$$250 \leq L_d^{dc} \leq 400 \tag{4.13}$$

$$0 \leq C_d^k \leq 2 \tag{4.14}$$

$$0 \leq G_d^k \leq 7 \tag{4.15}$$

Lastly, non-negativity and integer constraints are applied.

$$\forall d \in D;$$

$$F_d^{dc}, F_d^k \quad \geq \quad 0 \tag{4.16}$$

$$C_d^k, G_d^k \quad \in \quad \mathbb{Z}_+ \tag{4.17}$$

4.3 "Base Case" Optimization Model

The Base Case was performed to analyze the operations at DCWASA. For the Base Case study, the time horizon was set to 31 days using January 1, 2002–January 31, 2002 as a typical time period. Additionally, the model considered three hauling contractors consistent with actual operations. In considering the number of reuse fields to be included, we queried all visited reused fields during the year 2002 and removed the fields with incomplete data (e.g., no recorded geographic coordinates). Subsequently, there were 782 reuse fields with complete data to be included in the model. As for the daily DTS, it is not an easy task to measure for each day. In fact, DCWASA approximates the DTS on each day at 320 by multiplying the average daily wet ton solids with the approximated percent of solids. Thus, we assumed

320 DTS were processed on each day. Also, we assumed 90% of the flow from the blend tank handled by DCWASA and the remaining 10% was handled by the on-site contractor as Base Case values. Nevertheless, other percentages of flow were also tried in other models as part of a sensitivity analysis. In addition, the second odor prediction statistical model (see Section 3.2.7) and the second set of odor thresholds (see Section 3.4.2) were used in the Base Case. Next, we discuss the Base Case's optimization objectives and present the full optimization model.

The two objectives to be minimized simultaneously are the total odor, where

$$\text{total odor} = \sum_{d=1}^{|D|} O_d \tag{4.18}$$

and the total costs, where

$$\begin{aligned}\text{total costs} =& \text{DCWASA's lime cost} + \text{DCWASA's polymer cost} +\\ & \text{DCWASA's dewatering cost} + \text{DCWASA's centrifuge cost} +\\ & \text{on-site contractor's lime cost} +\\ & \text{on-site contractor's belt filter press cost} +\\ & \text{on-site contractor's centrifuge cost} +\\ & \text{on-site contractor's dewatering cost} + \text{hauling cost}\end{aligned} \tag{4.19}$$

and

$$\text{DCWASA's lime cost} = 0.06 \sum_{d=1}^{|D|} L_d^{dc} F_d^{dc} B_d \tag{4.20}$$

$$\text{DCWASA's polymer cost} = 1.26 \sum_{d=1}^{|D|} (B_d^{daf} P_d^{daf} + B_d P_d^{de} F_d^{dc}) \tag{4.21}$$

$$\text{DCWASA's dewatering cost} = 90 \sum_{d=1}^{|D|} B_d F_d^{dc} \tag{4.22}$$

$$\text{DCWASA's centrifuge cost}^{*} = 196\sum_{d=1}^{|D|}\lceil\frac{B_dF_d^{dc}}{50}\rceil^{**} \tag{4.23}$$

$$\text{on-site contractor's lime cost}^{***} = \sum_{d=1}^{|D|}\big((0.67)(10.10)B_dF_d^k + 7.40B_dF_d^k\big) \tag{4.24}$$

$$\text{on-site contractor's belt filter press cost} = 200\sum_{d=1}^{|D|}G_d^k \tag{4.25}$$

$$\text{on-site contractor's centrifuge cost} = 196\sum_{d=1}^{|D|}C_d^k \tag{4.26}$$

$$\text{on-site contractor's dewatering cost} = \sum_{d=1}^{|D|}\begin{cases} 71.75F_d^kB_d & \text{if } F_d^kB_d \leq 150 \\ 3225 + 50.25F_d^kB_d & \text{if } F_d^kB_d > 150^{****} \end{cases} \tag{4.27}$$

$$\text{hauling cost} = \sum_{d=1}^{|D|}\sum_{c=1}^{|C|}\sum_{f=1}^{|F|}H_{cf}F_{dcf} \tag{4.28}$$

*Since there is no variable for the number of DCWASA centrifuge in service in the odor prediction equation (3.2), we implicitly determine the number of DCWASA centrifuge needed from its capacity of 50 DTS per centrifuge and the amount of biosolids assigned to DCWASA. We note here that the number of DCWASA centrifuges determined here does not effect the odor calculation and is intended only to fairly determine costs incurred from the on-site contractor and DCWASA.

**$\lceil\ \rceil$ yields next closest integer rounded up (e.g., $\lceil 4.3\rceil = 5$)

***See on-site contractor's lime costs detail from Section 3.3.2.

****As shown in Section 3.3.2, the on-site contractor dewatering cost was $71.75 per DTS for the first 150 DTS and then $50.25 per DTS thereafter. Therefore, when more than 150 DTS were assigned, the dewatering cost could be determined from the dewatering cost for the first 150 DTS plus the dewatering cost thereafter or $71.75 \times 150 + 50.25(F_d^k B_d - 150) = 3225 + 50.25F_d^k B_d$.

Now, the full model of the Base Case is as follows.

$$\begin{aligned} &\text{minimize} && \text{Total odor} \\ &\text{minimize} && \text{Total cost} \\ &\text{subject to} && \text{(4.1)–(4.28)} \end{aligned} \tag{4.29}$$

and

$$O_d = 3.89 + .03T_d^{min} + .98B_d^e - .47G_d^k - 1.91C_d^k - .01L_d^{dc} + .56PD_d + 1.13LD_d \tag{4.30}$$

It is important to note that (4.30) corresponds to (3.2), however, with modified notation as discussed above.

4.3.1 Base Case Computational Results

The optimization model was solved using the XPRESS-MP optimization solver (version 17.01.02) with the modeling interface XPRESS-IVE (version 1.17.04) and the MOSEL language (version 1.6.2) (www.dashoptimization.com). The computer used was a Dell OPTIPLEX GX270, Pentium 4 with a CPU of 3.00 GHz and 2 GB of RAM memory. The model contained 219,394 constraints and 145,576 variables. In particular, there were 72,757 continuous variables, another 72,757 binary ones, and 62 non-binary discrete variables (relating to number of centrifuges and belt filter presses). These numbers were determined as follows:

1. Constraints (4.1), (4.2), (4.9), (4.10), and (4.12) were applied for each day. Therefore, there were $5 \times 31 = 155$ constraints.

2. Constraints (4.13), (4.14), and (4.15) were applied for each day and there were two constraints per day for each constraint. Therefore, there were $3 \times 2 \times 31 = 186$ constraints.

3. Constraint (4.3) was applied for each reuse field. Therefore, there were 782 constraints.

4. Constraint (4.4) was applied for each combination of day and contractor. Therefore, there were $31 \times 3 = 93$ constraints.

5. Constraints (4.5), (4.6), and (4.7) were applied for each combination of day, contractor, and reuse field. Therefore, there were $3 \times 31 \times 3 \times 782 = 218,178$ constraints.

Accounting for all of the above constraints, we obtained $155 + 186 + 782 + 93 + 218,178 = 219,394$ constraints (not including the binary integer constraints (4.8) and (4.11), the nonnegativity constraint (4.16), and the general integer constraint (4.17)). Next, the number of variables were determined as follows.

1. Variables F_{dcf} and δ_{dcf} involved combinations of days, contractors, and reuse fields. Therefore, there were $2 \times 31 \times 3 \times 782 = 145,452$ variables.

2. Variable L_d^{dc}, LD_d, C_d^k, and G_d^k were for each day. Therefore, there were $4 \times 31 = 124$ variables.

Accounting for all of the above variables, we obtained $145,452 + 124 = 145,576$ variables.

In finding Pareto optimal solutions, both weighting and constraint methods [17] were tried. Corresponding to the discussion in Section 2.1.2, the weighting problem for the Base Case was set up as follows.

$$\begin{array}{rcl} \text{minimize} & & (w^1 \times \text{Total odor}) + (w^2 \times \text{Total cost}) \qquad (4.31) \\ \text{subject to} & & (4.1)\text{–}(4.28), (4.30) \\ \text{and} & & \\ w^1 + w^2 & = & 1 \qquad (4.32) \\ w^1, w^2 & > & 0 \qquad (4.33) \end{array}$$

We begin our first subproblem by setting $w^2 = 0.01$ and $w^1 = 1 - w^2$. Then, the next subproblems were set up by increasing w^2 by 0.01 until $w^2 = 0.99$. Using this setup, 99 subproblems were tried. Although the weighting method is simple to implement, only one Pareto optimal point was obtained out of these 99 subproblems. In particular, the only Pareto optimal solution was obtain from the subproblem where $w^2 = 0.01$ and yielded the total cost of \$4,091,467.66978 and a total odor score of 135.86400[1]. The solution time was 942 seconds. For the other 98 subproblems, the solver could not finish within the preset maximum run time of 1,800 seconds. This was undoubtedly due to the non-convex nature of the models and, hence the existence of ”duality gap” points [17], discussed in Section 2.1.2. The solver might have been able to solve some subproblems to optimality if a larger preset maximum

[1] The unit for total odor is index point(s) summed over 31 days.

run time had been set, however, this was not desirable in practice, where several subproblems must be solved in order to generate the Pareto optimal set. It is also important to mention that, in fact, all solutions for the 98 subproblems were between 0.02% and 0.66% to the distance of their associated best (lower) bounds. These gaps are very acceptable for integer programming in practice. However, since we are to find the Pareto optimal set, the optimal solutions must be obtained otherwise the solutions can not be guaranteed to be non-inferior. Figure 4.1 displays the solutions for the 98 subproblems. Although the southwest envelope of Figure 4.1 does look like Pareto optimal points, however, they were actually not optimal solutions and should not be considered as Pareto optimal points.

The constraint method faired better in finding Pareto optimal solutions presumably due to its superior ability to find duality gap points, however with a large computational burden [49]. First, the total cost was minimized without constraining the total odor and the resulting total odor was recorded. Using this value as the maximum value τ, we set up the constraint problem corresponding to Section 2.1.1 as follows.

$$\begin{aligned} \text{minimize} \quad & \text{Total cost} \qquad (4.34) \\ \text{subject to} \quad & (4.1)\text{–}(4.28), (4.30) \end{aligned}$$

and

$$\sum_{d=1}^{|D|} O_d \leq \tau \qquad (4.35)$$

Then, the next subproblems were solved by decreasing the total odor constraint right-hand side τ by five until the problem was infeasible. The maximum run time

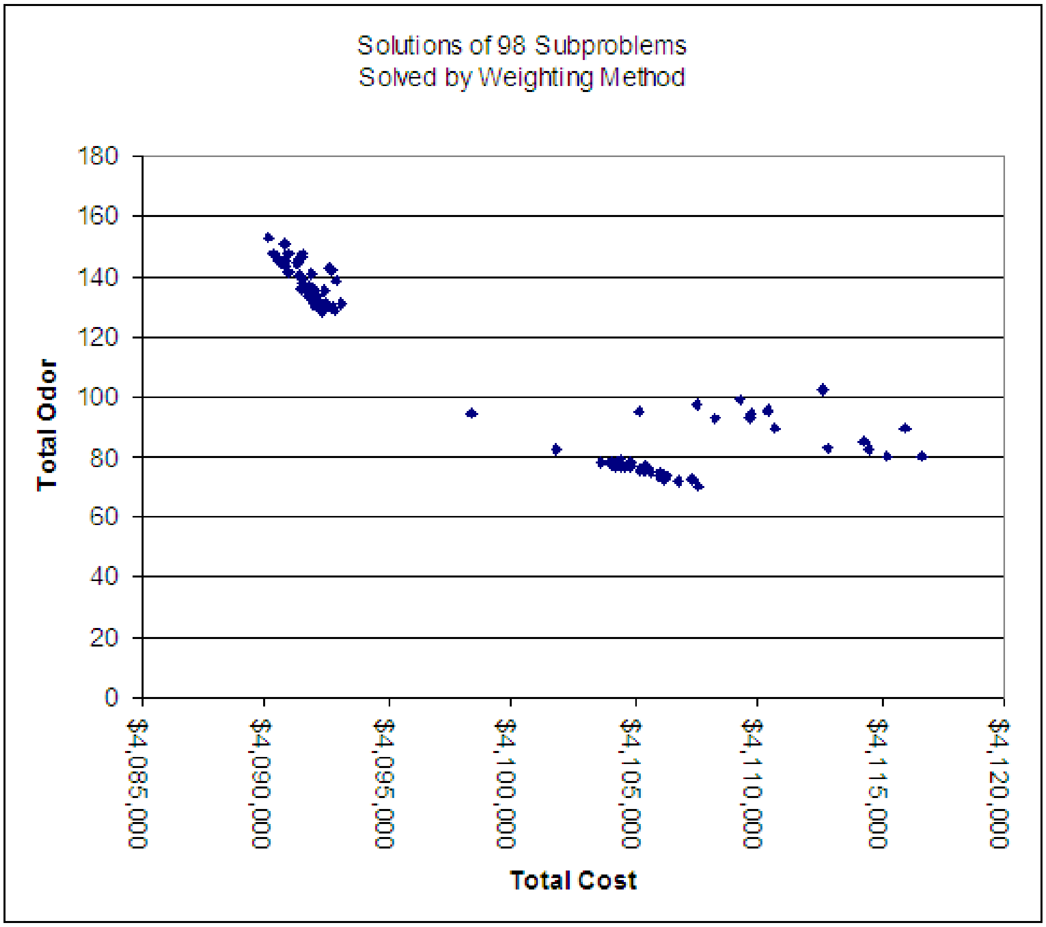


Figure 4.1: Solutions of 98 subproblems from Base Case solving by weighting method

for each subproblem was arbitrarily chosen to 1,800 seconds (for computational reasons). More specifically, if the solver could not solve a given problem to optimality within 1800 seconds, the solver stopped and reported the optimization status as *unfinished*. Then, the next subproblem was loaded and the solver continued. It is important to note that a longer preset maximum run time could have been used and some *unfinished* subproblems might have been solved to optimality. However, we experimented with several subproblems and found that most subproblems could be solved to optimality within 1,800 seconds. In addition, for subproblems that could not be solved to optimality within 1,800 seconds, most of them could not be solved within a number of hours and the large optimality gaps were observed for these subproblems.

Using the aforementioned setup, 51 optimization problems were tried. There were five subproblems which the solver could not solve to optimality. In particular, four subproblems were reported as *unfinished* and one subproblem (the last one) was reported as *infeasible* relevant to the set up. For 46 subproblems that the solver could solve to optimality, we verified the Pareto solutions by checking if the total odor constraints were binding (see Section 2.1.1 for verification of Pareto optimal solution in constraint method). After verifying, we found only 10 subproblems yielding Pareto optimal solutions (total odor constraints were binding). As for the other 36 subproblems, the total odor constraints were not binding and therefore these subproblems did not necessarily yield Pareto optimal solutions. Next, we proceed to discuss how additional Pareto optimal solutions were found.

Regardless of their optimization status (e.g. *optimal, unfinished*), we recorded 41 total cost values reported by the 41 subproblems that did not yield Pareto solutions. Each total cost value was used as the right hand side of the total cost constraint for each of the next 41 subproblems where total odor was the objective function to be minimized. In other words, we still used the constraint method to find Pareto optimal solutions. However, we used total cost as the objective constraint and total odor as the objective function in contrast to our previous setup. Using this setup, we were able to find 17 more Pareto optimal points. It is presumably that those new subproblems yielding Pareto optimal solutions could overcome the optimality gap due to the changes in its feasible region and objective function. Therefore, including the 10 points from before and the one point obtained from weighting method, there were 28 Pareto optimal points found in total. The run time as well as detailed solutions of these 28 Pareto optimal points can be found in Table 4.1. In particular, the run times varied from 40 to 1,545 seconds. It is important to note that the wide range of solution times is presumably caused by the optimality gap due to the non-convex nature of the problem. Next, we analyze these 28 Pareto points in detail.

4.3.2 Analysis of the Base Case

In this section, we analyze the optimal solutions obtained from the Base Case. Our analysis indicates that for various situations at hand (e.g., how many index points of odor level need to be decreased), each variable plays different roles in reducing biosolids odor levels. For example, for some situations, increasing the number of centrifuges in service was only a marginal activity for reducing biosolids odor levels. Our calculation shows that by increasing lime addition instead of the number of centrifuges could have caused additional expenses that were 1.2 times as much. Combinations of activities that are optimal are also important and not immediately obvious. For example, increasing lime additions and decreasing the number of belt filter presses may be the optimal combination for one position on the Pareto optimal curve, while the activities in opposite directions may be optimal for another. Next, we begin the analysis of the Base Case.

Figure 4.2 depicts the Pareto optimal points generated for the Base Case. First, note that certain odor scores were negative. This was due to the fact that the data in the time horizon selected went beyond the range of data used to generate the statistical model for odor prediction where the odor scores were in the range $[0, 9]$. It is noted here that the negative odor scores don't represent hedonic tone of the odor. Instead, these odor scores were only relative values and therefore, are still useful for management decisions. As indicated, the Pareto optimal points may be divided into three portions, (going left to right): first portion (Pareto optimal point number 1 to number 6), second portion (Pareto optimal point number 7 to number 22), and third portion (Pareto optimal point number 23 to number 28). More details on these Pareto optimal points can be found in Table 4.1.

Next, we estimated the number of dollars needed to reduce the odor by one index point for each portion of the Pareto optimal solutions as follows. To obtain these numbers, statistical regressions on the three portions of Pareto optimal points

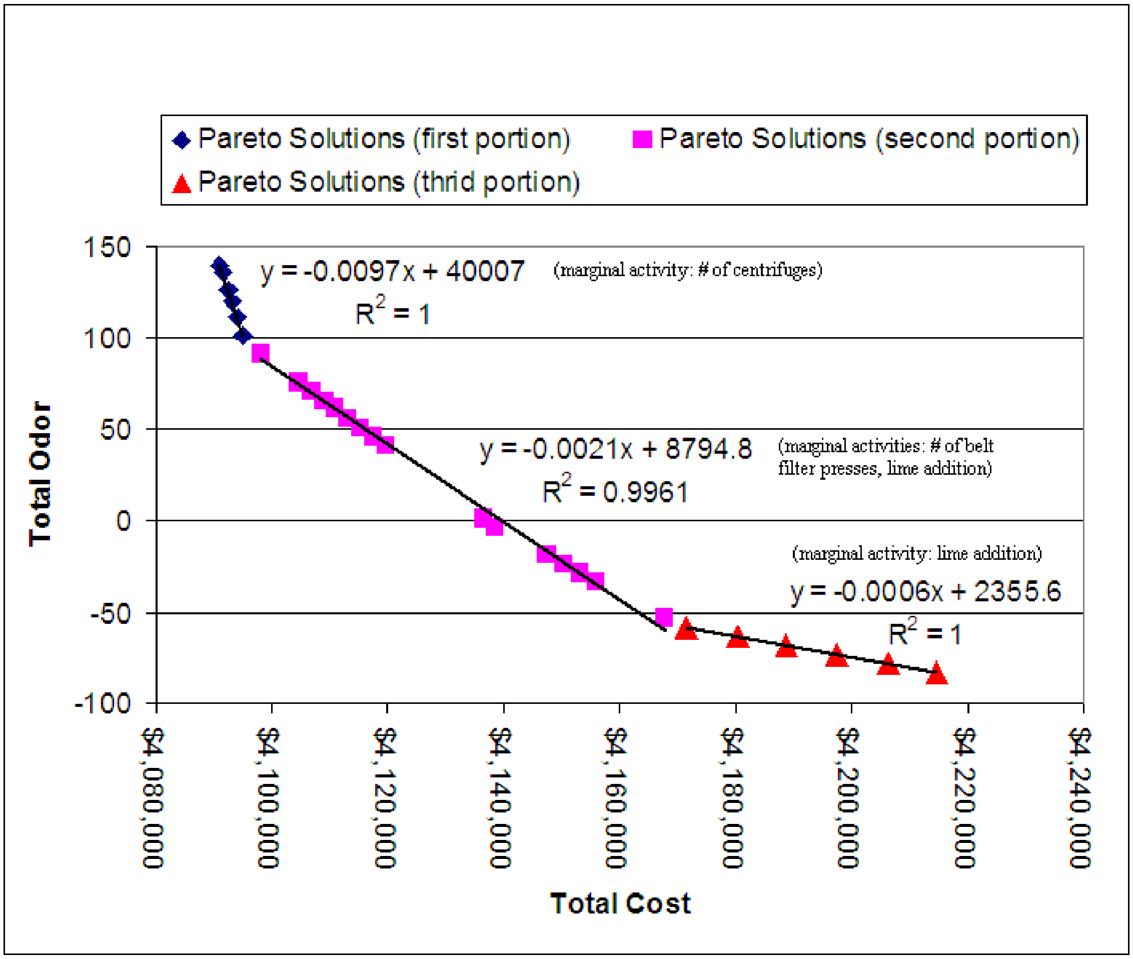


Figure 4.2: Base Case Pareto Optimal Solutions

Table 4.1: Base Case Pareto optimal points

Pareto point #	Total cost	Total odor	Total number of centrifuges	Total number of belt filter presses	Total amount of lime addition	Run time (sec)
1	4,091,075.66978	139.68400	40	0	7,750.00000	1,545
2	4,091,467.66978	135.86400	42	0	7,750.00000	941
3	4,092,447.66978	126.31400	47	0	7,750.00000	1,285
4	4,093,035.66978	120.58400	50	0	7,750.00000	1,412
5	4,094,015.66978	111.03400	55	0	7,750.00000	842
6	4,094,995.66978	101.48400	60	0	7,750.00000	1,243
7	4,097,987.66978	91.55400	62	13	7,750.00000	819
8	4,104,789.90960	75.57270	62	47	7,750.12962	826
9	4,106,789.90960	70.87270	62	57	7,750.12962	688
10	4,108,989.90960	65.70270	62	68	7,750.12962	155
11	4,110,787.66978	61.47400	62	77	7,750.00000	489
12	4,113,189.90960	55.83270	62	89	7,750.12962	844
13	4,115,389.90960	50.66270	62	100	7,750.12962	1,014
14	4,117,389.90960	45.96270	62	110	7,750.12962	589
15	4,119,589.90960	40.79270	62	121	7,750.12962	136
16	4,136,435.86242	1.64037	62	204	7,764.36300	68
17	4,138,387.66978	−3.38600	62	215	7,750.00000	45
18	4,147,416.66242	−18.35963	62	217	8,249.36300	72
19	4,150,288.08313	−23.35963	62	217	8,410.36300	65
20	4,153,070.16313	−28.35963	62	217	8,571.36300	66
21	4,155,825.74684	−33.39600	62	217	8,735.99983	43
22	4,167,741.96548	−53.49770	62	216	9,437.16989	888
23	4,171,625.94242	−58.35963	62	217	9,650.36300	46
24	4,180,265.94242	−63.35963	62	217	10,150.36300	46
25	4,188,905.94242	−68.35963	62	217	10,650.36300	47
26	4,197,545.94242	−73.35963	62	217	11,150.36300	46
27	4,206,185.94242	−78.35963	62	217	11,650.36300	40
28	4,214,830.83794	−83.35963	62	217	12,150.36300	41

were done. More specifically, these regression results were as follows.

$$\text{First portion: Total Odor} = -0.0097(\text{Total Cost}) + 40007$$

$$\text{Second portion: Total Odor} = -0.0021(\text{Total Cost}) + 8794.8$$

$$\text{Third portion: Total Odor} = -0.0006(\text{Total Cost}) + 2355.6$$

Each of these regressions had R^2 and adjusted R^2 values > 0.99 with statistically significant coefficients (i.e., t-statistics greater than 85 in absolute value). From these equations, to reduce the odor by one index point, one needed to pay on average $\frac{1}{0.0097} = \$103$, $\frac{1}{0.0021} = \$476$, and $\frac{1}{0.0006} = \$1667$, respectively for the first, second, and third portions of Pareto optimal points.

It is interesting to further analyze these Pareto optimal solutions. We begin our analysis by finding the ratios of the decrease in odor due to spending one dollar on lime, belt filter presses, and centrifuges. In addition, the dollars needed to spend on lime, belt filter presses, and centrifuges in order to reduce odor level by one index point are also determined. Table 4.2 shows the results from these calculations and the computational details follow. The ratio for the decrease in odor for one ton of biosolids due to dollars spent on lime was $\frac{0.01 \text{ points/(lb/DTS)}}{\$0.06/\text{lb}} = 0.17 \frac{\text{points}}{\$/\text{DTS}}$, obtained

Table 4.2: The decrease in odor level due to \$1 spending and dollars needed to reduce one odor index point

Variable	Decrease in odor level due to \$1 spending	\$ needed to reduce 1 odor index point
Lime	$0.17 \frac{\text{points}}{\$/\text{DTS}}$	$\frac{\$6}{\text{point}\times\text{DTS}}$
Belt filter presses	$2.35 \times 10^{-3} \frac{\text{point}}{\$}$	$\frac{\$425.53}{\text{point}}$
Centrifuges	$9.75 \times 10^{-3} \frac{\text{point}}{\$}$	$\frac{\$102.62}{\text{point}}$

by taking the coefficient for lime in the statistical model (0.01) and dividing it by the dollars per pound of lime (0.06). However, it is important to note that at the point where lime increases from less than 308 lbs/DTS to greater than or equal to this amount, another 1.13 points of odor decrease should be added. The reason was that the dummy variable would take on a value of zero in such a situation and hence the odor level would drop down for another 1.13 points (see Equation 3.2). It should be emphasized here that this is only a one time additional decrease (i.e. when lime addition changes from less than 308 lbs/DTS to greater than or equal to 308 lbs/DTS). Subsequently, when analyzing lime addition, one needs to go beyond just the total amount of lime addition in a time horizon. In other words, the daily detailed solutions for lime addition on each day may be needed. This analysis goes beyond the work done by Gabriel et al. [30], where this lime addition changing point was not discussed. Similar ratios for belt filter presses and centrifuges dollars were $\frac{0.47 \text{ points/belt filter press}}{\$200\text{/belt filter press}} = 2.35 \times 10^{-3} \ \frac{\text{points}}{\$}$ and $\frac{1.91 \text{ points/centrifuge}}{\$196\text{/centrifuge}} = 9.75 \times 10^{-3} \ \frac{\text{points}}{\$}$, respectively. It is important to note that the latter two ratios did not depend on the amount of biosolids being processed. This difference between these two ratios for belt filter presses and centrifuges versus lime corresponds directly to the units of variables themselves (e.g., lime: lbs/DTS).

Next, we find the dollars needed to spend on lime, belt filter presses, and centrifuges in order to reduce odor level by one index point. As for lime, one needed to spend $\frac{\$0.06\text{/lb}}{0.01 \text{ points/(lb/DTS)}} = \frac{\$6.00}{\text{points} \times \text{DTS}}$ in an attempt to reduce odor level of one ton of biosolids by one index point. But, again, one should take another 1.13 points decrease in odor level into account in the case where lime increases from less than 308 lbs/DTS to greater than or equal to 308 lbs/DTS. As for belt filter presses and centrifuges, the dollars spent to reduce odor level were not pertinent to the amount of biosolids being processed. In particular, the dollars needed to be spent on belt filter press and centrifuge were $\frac{\$200/\text{belt filter press}}{0.47\text{ point/belt filter press}} = \frac{\$425.53}{\text{point}}$, and $\frac{\$196/\text{centrifuge}}{1.91\text{ point/centrifuge}} = \frac{\$102.62}{\text{point}}$, respectively in order to reduce odor level by one index point. To this end, we may summarize that centrifuges were preferable to belt filter presses in reducing biosolids odor level. However, whether or not to prefer lime over belt filter presses or centrifuges or vice versa still depended on the amount of biosolids being processed and how far we needed to reduce odor level. The answers to this question will become clear and be revealed when we investigate a bit more deeply into the three portions of Pareto optimal points.

Before analyzing the three portions of these Pareto optimal points, we summarize the important findings for each portion. For the first portion, the marginal activity in reducing the biosolids odor level was the number of centrifuges in service. While the number of belt filter presses in service and lime addition were the marginal activity for the second portion. Lastly, lime additions were the marginal activity for the third portion. In addition, we also give some calculations on how much it would have cost if, in reducing odor levels, one were to employ activities (e.g., increase lime addition) different from those provided by the optimal solutions. Our calculation shows that a failure to understand key activities can cause up to an additional \$6,200 difference in total cost over a 30-day time horizon compared to the optimal solution. Next we scrutinize the first portion of Pareto optimal solutions.

For the first portion of Pareto optimal points, centrifuges were key to reduce biosolids odor level. We begin our analysis by comparing the solutions between the first two Pareto optimal points. We found that the only difference in optimal solutions between these two Pareto optimal points was the total number of centrifuges in service (confer Table 4.1). Then, we validate the optimal solutions by demonstrating how much it would have cost assuming that other variables (i.e., lime and belt filter presses) were key to reduce biosolids odor level.

The first Pareto optimal point had total cost = \$4091075.66978 and total odor = 139.68400 and the second point had total cost = \$4091467.66978 and total odor = 135.86400. By looking at the difference in cost between these two Pareto optimal points which was $\$4091467.66978 - \$4091075.66978 = \$392.00000$, we see that this number was exactly the cost of operating and maintaining two centrifuges ($2 \times \$196 = \$392$). Moreover, from (3.2), page 55, one centrifuge can reduce odor by 1.91 points and therefore two centrifuges can reduce odor level by $2 \times 1.91 = 3.82$ points, all else being equal. This number was exactly the decrease in odor level of $139.684 - 135.864 = 3.820$ point from the first to the second Pareto optimal points. In fact, the number two was the difference in the optimal solutions for the total numbers of centrifuges in service between the two Pareto optimal points which were 40 and 42, respectively for the first and second Pareto optimal points (see Table 4.1, page 107). In addition, this was the only difference in optimal solutions between these two Pareto optimal points.

Another interesting way of analyzing this solution was to see how much it would have cost, should one choose to employ either lime or belt filter press to achieve the decrease in odor level from the first to the second Pareto optimal point.

For the sake of discussion, Table 4.3 summarizes the calculations for the increases in total cost according to the non-optimal activities as well as the actual optimal solution. In particular, the first row represents optimal activity and the last three rows represent alternatives. As discussed earlier, in analyzing lime addition one needed to see whether or not lime increases from less than 308 lbs/DTS to greater than or equal to 308 lbs/DTS. First, we look at the case where lime increases from less than 308 lbs/DTS to greater than or equal to 308 lbs/DTS. According to Table 4.4 displaying details for Pareto optimal solutions 1 and 2, values for optimal lime additions were 250 lbs/DTS for both points. Suppose for the time being that the marginal activity was lime and in moving from the first to the second Pareto optimal point, the lime additions were to increase from 250 to 308 lbs/DTS for one day. The increase of $308 - 250 = 58$ lbs/DTS in lime additions for one day would have cost $58\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \frac{\text{DTS}}{\text{Day}} \times 90\% \times \frac{\$0.06}{\text{lb}} = \$1002.24$ and, subsequently, were to reduce $58\ \frac{\text{lbs}}{\text{DTS}} \times 0.01\ \frac{\text{points}}{\text{lbs/DTS}} + 1.13 = 1.71$ points of total odor level. Although it would only have reduced 1.71 points in odor level (3.82 points were needed), this cost figure

Table 4.3: Increases in total cost's calculations for moving from Pareto optimal points #1 to #2

Activity	Calculation	Total cost increases
Increasing 2 centrifuges	2 centrifuges × $\frac{\$196}{\text{centrifuge}}$	\$392
Increasing 58 $\frac{\text{lbs}}{\text{DTS}}$ of lime additions	$58\ \frac{\text{lbs}}{\text{DTS}} \times 320$ DTS $\times 90\% \times \frac{\$0.06}{\text{lb}}$	\$1002.24
Increasing 382 $\frac{\text{lbs}}{\text{DTS}}$ of lime additions	$382\ \frac{\text{lbs}}{\text{DTS}} \times 320$ DTS $\times 90\% \times \frac{\$0.06}{\text{lb}}$	\$6,600.96
Increasing 9 belt filter presses	9 belt filter presses × $\frac{\$200}{\text{belt filter press}}$	\$1800

Table 4.4: Pareto optimal points 1 and 2's detailed solutions

Day	Pareto optimal point number 1		Pareto optimal point number 2	
	C_d^k	L_d^{dc}	C_d^k	L_d^{dc}
1	1	250.00	1	250.00
2	1	250.00	2	250.00
3	1	250.00	1	250.00
4	1	250.00	1	250.00
5	1	250.00	1	250.00
6	1	250.00	1	250.00
7	2	250.00	2	250.00
8	2	250.00	1	250.00
9	1	250.00	1	250.00
10	1	250.00	1	250.00
11	1	250.00	1	250.00
12	1	250.00	1	250.00
13	1	250.00	1	250.00
14	1	250.00	1	250.00
15	1	250.00	1	250.00
16	1	250.00	1	250.00
17	1	250.00	1	250.00
18	1	250.00	1	250.00
19	1	250.00	1	250.00
20	2	250.00	2	250.00
21	1	250.00	1	250.00
22	2	250.00	2	250.00
23	1	250.00	2	250.00
24	1	250.00	2	250.00
25	2	250.00	2	250.00
26	2	250.00	1	250.00
27	2	250.00	1	250.00
28	2	250.00	2	250.00
29	1	250.00	2	250.00
30	2	250.00	2	250.00
31	1	250.00	2	250.00
Total	40	7750.00	42	7750.00

would already have surpassed the optimal increase of the total cost ($392).

Next, we analyze the case if the lime were to increase to a value less than 308 lbs/DTS. All things being equal, the decrease of 3.82 points in total odor level required $\frac{3.82 \text{ points}}{0.01 \text{ point/(lb/DTS)}} = 382.00$ lbs/DTS of total lime addition, which cost $382.00 \frac{\text{lbs}}{\text{DTS}} \times 320 \text{ DTS} \times 90\% \times \frac{\$0.06}{\text{lb}} = \$6,600.96$. On the other hand, one would have needed to spend $\frac{\$425.53}{\text{point}} \times 3.82 \text{ points} = \$1,625.52$ if belt filter presses were the only key parameter in question. Put differently, we would have needed $\frac{3.82 \text{ points}}{0.47 \text{ point/belt filter press}} = 8.13$ belt filter presses and it would have cost 8.13 belt filter presses×$200/belt filter press $= \$1,625.53$ consistent with earlier calculations. However, practically, 8.13 belt filter presses are not applicable and therefore at least 9 belt filter presses would have needed and cost 9 belt filter presses × $200/belt filter press = $1800. This was further evidence validating the optimal solutions. Moreover, by moving down to the last Pareto optimal point of the first portion of Pareto optimal points, similar observations can be obtained when comparing successive Pareto optimal points. Next, we examine the second portion of Pareto optimal points.

First, we studied the decrease in odor from the last Pareto optimal point (Pareto optimal point six) of the first portion (total cost = $4,094,995.66978 and total odor = 101.484) to the first Pareto optimal point (Pareto optimal point seven) of the second portion (total cost = $4,097,987.66978 and total odor = 91.554). We can see that, in moving from point six to point seven, there were two more centrifuges in service yielding 62 centrifuges in total which was the highest possible total number of centrifuges in service within the 31 days time horizon. In addition, compared to no belt filter presses for point six, there were 13 belt filter presses in service for point seven. As for lime, there were no changes in the amount of lime addition. The aforementioned solutions can be explained as follows. If one were to choose lime for reducing the odor level, $\frac{9.93 \text{ points}}{0.01 \text{ point/(lb/DTS)}} = 993.00$ lbs/DTS would have been needed. Therefore, it would have cost $993.00 \frac{\text{lbs}}{\text{DTS}} \times 320 \text{ DTS} \times 90\% \times \frac{\$0.06}{\text{lb}} = \$17,159.04$, which would have been way beyond the optimal increase in total cost. Next, centrifuges and belt filter presses should be considered. As discussed earlier, centrifuges dominated belt filter presses in reducing biosolids odor level, therefore centrifuges were utilized to their upper limits ($2 \frac{\text{centrifuges}}{\text{day}} \times 31$ days $= 62$ centrifuges in 31 days) before 13 belt filter presses were used. In fact, we can see that the decrease of $101.484 - 91.554 = 9.930$ points in odor level was from two centrifuges at 1.91 points per one centrifuge and 13 belt filter presses at 0.47 points per one belt filter press, that is $2 \times 1.91 + 13 \times 0.47 = 9.93$. Next, we analyze the decreases in odor level from one Pareto optimal point to another within the second portion of the Pareto optimal points.

All Pareto optimal solutions in the second portion have the same total number of centrifuges. In fact, for every Pareto optimal point, starting from the first Pareto optimal point of the second portion to the last Pareto optimal point of the third portion, centrifuges were utilized to their upper limit of 62. Therefore, what made the decreases in odor level from one Pareto optimal point to another were either additional use of belt filter presses or lime or both. In other words, belt filter presses and lime started to play roles in bringing down the odor level. Now, let us analyze the decrease in odor level from the first (Pareto optimal point #7) to the second

(Pareto optimal point #8) Pareto optimal point of this second portion. According to Table 4.1, there was a decrease of $91.55400 - 75.57270 = 15.98130$ points in total odor level and an increase of $\$4,097,987.66978 - \$4,104,789.90960 = \$6,802.23982$ in total cost. The total number of belt filter presses increased by $47 - 13 = 34$ units. In addition, the total amount of lime addition changed from 7,750.00000 lbs/DTS to 7,750.12962 lbs/DTS or an increase of 0.12962 lbs/DTS. Table 4.5 displays the calculations for the increases in total odor according to the optimal activities and comparative non-optimal activities. More specifically, row 1 shows the calculation for optimal activities which were adding 34 belt filter presses and 0.12962 lbs/DTS of lime additions and row 2 has 35 belt filter presses

Table 4.5: Increases in total cost's calculations for moving from Pareto optimal points #7 to #8

Activity	Calculation	Total cost increase
Increasing 34 belt filter presses & 0.12962 lbs/DTS of lime additions	$\frac{\$0.06}{\text{lb}} \times 0.12962\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\%$ $+\ 34\ \text{belt filter presses} \times \frac{\$200}{\text{belt filter presses}}$	\$6802.23983
Increasing 35 belt filter presses	$\frac{\$200}{\text{belt filter presses}} \times 35\ \text{belt filter presses}$	\$7000

First, let us analyze the decrease in odor level effected by belt filter presses only. The increase of 34 belt filter presses reduced odor by $\frac{0.47\ \text{points}}{\text{belt filter press}} \times 34$ belt filter press $= 15.98$ points. Subsequently, $15.98130 - 15.98 = 0.00130$ points decrease in odor level was still needed. The remaining reduction could come from either an additional belt filter press or more lime. Bringing one more belt filter press would have cost \$200 and resulted in 0.47 points decrease in odor level. On the other hand, 0.12962 lbs/DTS increases in total lime addition could reduce odor level by $0.12962\ \frac{\text{lbs}}{\text{DTS}} \times 0.01\ \frac{\text{points}}{\text{lbs/DTS}} = 0.00130$ points and cost only $\frac{\$0.06}{\text{lb}} \times 0.12962\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\%$ flow to DCWASA = \$2.23983. It is important to note here that, according to Table 4.6 displaying detailed solutions of Pareto optimal points 7 and 8, there was not any single day where lime increased from less than 308 lbs/DTS to greater than or equal to 308 lbs/DTS and, hence, the additional 1.13 point-decrease in odor level was not applicable. Next, adding the lime cost of \$2.23983 to the cost of 34 belt filter presses resulted in $\$2.23983 + 34 \times \$200 = \$6802.23983 \approx \$6802.23982$ increase, coinciding with the increase in total cost calculated earlier. So far, we have shown the situation where lime was preferable to belt filter presses in bringing down the odor level for the last small decrement. Next, we investigate another revealing situation where, when moving from one Pareto optimal point to the next, the total number of belt filter presses increased while the total amount of lime addition decreased.

Such a situation was found when moving from Pareto optimal points number 10 (total cost = $\$4,108,989.90960$, total odor = 65.70270) to number 11 (total cost = \$4,110,787.66978, total odor = 61.47400). Therefore, the increase in total cost was $\$4,110,787.66978 - \$4,108,989.90960 = \$1,797.76018$ and the decrease in total odor was $65.70270 - 61.47400 = 4.22870$. In addition, the changes in the total number of belt filter presses and total amount of lime addition were 68 to 77 belt filter presses and 7750.12962 to 7750.00000 lbs/DTS, respectively. Table 4.7 shows the optimal activities (first row) and non-optimal activities (second row) along with their

Table 4.6: Pareto optimal point numbers 7 and 8's detailed solutions

Day	Pareto optimal point number 7		Pareto optimal point number 8	
	G_d^k	L_d^{dc}	G_d^k	L_d^{dc}
1	0	250.00	0	250.00
2	0	250.00	0	250.00
3	0	250.00	0	250.00
4	0	250.00	0	250.00
5	0	250.00	0	250.00
6	0	250.00	0	250.00
7	0	250.00	0	250.00
8	0	250.00	0	250.00
9	0	250.00	0	250.00
10	1	250.00	6	250.00
11	0	250.00	5	250.00
12	0	250.00	3	250.00
13	1	250.00	2	250.00
14	0	250.00	0	250.00
15	0	250.00	0	250.00
16	0	250.00	5	250.00
17	0	250.00	5	250.12962
18	0	250.00	5	250.00
19	0	250.00	0	250.00
20	0	250.00	2	250.00
21	0	250.00	0	250.00
22	0	250.00	1	250.00
23	2	250.00	0	250.00
24	0	250.00	0	250.00
25	7	250.00	7	250.00
26	0	250.00	0	250.00
27	0	250.00	2	250.00
28	0	250.00	0	250.00
29	0	250.00	2	250.00
30	0	250.00	0	250.00
31	2	250.00	2	250.00
Total	13	7,750.00	47	7,750.12962

corresponding calculations for increases in total costs. First, we analyze what would

Table 4.7: Increases in total cost's calculations for moving from Pareto optimal points #10 to #11

Activity	Calculation	Total cost increases
Increasing 9 belt filter presses & decreasing 0.12962 $\frac{\text{lbs}}{\text{DTS}}$ of lime additions	9 belt filter presses $\times \frac{\$200}{\text{belt filter presses}}$ $-0.12962 \frac{\text{lbs}}{\text{DTS}} \times 320$ DTS $\times 90\% \times \frac{\$0.06}{\text{lb}}$	\$1797.76017
Increasing 8 belt filter presses & 46.87 $\frac{\text{lbs}}{\text{DTS}}$ of lime addition	8 belt filter presses $\times \frac{\$200}{\text{belt filter presses}}$ $+46.87 \frac{\text{lbs}}{\text{DTS}} \times 320$ DTS $\times 90\% \times \frac{\$0.06}{\text{lb}}$	\$2,400.91

have happened if, in moving from Pareto optimal points number 10 to number 11, the total number of belt filter presses were to increase by one less than the optimal increase and the total amount of lime addition were to increase instead. Hence, we assumed that there were eight more belt filter presses as oppose to the nine more indicated by the optimal solution. Under this assumption, the total odor would have decreased by 0.47 $\frac{\text{point}}{\text{belt filter press}} \times 8$ belt filter presses $= 3.76$ points. In other words, the total odor level could be brought down to $65.70270 - 3.76 = 61.94270$ and therefore $61.94270 - 61.47400 = 0.46870$ points decrease in odor level would still have been needed to achieve the total odor in question. The decrease of 0.46870 points in total odor level required $\frac{0.46870 \text{ point}}{0.01 \text{ point/(lb/DTS)}} = 46.87000$ lbs/DTS of total lime addition, which cost $46.87000 \frac{\text{lbs}}{\text{DTS}} \times 320$ DTS $\times 90\% \times \frac{\$0.06}{\text{lb}} = \$800.91360$. On the other hand, according to the optimal solution, when one belt filter press was brought in, it cost only \$200 more yet reduced the total odor by 0.47 points. However, the decrease of 0.47 points was more than the 0.4687 points needed by $0.47 - 0.4687 = 0.0013$ points. Hence, we could reduce the amount of lime addition and subsequently cut down the total cost. The amount of lime addition that can be reduced can be calculated as $\frac{0.0013}{0.01} = 0.13000$ lbs/DTS. This number was approximately the reduction of total lime addition of $7750.12962 - 7750 = 0.12962$ lbs/DTS reported by the optimal solution. To this end, the increase in the total cost of $\$4,110,787.66978 - \$4,108,989.90960 = \$1,797.76018$ can be calculated from the cost of nine belt filter presses less the cost of 0.12962 lbs/DTS of total lime addition. More specifically, $9 \times 200 - 0.12962 \times 320 \times 0.9 \times 0.06 = \$1797.76017 \approx \$1797.76018$.

Next, we investigated another behavior of Pareto optimal solutions where, in moving from one Pareto optimal point to another, the total number of belt filter presses decreased and total lime addition increased. It is important to emphasize that, in explaining the transition between the next two Pareto optimal points in question, the detailed daily solutions play a key role.

The next pair of Pareto optimal points to be analyzed were Pareto optimal points number 21 (total cost = \$4,155,825.74684, total odor = −33.39600) and number 22 (total cost = \$4,167,741.96548, total odor = −53.49770). The total numbers of belt filter presses were 217 and 216 while the the total amounts of lime addition were 8,735.99983 and 9,437.16989 lbs/DTS, respectively for Pareto optimal point numbers 21 and 22. From Table 4.1, we can see that there were a decrease of $217 - 216 = 1$ belt filter press and an increase of $9,437.16989 - 8,735.99983 = 701.17006$ lbs/DTS of the total amount of lime additions. An interesting question is why was there a reduction of one belt filter press? Since, at Pareto optimal point number 21, belt filter presses were utilized to their upper limit of 217 (7×31) in 31 days, why couldn't the next Pareto optimal point achieve the desired total odor level by just continuously increasing the amount of lime addition until the desired total odor level was met? Table 4.8 displays a detailed account of the daily solutions of the two Pareto optimal points in question. It is important to note that only the solutions for the number of belt filter presses in service and the amount of lime additions are shown, as they were the only variables under consideration. According to Table 4.8, there were 13 days where the amount of lime additions for Pareto optimal point number 22 were greater than those of Pareto optimal point number 21. As for belt filter presses in service, there was only one difference on day 20, where the total numbers of belt filter presses in service were 7 and 6, respectively for Pareto optimal point numbers 21 and 22. As we have raised the question earlier that why was there a reduction of one belt filter press? Or why couldn't we continuously increase the amount of lime addition until the desired odor level could be achieved? Now we propose alternative combination of days and lime additions to achieve the same result without a reduction of one belt filter press. Then, we will compare the total cost incurred from the conjectured combination with the total cost incurred from the model's optimal solution. Table 4.9 displays the optimal activities (first row) and non-optimal activities (second row). The increases in total costs are also provided. Suppose for the time being, we assume that there was no change in the total number of belt filter presses and therefore only lime were to be a key role. First, we examine the decrease in the odor level and the increase in the total cost, further assuming that there was no change in lime additions on day

Table 4.8: Pareto optimal point numbers 21 and 22's detailed solutions

Day	Pareto optimal point number 21		Pareto optimal point number 22	
	G_d^k	L_d^{dc}	G_d^k	L_d^{dc}
1	7	250.00	7	308.00
2	7	250.00	7	308.00
3	7	250.00	7	308.00
4	7	308.00	7	308.00
5	7	308.00	7	308.00
6	7	308.00	7	308.00
7	7	250.00	7	308.00
8	7	250.00	7	308.00
9	7	250.00	7	308.00
10	7	250.00	7	308.00
11	7	250.00	7	308.00
12	7	308.00	7	308.00
13	7	250.00	7	250.00
14	7	308.00	7	308.00
15	7	250.00	7	308.00
16	7	308.00	7	308.00
17	7	308.00	7	308.00
18	7	250.00	7	250.00
19	7	308.00	7	308.00
20	7	250.00	**6**	308.00
21	7	308.00	7	308.00
22	7	308.00	7	308.00
23	7	308.00	7	308.00
24	7	250.00	7	308.00
25	7	308.00	7	308.00
26	7	308.00	7	308.00
27	7	250.00	7	308.00
28	7	308.00	7	308.00
29	7	308.00	7	308.00
30	7	308.00	7	308.00
31	7	308.00	7	**313.17**
Total	217	8,736.00	216	9,437.17

Table 4.9: Increases in total cost's calculations for moving from Pareto optimal points #21 to #22

Activity	Calculation	Total cost increases
Increasing lime additions from 250 to 308 $\frac{\text{lbs}}{\text{DTS}}$ for 12 days & from 308 to 313.17 $\frac{\text{lbs}}{\text{DTS}}$ for 1 days & decreasing one belt filter presses	$12 \times 58\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\% \times \frac{\$0.06}{\text{lbs}}$ $+1 \times 5.17\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\% \times \frac{\$0.06}{\text{lbs}}$ $-1\ \text{belt filter press} \times \frac{\$200}{\text{belt filter presses}}$	\$11,916.22

Increasing lime additions
from 250 to 308 $\frac{\text{lbs}}{\text{DTS}}$
for 11 days
& from 308 to 313.17 $\frac{\text{lbs}}{\text{DTS}}$
for 1 days
& from 250 to 300 $\frac{\text{lbs}}{\text{DTS}}$
for 2 days
& from 250 to 274 $\frac{\text{lbs}}{\text{DTS}}$
for 1 days
	$11 \times 58\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\% \times \frac{\$0.06}{\text{lbs}}$
	$1 \times 5.17\frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\% \times \frac{\$0.06}{\text{lbs}}$
	$2 \times 50\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\% \times \frac{\$0.06}{\text{lbs}}$
	$1 \times 24\ \frac{\text{lbs}}{\text{DTS}} \times 320\ \text{DTS} \times 90\% \times \frac{\$0.06}{\text{lbs}}$
		\$13,256.70

20. Accordingly, there would have been 11 days where lime additions increased from 250 to 308 lbs/DTS and one day from 308 to 313.17 lbs/DTS. Thus, there would have been a decrease in odor levels of $11 \times (0.01\ \frac{\text{points}}{\text{lbs/DTS}} \times (308 - 250)\ \text{lbs/DTS} + 1.13\ \text{points}) + 0.01\ \frac{\text{points}}{\text{lbs/DTS}} \times (313.17 - 308)\ \text{lbs/DTS} = 18.8617$ points. Since the decrease in the odor level of $53.4977 - 33.396 = 20.1017$ points was needed, another $20.1017 - 17.1517 = 1.24$ points in odor reduction would still have been needed. There were several combinations of days and lime additions to achieve an additional 1.24-point reduction and one way to achieve this were to increase lime additions on days 13, 18, and 20 to 300, 300, and 274 lbs/DTS, respectively. Then, the decrease in odor level would have been $0.01\ \frac{\text{points}}{\text{lbs/DTS}} \times (300 - 250)\ \frac{\text{lbs}}{\text{DTS}} + 0.01\ \frac{\text{points}}{\text{lbs/DTS}} \times (300 - 250)\ \frac{\text{lbs}}{\text{DTS}} + 0.01\ \frac{\text{points}}{\text{lbs/DTS}} \times (274 - 250)\ \frac{\text{lbs}}{\text{DTS}} = 1.24$ points. Moreover, the increase in total cost would have been $90\% \times 320\ \text{DTS} \times (300 - 250)\ \frac{\text{lbs}}{\text{DTS}} \times \frac{\$0.06}{\text{lbs}} + 90\% \times 320\ \text{DTS} \times (300 - 250)\ \frac{\text{lbs}}{\text{DTS}} \times \frac{\$0.06}{\text{lbs}} + 90\% \times 320\ \text{DTS} \times (274 - 250)\ \frac{\text{lbs}}{\text{DTS}} \times \frac{\$0.06}{\text{lbs}} = \$2,142.72$. It is important to emphasize that this cost figure would have been incurred from the increase in lime addition only and there would have been no change in the total number of belt filter presses in service. Next, we compare this cost figure with the cost incurred from the model's optimal solution. The model achieved the reduction of 1.24 points in odor level by first increasing the lime addition on day 20 from 250 to 308 lbs/DTS. This resulted in a decrease in odor level of $0.01\ \frac{\text{points}}{\text{lbs/DTS}} \times (308 - 250)\ \frac{\text{lbs}}{\text{DTS}} + 1.13\ \text{points} = 1.71$ points. Obviously, this was more than the required decrease in odor level of 1.24 points. Second, the model offset the excess decrease by reducing one belt filter press in service, which in tern incurred another 0.47 points of odor level. Hence, altogether there

was a decrease of $1.71 - 0.47 = 1.24$ points in odor level and the associated cost was $90\% \times 320$ DTS $\times (308 - 250) \frac{\text{lbs}}{\text{DTS}} \times \frac{\$0.06}{\text{lbs}} - 1 \times \$200 = \$802.24$, significantly less than the \$2142.72 figure determined earlier. Thus, the previous calculations were consistent with the Pareto optimal solutions. To this end, we determined the overall increases in cost and the overall decreases in odor incurred by the optimal solutions. The increases in total cost comprised the increments of lime additions of $308 - 250 = 58$ lbs/DTS for 12 days and of $313.17 - 308 = 5.17$ lbs/DTS for 1 day and the decrement of one belt filter press for 1 day. In particular, the increases in total cost were $12 \times 58 \frac{\text{lbs}}{\text{DTS}} \times 320$ DTS $\times 90\% \times \frac{\$0.06}{\text{lbs}} + 1 \times 5.17 \frac{\text{lbs}}{\text{DTS}} \times 320$ DTS $\times 90\% \times \frac{\$0.06}{\text{lbs}} - \$200 = \$11,916.22$, coinciding with the optimal decreases in the total cost when moving from Pareto optimal point numbers 21 to 22. Next, we analyze the third portion of the Pareto optimal solutions.

All Pareto optimal points in the third portion had centrifuges and belt filter presses utilized at their full capacities. In particular, there were the total of 62 centrifuges and 217 belt filter presses in services for each of the Pareto optimal points in the third portion. The only differences in solutions among these Pareto optimal points were the amounts of lime added. In other words, lime was the marginal odor reduction activity. This observation is borne out in comparing points number 27 and number 28. For these points, the reduction in odor levels was five points $(-78.35963 - (-83.35963))$ with a corresponding increase in total lime additions of $\frac{5}{0.01} = 500$ lbs/DTS as shown in Table 4.1.

So far, we have analyzed the effects and roles of all management decision variables in the wastewater treatment process according to optimal solutions. It can be seen that at various stages and/or situations at hand, one variable may be a critical factor in effectively reducing total odor level while the others may not. Failure to understand these situations may result in a significantly higher increase in total cost. For example, in reducing the total odor from Pareto optimal point numbers 1 to 2, one could have paid $6,600.96 by using lime to reduce odor level, while it cost only $392 for having two more centrifuges in service in order to achieve the same amount of odor reduction. Having analyzed the wastewater treatment process's variables, we next analyze the biosolids shipment variables, F_{dcf}.

For the Base Case, biosolids' hauling costs were charged on a tonnage basis regardless of to which reuse field they hauled biosolids (see hauling cost details in Section 3.3.5). It is also important to note that the shipment pattern was also constrained by (4.4), where biosolids assigned to each contractor were limited at pre-specified values. An analysis of the optimal solutions revealed that first, biosolids were assigned to the contractor whose hauling cost was cheapest. After the daily limit for that contractor was met, biosolids were next assigned to the second cheapest one and then to the third one after the second contractor's capacity was reached. In fact, it is shown that contractor 2 was never chosen because it had the most expensive hauling costs. In addition, the daily hauling capacities for contractors 1 and 3 altogether already covered the daily biosolids production.

In choosing the reuse fields for each Pareto optimal point, since there was no penalty of going to fields that were further away, the model picked those fields that satisfied the odor threshold limit constraint (see constraint 4.7), regardless of distances from Blue Plains. In other words, there were multiple solutions for F_{dcf}.

We checked this by fixing values of some F_{dcf} whose solutions were not zero to zeros and reran the model with the same set of constraints. It turned out that the model selected other fields with the same objective function value (total cost). Hence, there were solutions other than the originally reported ones.

Nevertheless, comparing among Pareto optimal points, we found an interesting relationship between the total odors (as well as total costs) and their associated selected reuse fields' average odor thresholds. As indicated in Figures 4.3 and 4.4, when Pareto optimal total odors decrease from 139.68 to 75.57 (or equivalently, Pareto optimal total costs increase from \$4,091,075.67 to \$4,104,789.91), the selected reuse fields' average odor thresholds decrease from 5.36 to 4.63. For the remaining Pareto optimal points, the average odor thresholds remains at about the same level with a mean value of 4.65. There was one exception when total odor equals -78.36 (total cost equals \$4,206,185.94) where average odor threshold equals 4.95. The above relationships can be explained as follows. When total odors (summed over 31 days) were high, the daily average odor was also high. Consequently, on average, biosolids must be hauled to reuse fields with higher odor thresholds. Therefore, the selected reuse fields' average odor thresholds were also high. On the other hand, when total odors (as well as daily average odors) were low, biosolids could be hauled to reuse fields with lower corresponding odor thresholds. Nevertheless, these biosolids with lower odors could also be hauled to reuse fields with higher corresponding odor thresholds as we see when total odor equals -78.36, as the one exception, we mentioned above. Figure 4.5 shows selected reused fields taken from two Pareto optimal points when total odors equal 75.5727 and 139.684. In particular, to show

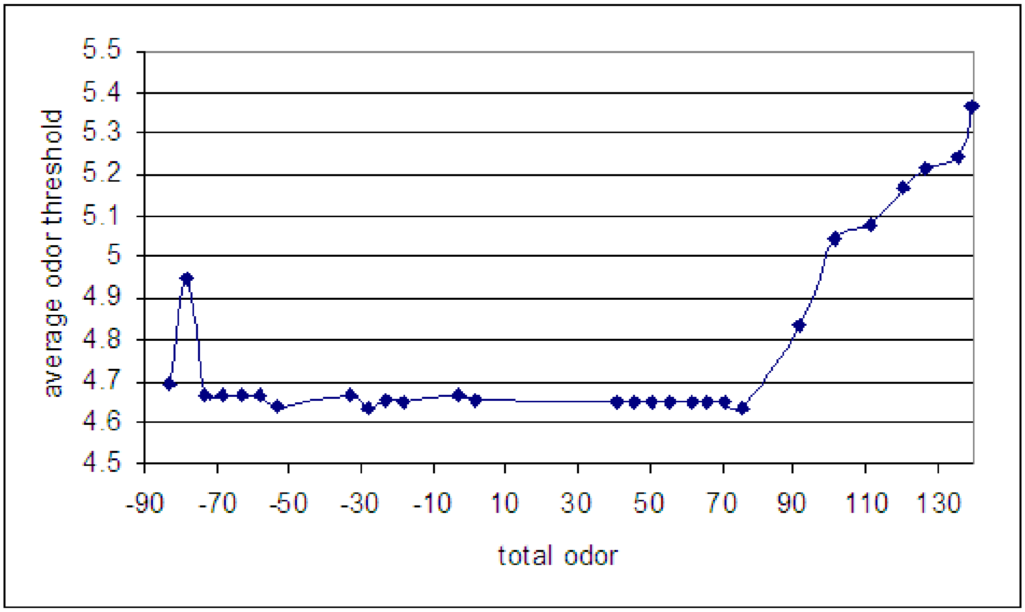


Figure 4.3: Pareto optimal total odors versus selected reuse fields' average odor thresholds

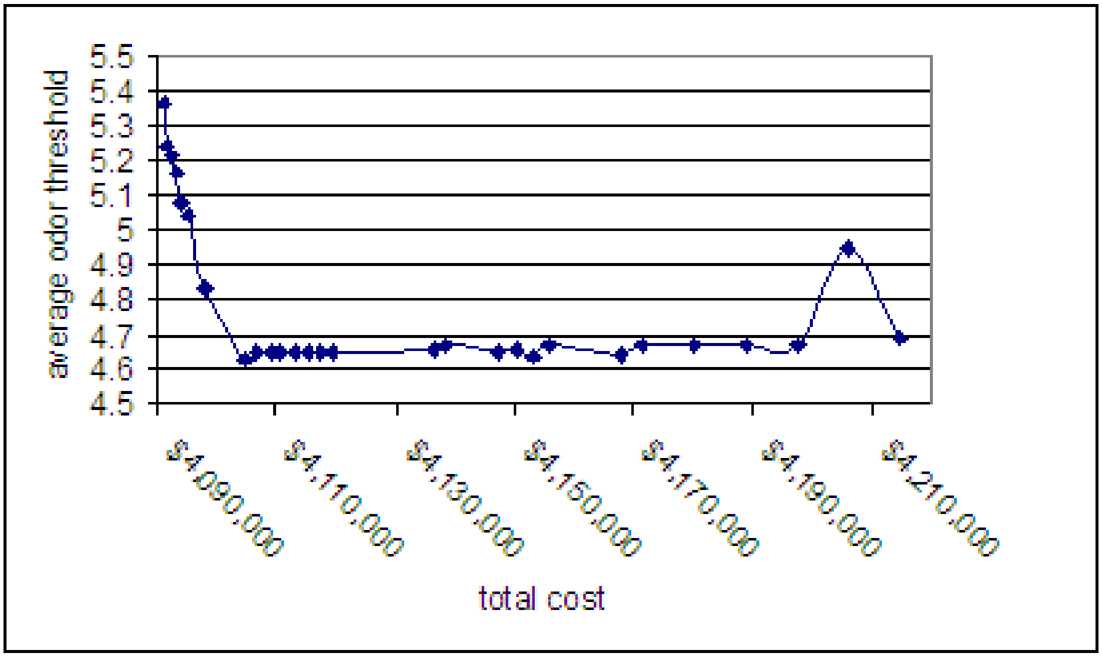


Figure 4.4: Pareto optimal total costs versus selected reuse fields' average odor thresholds

the differences in odor thresholds associated with selected reuse fields corresponding to each Pareto optimal point, only reuse fields selected exclusively by either one of the Pareto optimal points are displayed. In other words, reuse fields selected by both Pareto optimal points are not shown in the figure. It is indicated that odor thresholds associated with reuse fields selected by the Pareto optimal point when total odor equals 57.5727 were generally lower than odor thresholds associated with reuse fields selected by the Pareto optimal point when total odor equals 139.684.

To conclude the analysis of the Base Case, we compare the Pareto optimal total costs and total odors with their counterparts obtained by fixing values of all decision variables to the values implemented by DCWASA or the on-site contractor during the time period in question, however, with some exceptions. Although we were able to fix values of lime additions and the numbers of the on-site contractor belt filter presses and centrifuges to actual ones implemented by DCWASA or the on-site contractor, we were not able to fix the amount of wet ton solids hauled to each reuse field due to the following reasons. Also, some of those visited reuse fields without latitude and longitude data were not included in our optimization model as previously discussed. Therefore, biosolids applied to those excluded reuse fields were not taken into account. Subsequently, the total wet tonnage determined from actual application (excluding from those visited reuse fields with incomplete data) was significantly less than the total wet ton obtained from the optimization model. Thus, we were able only to fix what reuse fields to which biosolids should be hauled but not the amount of biosolids to be applied. Finally, we reran the Base Case (problem 4.29) with the following additional constraints and for the sake of

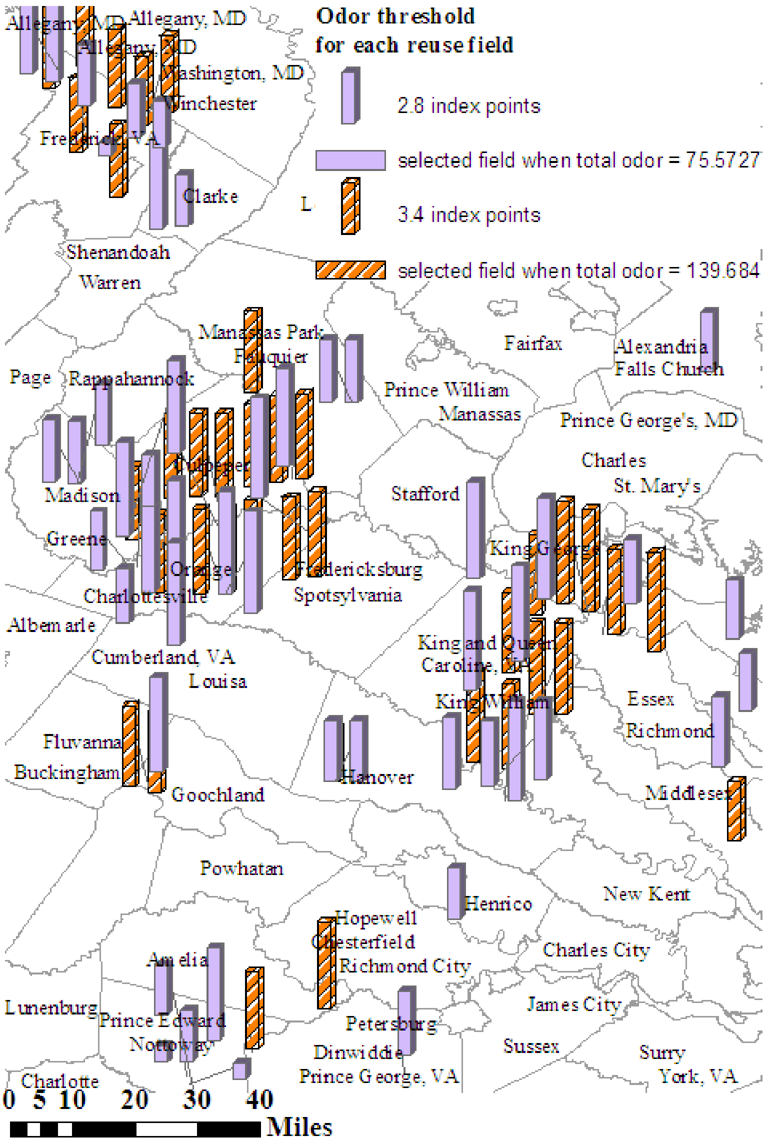


Figure 4.5: Optimal reuse fields when total odor scores equal 75.5727 and 139.684

discussion, we call this modified Base Case “DCWASA Case”.

$\forall d \in D;$

$$C_d^k = C_d^{ka} \tag{4.36}$$

$$G_d^k = G_d^{ka} \tag{4.37}$$

$$L_d^{dc} = L_d^{dca} \tag{4.38}$$

$\forall f \in F^a;$

$$\sum_{d=1}^{|D|} \sum_{c=1}^{|C|} \delta_{dcf} >= 1 \tag{4.39}$$

where

C_d^{ka} = Recorded number of the on-site contractor centrifuges in service on day d

G_d^{ka} = Recorded number of the on-site contractor belt filter presses in service on day d

L_d^{dca} = Recorded DCWASA lime additions on day d

F^a = Set of visited reuse fields during the time period D

Equations (4.36), (4.37), and (4.38) enforce variables C_d^k, G_d^k, and L_d^{dc} to take their corresponding recorded values C_d^{ka}, G_d^{ka}, and L_d^{dca}, respectively. While, (4.39) enforces at least one of δ_{dcf} (note that $\delta_{dcf} \in \{0, 1\}$; see equation (4.8)) must be one if $f \in F^a$. Consequently, according to (4.5), F_{dcf} must be greater than or equal to 23 and this satisfies our restriction that biosolids must be hauled to reuse field $f \in F^a$.

The optimal solution from the DCWASA Case indicates a total cost of $\$4,131,458.98617$ and a total odor score of 76.13069. Figure 4.6 depicts the Pareto optimal points from the Base Case and the DCWASA Case. According to this

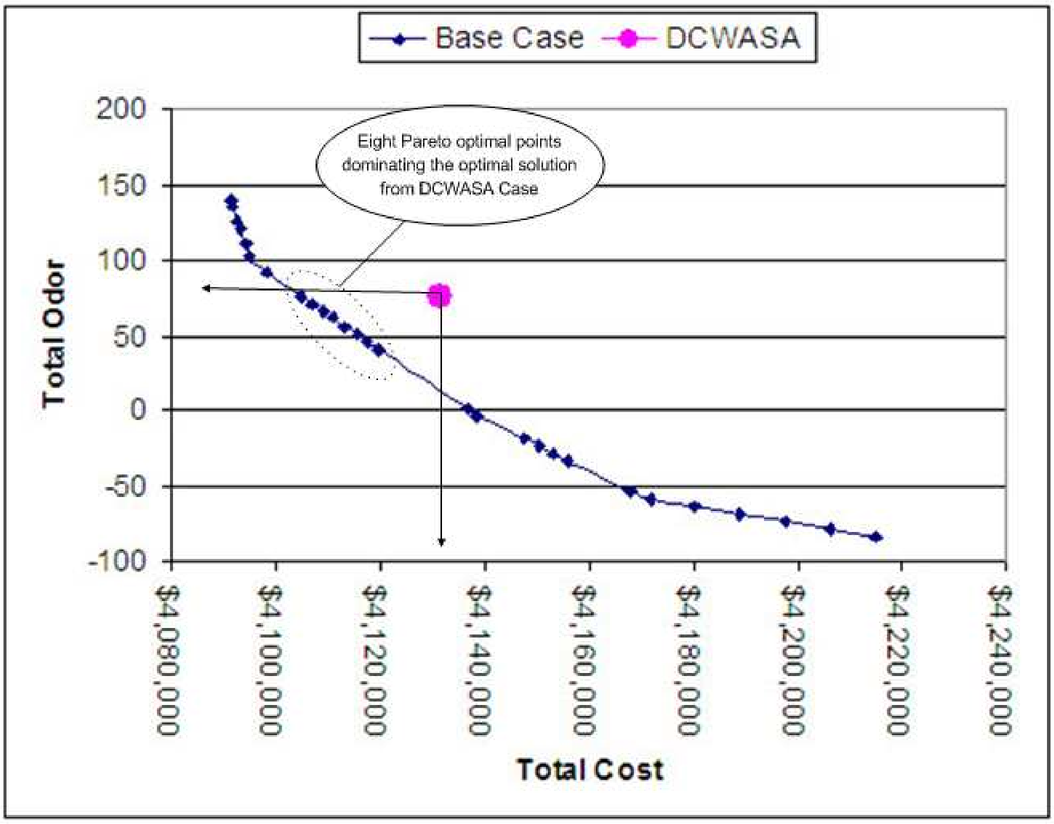


Figure 4.6: Pareto optimal points from Base Case and DCWASA Case

figure, there are eight Pareto optimal points from the Base Case dominating the optimal solution from the DCWASA Case. From Table 4.1, these dominating Pareto optimal points are points number eight through 15 with the total costs range from $4,104,789.90960 to $4,119,589.90960 and total odors range from 75.57270 to 40.79270. It can be seen that total odors and total costs from these eight Pareto optimal points are lower than optimal total odor and total cost obtained from DCWASA Case. Hence, one should prefer employing the solutions from these eight Pareto optimal points to employing the solution from the DCWASA Case and this validates the use of our optimization model. Table 4.10 shows the details for the optimal solution from the Pareto optimal point number eight (total cost = $4,104,789.9096, total odor score = 75.5727) and the optimal solution when fixing all variables according to DCWASA or the on-site contractor actual implementation during the time period in question. Figures 4.7, 4.8, and 4.9 show the optimal solution for Pareto optimal point eight and the actual implemented values for lime additions and the numbers of the on-site contractor centrifuges and belt filter presses, respectively. According to Table 4.10 and Figure 4.7, the actual daily lime additions fluctuated between 251.17 lbs/DTS and 294.90 lbs/DTS with an average of 270.82 lbs/DTS. While the optimal solution corresponding to Pareto optimal point number eight indicates most daily lime additions of 250 lbs/DTS with an exception of 250.13 lbs/DTS on day 17. It can be seen that the optimal solution for lime additions significantly differs from the actual implemented values. Similar observations can also be found for the total numbers of centrifuges and belt filter presses. To

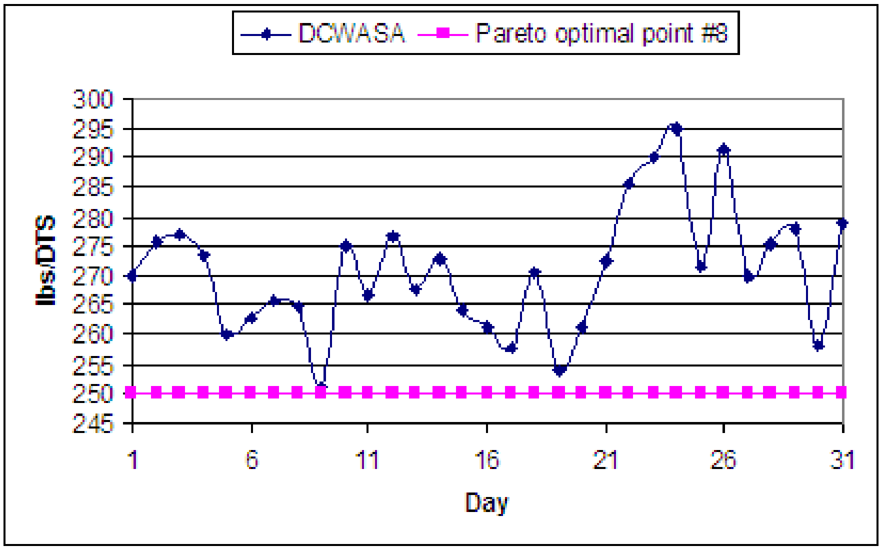


Figure 4.7: Daily solutions for lime additions

Table 4.10: Solutions for Pareto optimal point number eight and DCWASA actual application

Day	Actual implementation			Point #8 solution		
	L_d^{dc}	C_d^k	G_d^k	L_d^{dc}	C_d^k	G_d^k
1	269.91	1	4	250.00	2	0
2	275.73	1	4	250.00	2	0
3	277.03	1	4	250.00	2	0
4	273.49	1	4	250.00	2	0
5	259.92	1	4	250.00	2	0
6	262.82	1	4	250.00	2	0
7	265.77	1	4	250.00	2	0
8	264.88	2	4	250.00	2	0
9	251.17	1	5	250.00	2	0
10	275.06	1	5	250.00	2	6
11	266.68	1	5	250.00	2	5
12	276.74	1	5	250.00	2	3
13	267.72	1	5	250.00	2	2
14	272.74	1	5	250.00	2	0
15	264.31	1	5	250.00	2	0
16	261.28	1	6	250.00	2	5
17	257.9	1	5	250.13	2	5
18	270.72	1	5	250.00	2	5
19	254	1	5	250.00	2	0
20	261.18	1	5	250.00	2	2
21	272.4	1	5	250.00	2	0
22	285.71	1	5	250.00	2	1
23	290.12	1	5	250.00	2	0
24	294.9	1	5	250.00	2	0
25	271.4	1	5	250.00	2	7
26	291.32	1	6	250.00	2	0
27	270.02	1	6	250.00	2	2
28	275.24	1	6	250.00	2	0
29	278.04	1	6	250.00	2	2
30	258.13	1	6	250.00	2	0
31	279.01	1	6	250.00	2	2
Total	8395.34	32	154	7750.13	62	47

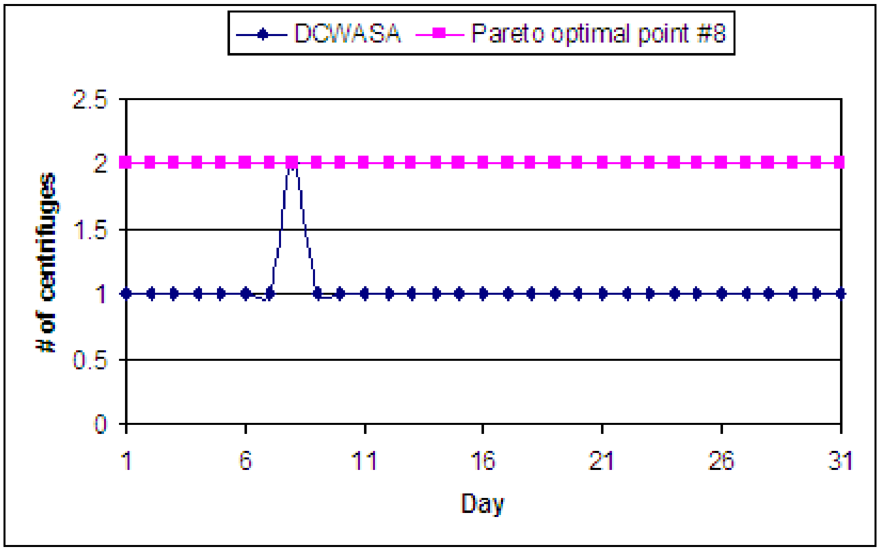


Figure 4.8: Daily solutions for the number of the on-site contractor centrifuges in service

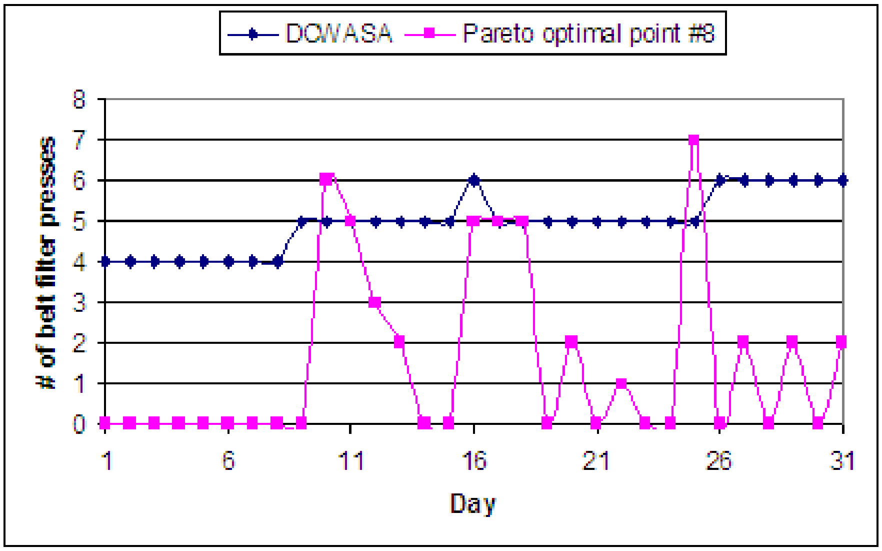


Figure 4.9: Daily solutions for the number of the on-site contractor belt filter presses in service

this end, it is important to point out that the actual implementation could be dominated by the optimal solution by at least $\$4,131,459 - \$4,119,590 = \$11,869$ up to $\$4,131,459 - \$4,104,790 = \$26,669$ over a 31-day time horizon, where $4,131,459 is the actual total cost implemented and $4,119,590 and $4,104,790 is the highest and lowest total cost values in the set of eight Pareto optimal points dominating the actual implementation (see Figure 4.6 on Page 132). Using similar calculations, the optimal total odor score could dominate the implemented total odor by at least 0.56 (i.e., $76.13 - 75.57$) odor index points up to 35.34 (i.e., $76.13 - 40.79$) odor index points. Hence, the analysis as previously discussed is necessary and useful to maintain the cost-effective wastewater treatment process.

Next, we perform a sensitivity analysis on the Base Case. First, we vary the percent flow from the blend tank to DCWASA and the on-site contractor. Then, we change the odor thresholds for each reuse field. More specifically, we use odor threshold set one where no wind directions were taken into account in the odor threshold calculations. Lastly, we vary DCWASA's dewatering cost.

4.4 Sensitivity Analysis on the Base Case

4.4.1 Sensitivity Analysis on the Percent Flow from Blend Tank

According to our previous analysis in this chapter, the percentage flow from blend tank to DCWASA was exogenously fixed to 90%. However, practically, there are many choices of flow we can choose ranging from 0% up to 100%. Therefore, it is interesting to see how the optimal solution would change when we vary the percentage flow. However, enumerating all values of percentage flows could be computationally prohibitive. Thus, we arbitrarily tried $10\%, 20\%, \ldots, 80\%$ of percentage flows as a reasonable compromise.

Next, in order to see how the Pareto optimal points would change relative to existing ones from the Base Case, for each $F_d^{dc} \in \{10\%, 20\%, \ldots, 80\%\}$, we reran problem (4.34) (see page 101 for problem (4.34)) and varied the maximum values τ in (4.35) with 28 total odor values from the 28 Pareto optimal points of the Base Case. Using the specified setup and given that we arbitrarily tried $10\%, 20\%, \ldots, 80\%$ of percentage flows (F_d^{dc}), there were $8 \times 28 = 224$ more subproblems in total. However, all 28 subproblems when $F_d^{dc} = 10\%$ were infeasible for the following reasons. There were at most two on-site contractor centrifuges with a maximum capacity of 50 DTS/centrifuge and seven belt filter presses with at most 25 DTS/belt filter press. Hence, the maximum daily DTS that the on-site contractor could handle was $2 \times 50 + 7 \times 25 = 275$ DTS. However, when 10% of flow was assigned to DCWASA, the remaining 90% of the flow or $0.9 \times 320 = 288$ DTS exceeded the maximum amount of solids that the on-site contractor could handle. Therefore, the problem became infeasible. It should be noted that this maximum processing constraint is (4.12) (see page 94).

For the remaining 196 subproblems where $F_d^{dc} \in \{20\%, 30\%, \ldots, 80\%\}$, all of them could be solved to optimality. However, not all subproblems returned Pareto optimal solutions. In particular, the optimal total odors from those subproblems did not equal their corresponding maximum value τ (see Section 2.1.1 for the verification of the Pareto optimal solution). Next, we discuss the solutions to these subproblems and compare the Pareto optimal points to Base Case.

Figure 4.10 depicts the Pareto optimal points found when $F_d^{dc} \in \{20\%, 30\%, \ldots, 90\%\}$. It can be seen that the Pareto optimal points move to the left when the percent flow from the blend tank to DCWASA decreases. Otherwise stated, less total costs and less total odors were expected when less flow was handled by DCWASA. It is important to point out here first that this observation does not necessarily imply that the on-site contractor would be more cost efficient than DCWASA. This cost behavior corresponds to how the model was set up and the odor prediction equation used rather than the efficiencies of either DCWASA or the on-site contractor themselves. For example, the on-site contractor did not apply polymer during the dewatering

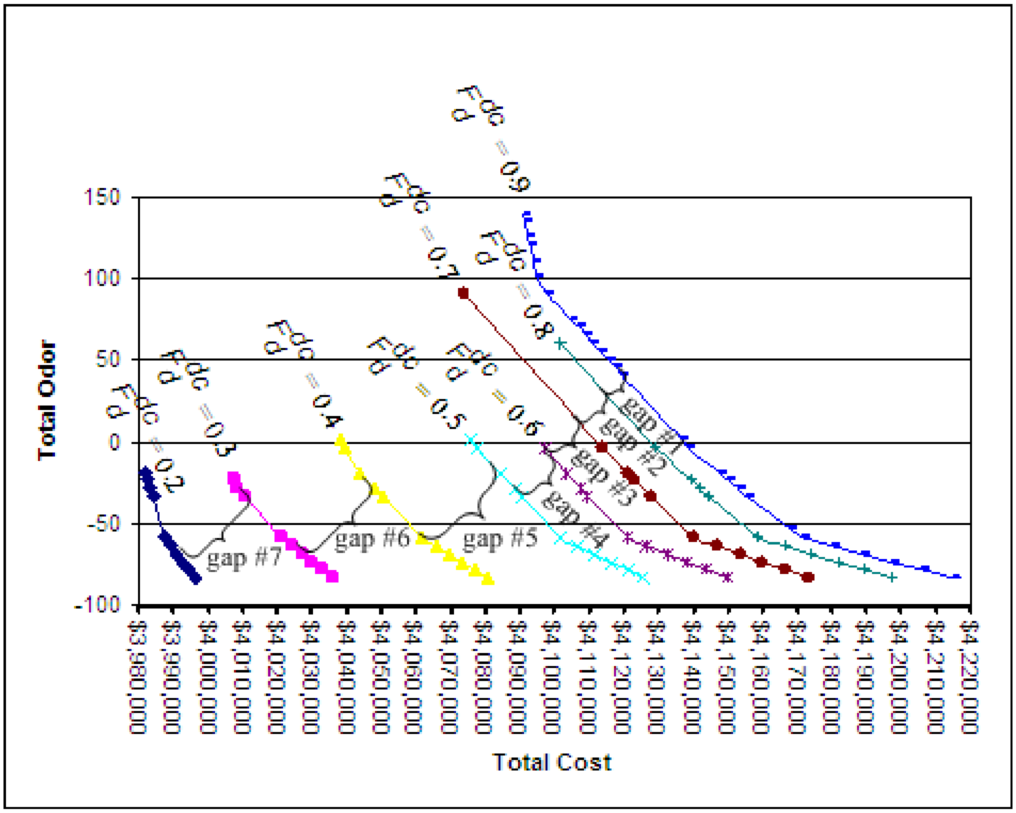


Figure 4.10: Pareto optimal points when $F_d^{dc} \in \{20\%, 30\%, \ldots, 90\%\}$

process and therefore less polymer cost was expected when more flow was assigned to them. This resulted in part in the overall total costs. Another reason is that this optimization model was constructed from the odor prediction equation (see equation (3.2), page 55) containing mostly odor-reducing variables relevant to the on-site contractor. Thus, when more flows were assigned to the on-site contractor, less total odors could be achieved. Next, we discuss this cost behavior in more detail.

From Figure 4.10, it can be seen that the set of Pareto optimal points when $F_d^{dc} = 0.2$ dominates the set of Pareto optimal points when $F_d^{dc} = 0.3$. Also, the set of Pareto optimal points when $F_d^{dc} = 0.3$ dominates the set of Pareto optimal points when $F_d^{dc} = 0.4$, and so on. In addition, the gaps between the Pareto optimal curves when $F_d^{dc} \in \{0.2, 0.3, \ldots, 0.5\}$ (gaps #5, #6, and #7) are significantly wider than those when $F_d^{dc} \in \{0.5, 0.6, \ldots, 0.9\}$ (gaps #1 to #4). This last observation corresponds directly to the on-site contractor's dewatering cost being a step function and will be discussed later.

To see things more clearly, we can categorize all costs constituting the total cost in (4.19) into five categories regardless of whether they are from an on-site contractor or DCWASA. The five categories of costs are:

1. Lime cost

2. Polymer cost

3. Dewatering cost

4. Centrifuge and belt filter press cost

5. Hauling cost

To further analyze these costs behaviors and be able to compare the total costs, we arbitrarily picked one Pareto optimal point sharing a common total odor score from each of the Pareto optimal sets when $F_d^{dc} \in \{20\%, 30\%, \ldots, 90\%\}$; this point has an odor score of 33.396. Table 4.11 shows five important cost values to analyze these Pareto optimal points. From Table 4.11, it can be seen that cost categories two (polymer cost) and three (dewatering cost) always decreased when F_d^{dc} decreased. Meanwhile, cost category five (hauling cost) always increased. As for the other cost categories, the cost figures were not monotonic. Next, we discuss the reasons for these costs behaving as they did.

Table 4.11: Five categories of costs from the Pareto optimal points when total odors $= -33.396$

F_d^{dc}	Cost category					Total cost
	1 (lime cost)	2 (polymer cost)	3 (Dewatering cost)	4 (Centrifuge & belt filter press cost)	5 (Hauling cost)	
0.9	\$165,012	\$1,962,353	\$874,696	\$92,008	\$1,061,757	\$4,155,826
0.8	\$162,292	\$1,944,431	\$856,592	\$92,008	\$1,089,474	\$4,144,798
0.7	\$159,573	\$1,926,510	\$838,488	\$85,932	\$1,117,192	\$4,127,694
0.6	\$166,242	\$1,908,588	\$820,384	\$69,656	\$1,144,909	\$4,109,779
0.5	\$161,958	\$1,890,666	\$795,615	\$69,656	\$1,172,626	\$4,090,521
0.4	\$157,674	\$1,872,745	\$756,183	\$63,580	\$1,200,343	\$4,050,525
0.3	\$153,389	\$1,854,823	\$716,751	\$57,504	\$1,228,422	\$4,010,889
0.2	\$145,976	\$1,836,901	\$677,319	\$67,704	\$1,256,173	\$3,984,073

First, since there was no polymer cost charged by the on-site contractor in the dewatering process, total polymer cost always decreased when more flow was handled by the on-site contractor. It is important to note that polymer added at DAF was added to a portion of flow before the flow was directed to either DCWASA or the on-site contractor. Therefore, the polymer cost at DAF was calculated independently from the percent flows to DCWASA or the on-site contractor.

Second, DCWASA's dewatering cost was \$90/DTS. Meanwhile, the on-site contractor's dewatering cost was either \$71.75/DTS or \$50.25/DTS[2]. Obviously, the more flow assigned to the on-site contractor, the less dewatering cost expected.

Third, biosolids produced by the on-site contractor contained less percent solids. In other words, given the same amount of flow, biosolids produced by the on-site contractor weighted more than those produced by DCWASA. Consequently, the more flow handled by the on-site contractor, the greater the hauling cost, all things being equal.

Lastly, DCWASA's lime additions and the on-site contractor's centrifuges and belt filter presses are decision variables in the optimization problem. Since they are interchangeable in reducing biosolids odor levels, a particular solution may favor one variable over another at various stages (e.g., amount of solids being processed and desired odor level). Thus, cost categories one and four were not monotonic. So far, we have shown that for two of the five cost categories, less cost was expected when less flow was handled by DCWASA. The only cost that always increased when less flow was handled by DCWASA was the hauling cost. This observation corresponds directly to (4.2), where the amount of wet tons of biosolids produced, on each day, by DCWASA is $\frac{B_d F_d^{dc}}{0.3107}$ and by the on-site contractor it is $\frac{B_d F_d^k}{0.2580}$. It can be seen that

[2] See Section 3.3.2 for the on-site contractor's dewatering cost details.

when less flow goes to DCWASA (or, equivalently, more flow goes to the on-site contractor), more wet tons of biosolids would be generated and therefore hauling costs would increase. Tables 4.12 displays the changes in five categories of costs when F_d^{dc} decreased, where positive values indicate an increase in cost and negative values a decrease.

Table 4.12: Changes in five categories of costs from the Pareto optimal point with total odor = −33.396

F_d^{dc}	Cost category					Total changes
	1 (lime cost)	2 (polymer cost)	3 (dewatering) cost	4 (centrifuge & belt filter press cost)	5 (hauling cost)	
0.9–0.8 (gap #1)	−$2,719	−$17,922	−$18,104	$0	$27,717	−$11,028
0.8–0.7 (gap #2)	−$2,719	−$17,922	−$18,104	−$6,076	$27,717	−$17,104
0.7–0.6 (gap #3)	$6,670	−$17,922	−$18,104	−$16,276	$27,717	−$17,915
0.6–0.5 (gap #4)	−$4,284	−$17,922	−$24,769	$0	$27,717	−$19,258
0.5–0.4 (gap #5)	−$4,284	−$17,922	−$39,432	−$6,076	$27,717	−$39,997
0.4–0.3 (gap #6)	−$4,284	−$17,922	−$39,432	−$6,076	$28,078	−$39,636
0.3–0.2 (gap #7)	−$7,414	−$17,922	−$39,432	$10,200	$27,751	−$26,816

The last column in Table 4.12 clearly shows why Pareto optimal points moved to the left when F_d^{dc} decreased. Another observation we have previously mentioned is that gaps #5, #6, and #7 are significantly wider than gaps #1, #2, #3, and #4 according to Figure 4.10 and as evidenced in the last three rows of Table 4.12.

Comparing these three rows with the rest, out of all cost categories, the decreases in cost category three made gaps #5, #6, and #7 significantly wider than other gaps. This can be explained as follows.

The on-site contractor's dewatering cost was a step function where the dewatering cost became cheaper for a certain level of flow from the blend tank. In particular, the on-site contractor's dewatering cost was \$71.75/DTS for the first 150 tons and then \$50.25/DTS thereafter[3]. Consequently, when more than 150 DTS were handled by the on-site contractor the dewatering cost per DTS became cheaper. Now, given that there were 320 DTS per day, when $F_d^{dc} = 0.5$ there were $(1 - 0.5) \times 320 = 160$ DTS handled by the on-site contractor. Using similar calculations, there were 192, 224, and 256 DTS handled by the on-site contractor, respectively, when $F_d^{dc} = 0.4$, 0.3, and 0.2. This is the reason why gaps #5, #6, and #7 are significantly wider than the others.

Another interesting observation, as can be seen from Figure 4.10, is that there are few Pareto optimal points where total odor > 0 when $F_d^{dc} \in \{0.2, 0.3, \ldots, 0.6\}$. For the sake of discussion and to address the subject more clearly, Table 4.13 shows the optimal objective function values from all 28 runs when $F_d^{dc} = 0.2$. As discussed in Section 4.2.2, in verifying the Pareto optimal solutions, one needs to verify if all the constraints on the objectives are binding at the optimal solution. For our model, we have only one constraint on the objective which was the total odor. Therefore, we only need to verify, for each run, if the reported total odor (4th column) equaled its corresponding τ value (2nd column). From Table 4.13, it can be seen that, for

[3] See Section 3.3.2.

Table 4.13: Optimal objective function values when $F_d^{dc} = 0.2$

Run #	τ	Total cost ($)	Total odor (index points)	Pareto optimal (yes/no)
1	139.68400	3,980,286.80598	−4.32600	No
2	135.86400	3,980,286.80598	−4.32600	No
3	126.31400	3,980,286.80598	−4.32600	No
4	120.58400	3,980,286.80598	−4.32600	No
5	111.03400	3,980,286.80598	−4.32600	No
6	101.48400	3,980,286.80598	−4.32600	No
7	91.55400	3,980,286.80598	−4.32600	No
8	75.57270	3,980,286.80598	−4.32600	No
9	70.87270	3,980,286.80598	−4.32600	No
10	65.70270	3,980,286.80598	−4.32600	No
11	61.47400	3,980,286.80598	−4.32600	No
12	55.83270	3,980,286.80598	−4.32600	No
13	50.66270	3,980,286.80598	−4.32600	No
14	45.96270	3,980,286.80598	−4.32600	No
15	40.79270	3,980,286.80598	−4.32600	No
16	1.64037	3,980,286.80598	−4.32600	No
17	−3.38600	3,980,286.80598	−4.32600	No
18	−18.35963	3,981,892.47958	−18.35963	Yes
19	−23.35963	3,982,510.71958	−23.35963	Yes
20	−28.35963	3,983,128.95958	−28.35963	Yes
21	−33.39600	3,984,073.04555	−33.39600	Yes
22	−53.49770	3,986,491.23143	−53.91600	No
23	−58.35963	3,987,430.49158	−58.35963	Yes
24	−63.35963	3,989,350.49158	−63.35963	Yes
25	−68.35963	3,991,270.49158	−68.35963	Yes
26	−73.35963	3,992,948.99158	−73.35963	Yes
27	−78.35963	3,994,868.99158	−78.35963	Yes
28	−83.35963	3,996,788.99158	−83.35963	Yes

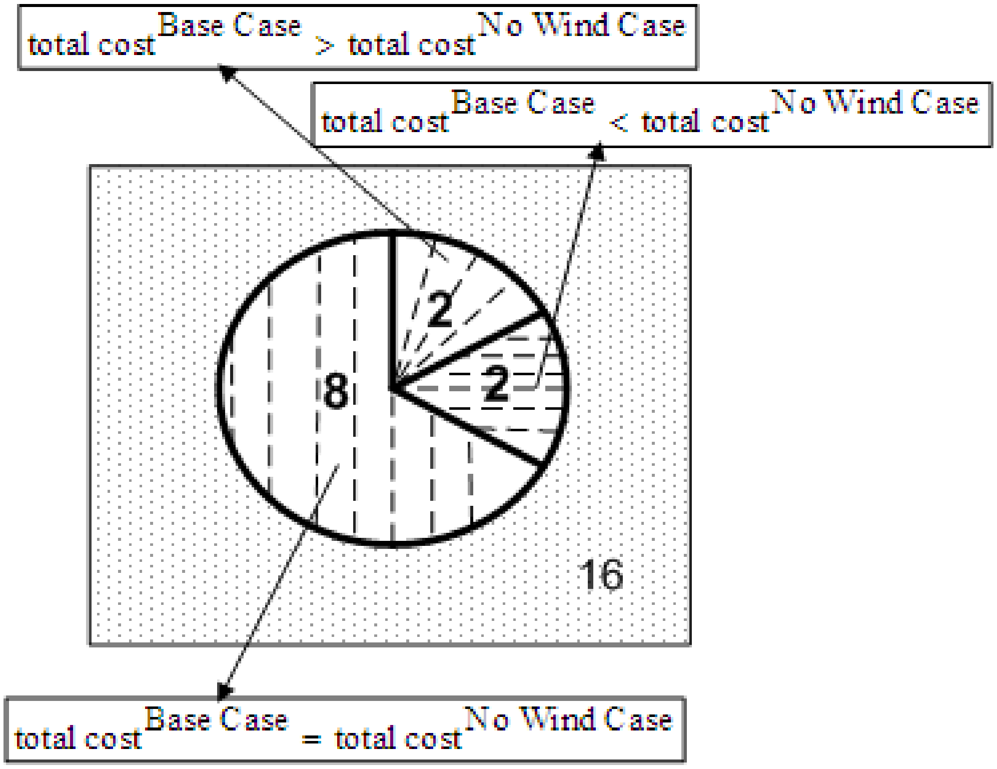
the first 17 runs, none of the the reported total odors equaled their corresponding τ values and hence no Pareto optimal solutions were obtained. In addition, the reported total odor values were a lot lower than their corresponding τ values. The first Pareto optimal point found (point #18) was when $\tau = -18.35963$.

In the next section, we perform another sensitivity analysis where the input for odor thresholds changes.

4.4.2 Sensitivity Analysis on Odor Threshold Input

In this case, the field-specific odor thresholds (O_f^{up}) were adjusted (labeled "No Wind"). In particular, we used the odor threshold set 1 (see Section 3.4.1 for odor threshold set 1 calculations) as input to odor threshold for each field f (O_f^{up}). As the odor threshold for each reuse field was changed, some reuse fields would be more restricted meaning O_f^{up} decreased while other reuse fields would be less restricted (corresponding to when O_f^{up} increased) due to (4.7) (see page 92). In fact, there were 559 reuse fields that became more restricted, while the remaining 223 reuse reuse fields became less restricted. Consequently, higher total costs were expected for Pareto optimal points with the same total odors from the Base Case. However, the results are counter-intuitive. Next, we discuss the solution procedures and computational results.

Since we were interested in investigating how the Pareto optimal points would change relative to the existing ones from the Base Case, we used the total odor values from the Pareto optimal points of the Base Case as the maximum values τ in (4.35) and reran problem (4.34) with odor threshold from set 1 as input to O_f^{up}. Given there were 28 Pareto optimal points from the Base Case, we also had 28 total odor values to be used as the τ values. Subsequently, 28 subproblems were constructed. We were able to solve all 28 subproblems to optimality. However, after verification, only 12 distinct Pareto optimal solutions resulted. In particular, there were only 12 subproblems for which total odors equaled their corresponding τ values. Figure 4.11 shows how total costs associated with these 12 Pareto optimal points obtained from the No Wind Case are related to the total costs associated with the Pareto optimal points[4] obtained from Base Case. In particular, the rectangle represents the



$\text{total cost}^{\text{Base Case}} > \text{total cost}^{\text{No Wind Case}}$
$\text{total cost}^{\text{Base Case}} < \text{total cost}^{\text{No Wind Case}}$
2
2
8
16
$\text{total cost}^{\text{Base Case}} = \text{total cost}^{\text{No Wind Case}}$

Figure 4.11: Pareto optimal points from Base Case and No Wind Case

[4] Only Pareto optimal points from Base Case with common total odors to the 12 Pareto optimal points obtained from the No Wind Case.

set of all 28 subproblems being solved and the area outside the circle represents the 16 subproblems not yielding Pareto optimal solutions. In addition, the whole circle represents the 12 Pareto optimal points obtained. In particular, when comparing these Pareto optimal points to those from Base Case with common total odors, eight of them had similar total costs to those from Base Case. In other words, these eight Pareto optimal points were common to both the Base and No Wind cases. For the other four Pareto optimal points, two of them had total costs greater than those of the Base Case, while the remaining two Pareto optimal points had total costs less than those of the Base Case. Figure 4.12 depicts those 12 Pareto optimal points from both the Base and No Wind cases while Table 4.14 shows the details for total cost and total odor values of those from the No Wind Case. In addition, the fourth column of the table displays the sign telling if the total cost was less or greater than or equal to the total cost for the corresponding point from the Base Case. These signs corresponding to their Pareto optimal points are also displayed in Figure 4.12. Thus, we may conclude that the changes in odor thresholds may or may not always effect the optimal objective function value of total cost, given the Pareto optimal total odor in question. This is due presumably to the following three reasons. First, the distribution costs to each reuse fields were the same. Second, the set of candidate reuse fields are large. Finally, the modified odor thresholds were not restrictive enough (i.e., modified odor thresholds were not low enough). Thus, although the odor thresholds were modified, the solver could still find other combinations of reuse fields from the large set available to arrive at total costs the same as before or not much different.

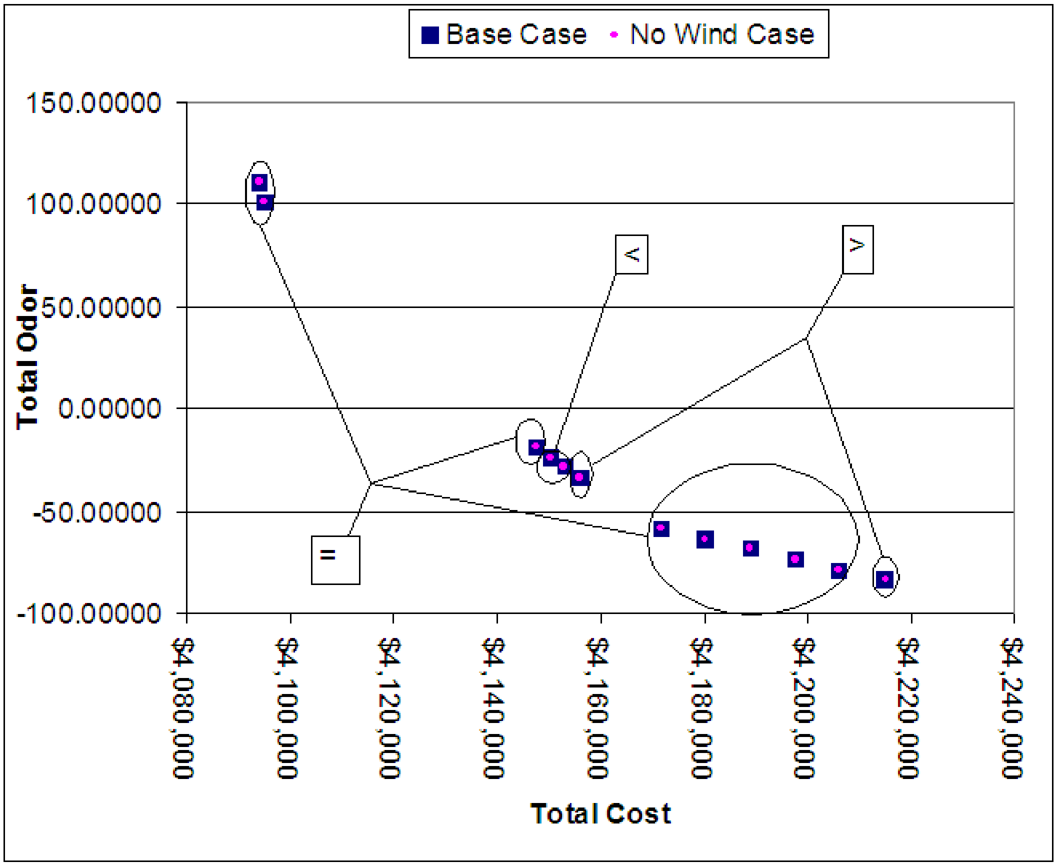


Figure 4.12: Pareto optimal points from Base Case and No Wind Case

Table 4.14: "No Wind" Case Pareto optimal points

Point	Total cost ($)	Total odor	Compared to the total cost from the Base Case
1	4,094,015.66978	111.03400	=
2	4,094,995.66978	101.48400	=
3	4,147,416.66242	−18.35963	=
4	4,150,198.74242	−23.35963	<
5	4,152,980.82242	−28.35963	<
6	4,155,825.74783	−33.39600	>
7	4,171,625.94242	−58.35963	=
8	4,180,265.94242	−63.35963	=
9	4,188,905.94242	−68.35963	=
10	4,197,545.94242	−73.35963	=
11	4,206,185.94242	−78.35963	=
12	4,215,204.39042	−83.35963	>

To summarize this section, the modified odor thresholds may not always affect the Pareto optimal points. Nevertheless, the optimal solutions for decision variables may change (e.g., F_{dcf}, L_d^{dc}) and need to be taken into account in implementing the solutions.

Next, we discuss a sensitivity analysis on DCWASA's operating costs.

4.4.3 Sensitivity Analysis on DCWASA's Operating Costs

In this section, we reran problem (4.34) with two different values for DCWASA's dewatering cost: \$80/DTS and \$70/DTS. The maximum τ values were set up the same way we solved the Base Case. Figures 4.13 and 4.14 display the Pareto optimal points when DCWASA's dewatering cost = \$80/DTS and \$70/DTS, respectively. We see that when DCWASA's dewatering cost = \$80/DTS, the Pareto optimal set when $F_d^{dc} = 0.2$ still dominates the Pareto optimal sets when $F_d^{dc} \in \{0.3, \ldots, 0.9\}$. This phenomenon is similar to the case when DCWASA's dewatering cost = \$90/DTS. However, when DCWASA's dewatering cost = \$70/DTS, the highlighted Pareto optimal points in Figure 4.14 corresponding to $F_d^{dc} \in \{0.2, 0.3, 0.9\}$ are to the southwest of the other groups of Pareto optimal points and are non-dominated with respect to each other. Table 4.15 also shows their corresponding detailed total costs and total odors.

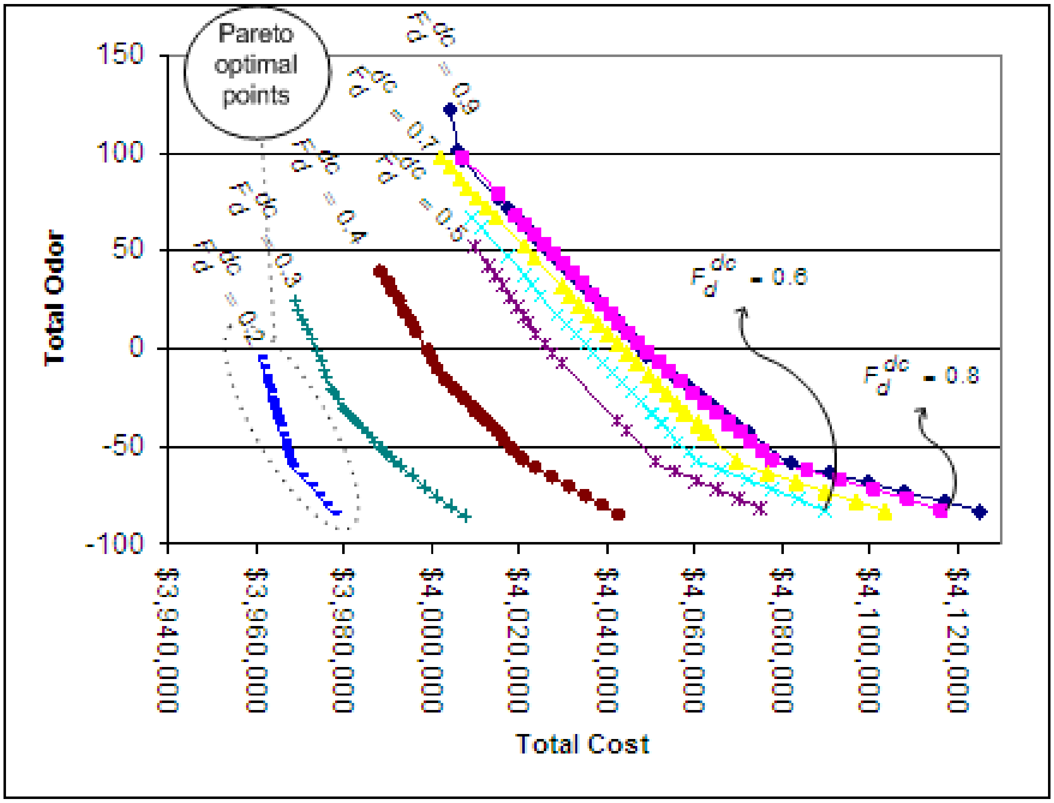


Figure 4.13: Pareto optimal points when DCWASA's operating cost = $80/DTS

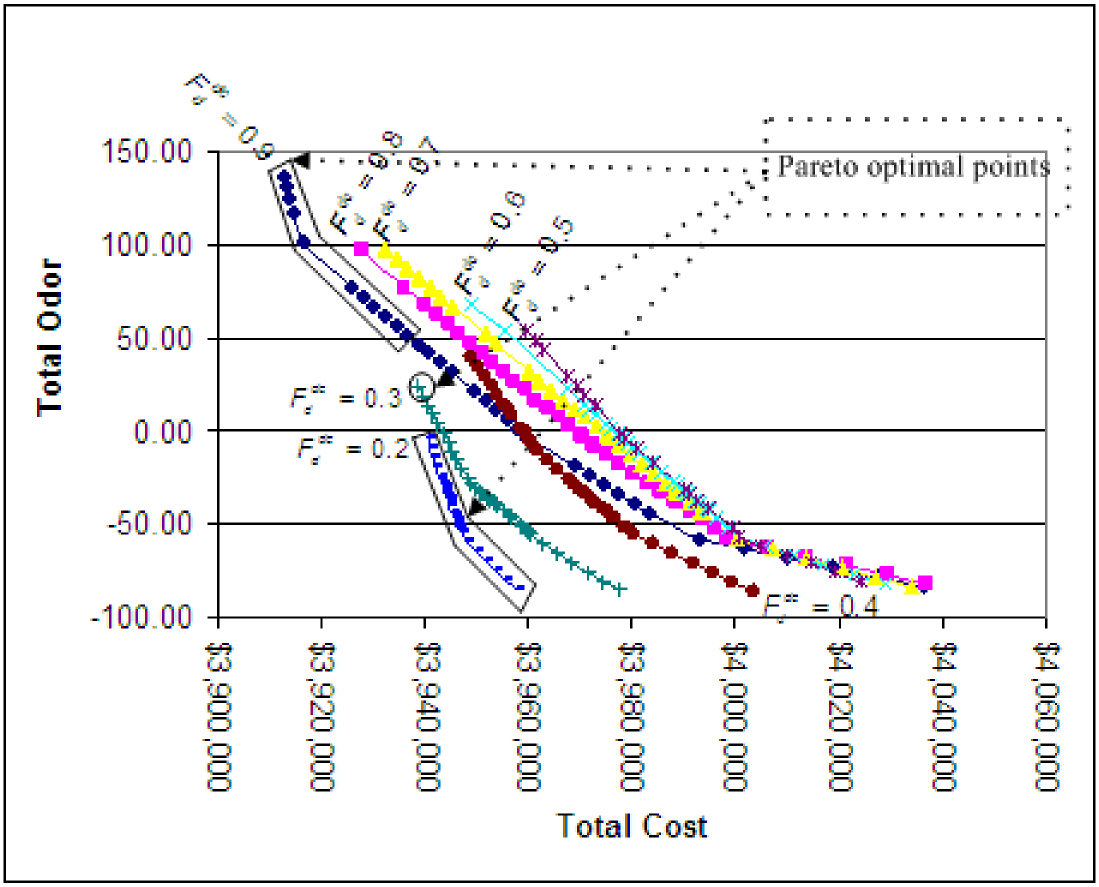


Figure 4.14: Pareto optimal points when DCWASA's operating cost = $70/DTS

Table 4.15: Pareto optimal points obtained when DCWASA's dewatering cost = $70/DTS

Pareto optimal point #	F_d^{dc}	Total cost ($)	Total odor
1	0.9	3,912,907.67	135.86
2	0.9	3,913,495.67	130.13
3	0.9	3,914,083.67	124.40
4	0.9	3,914,896.98	116.75
5	0.9	3,916,435.67	101.48
6	0.9	3,926,029.91	76.68
7	0.9	3,928,229.91	71.51
8	0.9	3,930,027.67	66.64
9	0.9	3,932,429.91	61.64
10	0.9	3,934,629.91	56.47
11	0.9	3,936,427.67	51.60
12	0.3	3,938,795.47	24.81
13	0.3	3,939,730.8	19.68
14	0.2	3,940,506.72	−4.33
15	0.2	3,940,963.09	−9.46
16	0.2	3,941,036.05	−8.52
17	0.2	3,941,631.25	−14.59
18	0.2	3,942,299.41	−19.72
19	0.2	3,943,121.17	−25.25
20	0.2	3,943,201.81	−24.33
21	0.2	3,943,413.01	−28.27
22	0.2	3,943,820.05	−29.33
23	0.2	3,944,081.17	−33.40
24	0.2	3,944,154.13	−32.46
25	0.2	3,944,357.65	−35.25
26	0.2	3,944,438.29	−34.33
27	0.2	3,944,538.13	−35.72
28	0.2	3,944,718.61	−36.19
29	0.2	3,944,749.33	−38.53
30	0.2	3,944,822.29	−37.59
31	0.2	3,945,056.53	−39.33
32	0.2	3,945,474.91	−43.66
33	0.2	3,945,732.19	−44.33
34	0.2	3,945,962.59	−46.06
35	0.2	3,946,216.03	−47.85
36	0.2	3,946,350.43	−49.33
37	0.2	3,946,465.63	−50.76
38	0.2	3,946,646.11	−51.23

Continued on next page

Table 4.15 – continued from previous page

Pareto optimal point#	F_d^{dc}	Total Cost ($)	Total Odor
39	0.2	3,946,884.19	-52.98
40	0.2	3,946,968.67	-54.33
41	0.2	3,948,004.86	-59.33
42	0.2	3,950,133.39	-64.33
43	0.2	3,952,015.2	-69.33
44	0.2	3,953,723.41	-74.33
45	0.2	3,955,560.08	-79.33
46	0.2	3,957,875.29	-84.33

Given that we only obtained subsets of the Pareto frontiers, we should make sure that the Pareto optimal points corresponding to $F_d^{dc} \in \{0.2, 0.3, 0.9\}$ are not dominated by Pareto optimal points from other values of F_d^{dc} but not yet generated. That is we need to make sure that no other Pareto optimal points not yet generated will lie to the southwest of existing ones. In fact, we are able to show this by obtaining the minimum possible total odor when $F_d^{dc} \in \{0.2, 0.3, \ldots, 0.8\}$ and plotting these values as indicated by the vertical lines in Figure 4.15. We can see that if more Pareto optimal points were generated, they would be to the right of these vertical lines with higher total odors and, therefore, would not lie to the southwest of the Pareto optimal points. Thus, our existing Pareto optimal points cannot be dominated by Pareto optimal points not yet generated.

Next, we discuss why the Pareto optimal points[5] did not always correspond to when $F_d^{dc} = 0.2$, when we set DCWASA's dewatering cost = $70/DTS as opposed to

[5] Note that all points on Figure 4.14 are Pareto optimal associated with each F_d^{dc} value. However, when we consider all of them together, the actual Pareto optimal points are only points on the southwest envelop.

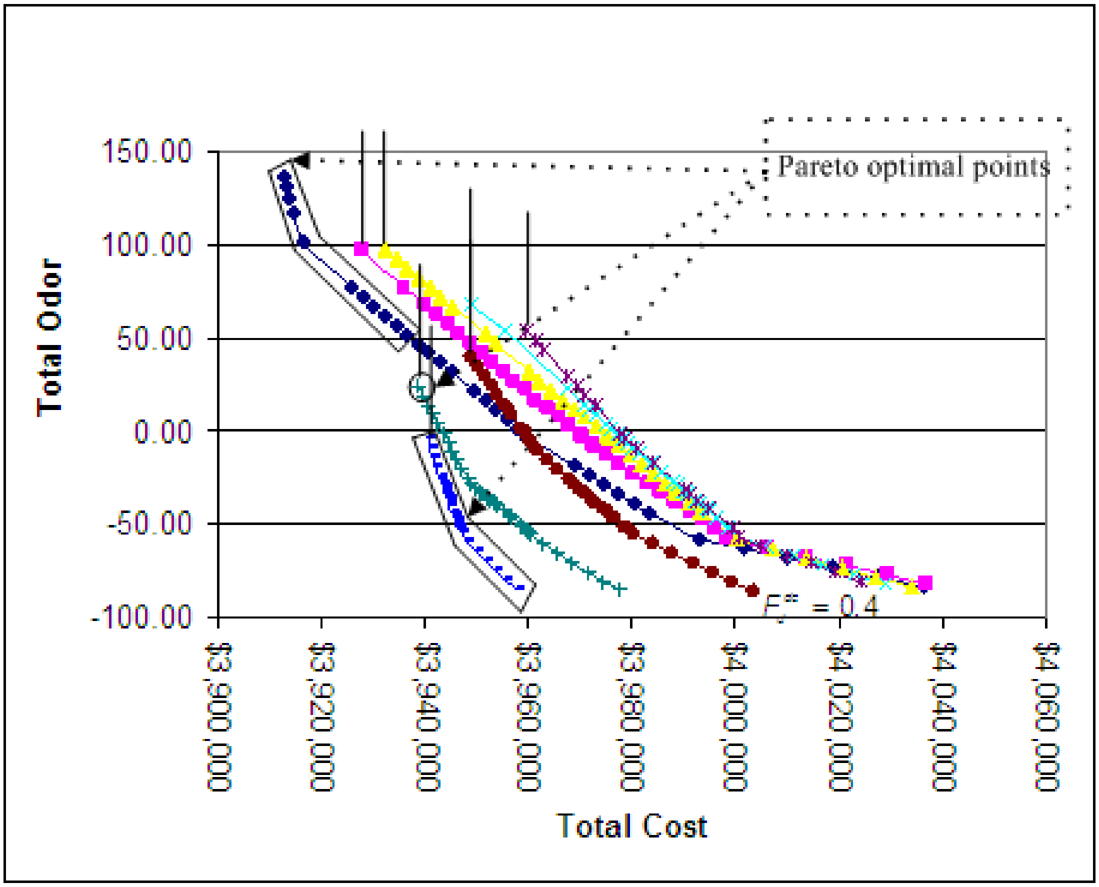


Figure 4.15: Possible minimum total costs when $F_d^{dc} \in \{0.2, 0.3, \ldots, 0.8\}$

\$90/DTS. This cost behavior is due to the following two connected reasons. First, when $F_d^{dc} = 0.2$, or 80% of the flow was handled by the on-site contractor, all of the on-site contractor's centrifuges and belt filter presses needed to be in service in order to satisfy (4.12) (see page 94 for equation (4.12)). In other words, on each day, two on-site contractor's centrifuges and seven on-site contractor's belt filter presses needed to be in service in order to handle $320 \times 0.8 = 256$ DTS assigned to them. Hence, when $F_d^{dc} = 0.2$, the total costs could only decrease a certain amount taking into account that the centrifuges and belt filter presses were used to their limit. Second, although the first observation is also true regardless of the value of DCWASA's dewatering cost, it is only when this cost = \$70/DTS that the total cost from $F_d^{dc} = 0.9$ is as low as the one from $F_d^{dc} = 0.2$ (for some points).

4.5 Conclusion

In this chapter we began our analysis with the Base Case optimization model and discussed the three portions of the Pareto optimal frontier. The roles of decision variables (e.g., lime addition, # of centrifuges) for each of these three portions were analyzed. Failure to understand the key roles of decision variables at various stages may cause as much as \$6,600 in cost discrepancy over a 31-day time horizon. Then, we provided a sensitivity analysis on the Base Case beginning the analysis by varying the percentage flow from the blend tank to DCWASA in the range $\{0.10, 0.20, \ldots, 0.90\}$. We found that the set of Pareto optimal points for $F_d^{dc} = 0.2$ dominated the other values for F_d^{dc}. In addition, we explained why when $F_d^{dc} = 0.10$, the problem became infeasible. Next sensitivity analysis was described for odor threshold. It was shown that the input for odor thresholds had no strong effects on determining the Pareto optimal points; that is most of the Pareto optimal points remained unchanged. We also suggested if there would have been less reuse fields, changes in odor thresholds might have had more of an effect on the solutions. Lastly, we did a sensitivity analysis on DCWASA's operating costs. We showed that when DCWASA's operating cost equaled \$70/DTS, the Pareto optimal points may not always correspond to $F_d^{dc} = 0.2$.

Thus, in the next chapter, the percentage of the flow to DCWASA F_d^{dc}, is taken to be a decision variable as oppose to being exogenously determined. Not only does this allow us to solve for optimal F_d^{dc} endogenously, but this also allows F_d^{dc} to vary across all days. Nevertheless, having F_d^{dc} as a decision variable transforms the problem from a mixed integer linear program to a mixed integer non-linear one due to the presence of *bilinear* terms (e.g., product of decision variables). In the next chapter, we discuss this modified problem and its approximation by a mixed integer linear program. The method used to approximate the bilinear term as well as the optimal solutions to this modified model are also discussed.

Chapter 5

Optimization Models with Decomposition Methods

One of the common setups we had for all previous optimization models was that the percent flows from the blend tank to DCWASA were fixed for each optimization model run. The setup is practical in the sense that DCWASA may want to know the optimal solutions for processing and distributing of biosolids, given the percentage flow handled by the on-site contractor. However, it is still interesting to find the optimal percent flow from the blend tank to DCWASA along with other optimal processing and distribution variables. Eventually, we could enumerate all problems by varying the percent flow for each day. However, doing so requires millions of models to be solved and becomes a large computational burden. For example, if we would like to try $F_d^{dc} = 0.1$–0.9 with a 0.1 increment over a 31-day period, it would require us to solve 9^{31} problems for one value of maximum value τ (if we were to try the constraint method). Then we need to compare 9^{31} optimal objective function values and pick the best one. Moreover, solving these 9^{31} problems does not even guarantee that a Pareto optimal point would be obtained.

Thus, we now present a modified model where the percentage flows F_d^{dc} are defined as decision variables as opposed to being exogenously determined. This will necessarily lead to computational difficulty but with a gain in model insights. Not only does the bilinear term that arises from the product of F_d^{dc} and L_d^{dc} make the objective function nonlinear, it also makes it nonconvex. Due to the existence of several suboptimal local optima in a bilinear program, even the most uncomplicated problem is hard to solve [50]. In this thesis, we take advantage of the decomposition technique presented by Gabriel et al [29] to decompose the bilinear term. The resulting approximation problem is then a linear integer program with can be solved by existing solvers, where no specialized procedures are required.

The rest of this chapter is organized as follows. In Section 5.1, we discuss the approximation of bilinear term involving two steps; the separation of bilinear term using Schur's decomposition [36, 29] and the approximation of quadratic terms, that arise from Schur's decomposition, using special ordered set type 2 (SOS2) variables [6, 5, 27]. In addition, extra constraints from the approximation of the bilinear terms are also discussed. Section 5.2 presents the full modified model where some necessary constraints pertinent to having defined F_d^{dc} as a decision variable are also discussed. Next, an analysis on the number of breakpoints used for the approximation process is made. Finally, we present the algorithm used to solve the modified model as well as computational results. Being able to obtain only one solution with zero percent optimality gap, in Section 5.3 we apply the Dantzig-Wolfe decomposition method to the modified model in order to obtain better lower bounds. With these modified lower bounds, in Section 5.4, we numerically prove that most of the integer solutions discussed in Section 5.3 were already close to optimal. We also present five more Pareto optimal points. Although we are able to obtain more Pareto optimal points by utilizing better bounds obtained from Dantzig-Wolfe decomposition technique, the overall run time was still extensive. Therefore, in Section 5.5, we modify the algorithm presented in Section 5.2 such that the modified bounds are included in the algorithm. The computational results follow and indicate that the run times were reduced substantially (e.g. as much as $\frac{1}{5}$ of the times from before). In addition, several more Pareto optimal points are obtained and presented. Section 5.6 compares the Pareto optimal points obtained in this chapter with the ones obtained in Chapter 4. The analysis shows that the set of Pareto optimal points obtained in this chapter dominates those obtained in Chapter 4. Finally, Section 5.7 concludes this chapter and suggests possible future work.

5.1 Approximation of the Bilinear Cost Function

The two major steps involved in the approximation of the bilinear term are the separation via Schur's decomposition [36, 29] and the piecewise linearization of the quadratic terms resulting from this decomposition using special ordered set of type 2 (SOS2) variables [6, 5, 27].

First, consider (4.20), DCWASA's lime cost $= 0.06 \sum_{d=1}^{|D|} L_d^{dc} F_d^{dc} B_d$. Given F_d^{dc} is decision variable, the bilinear term arises from the product of F_d^{dc} and L_d^{dc}. According to [29], the term $F_d^{dc} L_d^{dc}$ can be transformed to two quadratic terms using Schur's decomposition and some extra linear constraints. It can be shown that $F_d^{dc} L_d^{dc} = \frac{1}{2}({u_d}^2 - {v_d}^2)$ where $u_d = \frac{L_d^{dc}}{\sqrt{2}} + \frac{F_d^{dc}}{\sqrt{2}}$ and $v_d = -\frac{L_d^{dc}}{\sqrt{2}} + \frac{F_d^{dc}}{\sqrt{2}}$. Next, the terms ${u_d}^2$ and ${v_d}^2$ can be piecewise linearized by SOS type 2 variables, z_{id}^u and z_{jd}^v, as follows. Consider the breakpoints b_{id}^u and b_{jd}^v for the terms u_d and v_d, respectively. Then u_d can be written as $\sum_{i=1}^{|I|} z_{id}^u b_{id}^u$ and v_d as $\sum_{j=1}^{|J|} z_{jd}^v b_{jd}^v$ and, ${u_d}^2, {v_d}^2$ are replaced respectively, by $\sum_{i=1}^{|I|} z_{id}^u (b_{id}^u)^2$ and $\sum_{j=1}^{|J|} z_{jd}^v (b_{jd}^v)^2$ where $\sum_{i=1}^{|I|} z_{id}^u = 1, \sum_{j=1}^{|J|} z_{jd}^v = 1, 0 \leq z_{id}^u \leq 1, 0 \leq z_{jd}^v \leq 1, \forall i = 1, \ldots, |I|, j = 1, \ldots, |J|$. Thus, the bilinear term $F_d^{dc} L_d^{dc}$ can be approximated by $\frac{1}{2}\left(\sum_{i=1}^{|I|} z_{id}^u (b_{id}^u)^2 - \sum_{j=1}^{|J|} z_{jd}^v (b_{jd}^v)^2\right)$, with the additional constraints:

$$\forall d \in D;$$

$$\sum_{i=1}^{|I|} z_{id}^u b_{id}^u = \frac{L_d^{dc}}{\sqrt{2}} + \frac{F_d^{dc}}{\sqrt{2}} \tag{5.1}$$

$$\sum_{j=1}^{|J|} z_{jd}^v b_{jd}^v = -\frac{L_d^{dc}}{\sqrt{2}} + \frac{F_d^{dc}}{\sqrt{2}} \tag{5.2}$$

$$\sum_{i=1}^{|I|} z_{id}^u = 1 \tag{5.3}$$

$$\sum_{j=1}^{|J|} z_{jd}^v = 1 \tag{5.4}$$

$$\sum_{i=1}^{|I|-1} y_{id}^u = 1 \tag{5.5}$$

$$\sum_{j=1}^{|J|-1} y_{jd}^v = 1 \tag{5.6}$$

$$z_{1d}^u \leq y_{1d}^u, z_{2d}^u \leq y_{1d}^u + y_{2d}^u, ..., z_{|I|-1,d}^u \leq y_{|I|-2,d}^u + y_{|I|-1,d}^u, z_{|I|,d}^u \leq y_{|I|-1,d}^u \tag{5.7}$$

$$z_{1d}^v \leq y_{1d}^v, z_{2d}^v \leq y_{1d}^v + y_{2d}^v, ..., z_{|J|-1,d}^v \leq y_{|J|-2,d}^v + y_{|J|-1,d}^v, z_{|J|,d}^v \leq y_{|J|-1,d}^v \tag{5.8}$$

$$\forall d \in D, i \in I;$$

$$0 \leq z_{id}^u \leq 1 \tag{5.9}$$

$$y_{id}^u \in \{0, 1\} \tag{5.10}$$

$$\forall d \in D, j \in J;$$

$$0 \leq z_{jd}^v \leq 1 \tag{5.11}$$

$$y_{jd}^v \in \{0, 1\} \tag{5.12}$$

Lastly, (4.20) can be replaced by

$$\text{DCWASA's lime cost} = \frac{0.06}{2} \sum_{d=1}^{|D|} B_d \left(\sum_{i=1}^{|I|} z_{id}^u (b_{id}^u)^2 - \sum_{j=1}^{|J|} z_{jd}^v (b_{jd}^v)^2 \right) \tag{5.13}$$

It is important to note here that, the minimum and maximum values for the breakpoint b_{id}^u are consistent with the minimum and maximum value for the term $\frac{L_d^{dc}}{\sqrt{2}} + \frac{F_d^{dc}}{\sqrt{2}}$, determined by accordingly substituting the lower bound and upper bound for variables L_d^{dc} and F_d^{dc}. Similarly, $-\frac{L_d^{dc}}{\sqrt{2}} + \frac{F_d^{dc}}{\sqrt{2}}$ can be used to find the minimum and maximum values for the breakpoint b_{jd}^v. In particular, the minimum and maximum values for breakpoints b_{id}^u and b_{jd}^v can be calculated as follows.

$$\begin{aligned}
b_{id}^{u\ \min} &= \frac{L_d^{dc^{\min}}}{\sqrt{2}} + \frac{F_d^{dc^{\min}}}{\sqrt{2}} \\
&= \frac{250}{\sqrt{2}} + \frac{0}{\sqrt{2}} \\
&\approx 176.78 \\
b_{id}^{u\ \max} &= \frac{L_d^{dc^{\max}}}{\sqrt{2}} + \frac{F_d^{dc^{\max}}}{\sqrt{2}} \\
&= \frac{400}{\sqrt{2}} + \frac{1}{\sqrt{2}} \\
&\approx 283.55 \\
b_{jd}^{v\ \min} &= -\frac{L_d^{dc^{\max}}}{\sqrt{2}} + \frac{F_d^{dc^{\min}}}{\sqrt{2}} \\
&= -\frac{400}{\sqrt{2}} + \frac{0}{\sqrt{2}} \\
&\approx -282.84 \\
b_{jd}^{v\ \max} &= -\frac{L_d^{dc^{\min}}}{\sqrt{2}} + \frac{F_d^{dc^{\max}}}{\sqrt{2}} \\
&= -\frac{250}{\sqrt{2}} + \frac{1}{\sqrt{2}} \\
&\approx -176.07
\end{aligned}$$

Next, we demonstrate the use of SOS2 variables through the approximation of the quadratic term u_d^2 above. However, for the sake of discussion, we modified the minimum and maximum values of b_{id}^u to 0 and 30, respectively. In addition, we drop an index d and suppose that there is only one u^2 function to be approximated. Finally, we use four breakpoints; $b_1^u = 0, b_2^u = 10, b_3^u = 20$ and $b_4^u = 30$ to approximate u^2, denoted by $\tilde{u^2}$. Figure 5.1 depict the graph of function u^2 being approximated by three linear lines corresponding to the four breakpoints. Now, following the

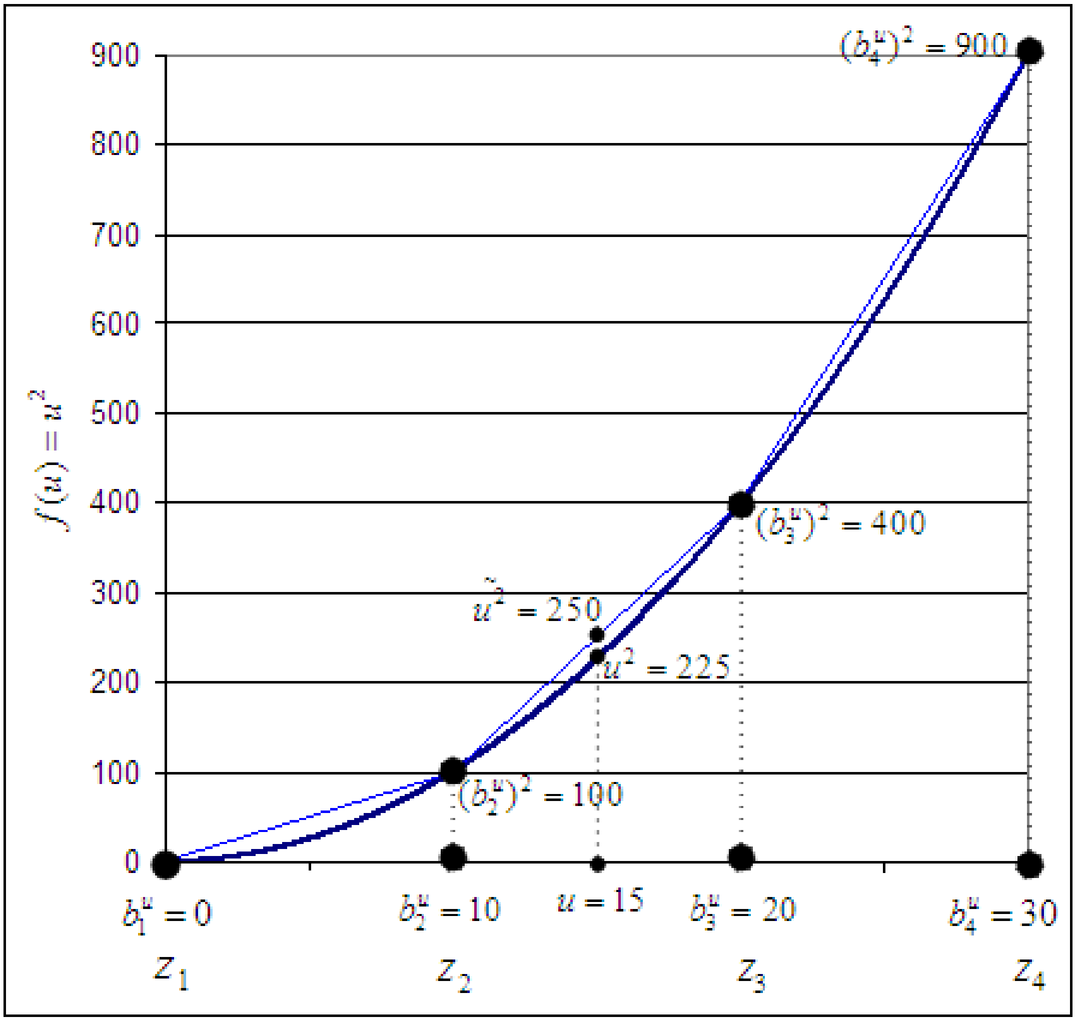


Figure 5.1: Demonstration for the use of SOS2 variables

formulation we have above, u may by written as $\sum_{i=1}^{4} z_i^u b_i^u$ and u^2 may be approximated by $\sum_{i=1}^{4} z_i^u (b_i^u)^2$ with the following constraints.

$$\begin{aligned} z_1^u \leq & y_1^u \\ z_2^u \leq & y_1^u + y_2^u \\ z_3^u \leq & y_2^u + y_3^u \\ z_4^u \leq & y_3^u \\ z_1^u + z_2^u + z_3^u + z_4^u = & 1 \\ y_1^u + y_2^u + y_3^u = & 1 \\ z_1^u, z_2^u, z_3^u, z_4 \geq & 0 \\ y_1^u, y_2^u, y_3^u \in & \{0, 1\} \end{aligned}$$

To illustrate the solution idea, take $u = 15$ as shown in Figure 5.1. Then the solution to the approximating problem is $y_2^u = 1, z_2^u = 0.5$ and $z_3^u = 0.5$. All other variables are equal to zero. Finally, we may write

$$\begin{aligned} u = & z_2^u b_2^u + z_3^u b_3^u \\ = & \frac{1}{2}(10) + \frac{1}{2}(20) = 15 \end{aligned}$$

and u^2 may be approximated by

$$\begin{aligned} \tilde{u^2} = & z_2^u (b_2^u)^2 + z_3^u (b_3^u)^2 \\ = & \frac{1}{2}(100) + \frac{1}{2}(400) = 250 \end{aligned}$$

To conclude this section, we would like to note that the separation technique used here relies mostly on the work developed by Gabriel et al. [29] and is general in the sense that it does not limit the number of variables in the product term. However, it is important to note that the earlier work by Beale [4] has also proposed a technique to express the bilinear term as a difference between two quadratic functions. According to Beale [4], it can be shown that $xy = \left(\frac{x+y}{2}\right)^2 - \left(\frac{x-y}{2}\right)^2$. In fact, after some some arithmetic, it is easy to show that the resulting separation terms from Beale's and the technique we used are similar. In the next section, we discuss some additional constraints and present the newly modified optimization model.

5.2 Optimization Model with Schur's Decomposition and SOS2 Variables

Besides the constraints relevant to the approximation of the bilinear terms, there are other constraints arisen from having defined F_d^{dc}, and subsequently also F_d^k, as decision variables. In addition, some constraints need to be modified. First, consider constraint (4.12),

$$\forall d \in D;\ 50C_d^k + 25G_d^k \geq F_d^k B_d.$$

This constraint works well when F_d^k is exogenously determined and given that $F_d^k > 0$, which were the case in Chapter 4. Now, that F_d^k is a decision variable and could take a zero value, we need additional constraints to force C_d^k and G_d^k to take zero values, whenever F_d^k equals zero. Moreover, from a cost-efficiency perspective, the flow should be assigned to the on-site contractor at least to have either one centrifuge or one belt filter press operating at their full capacities. According to our problem statement in Section 4.1, the dewatering capacities of one centrifuge and one belt filter press were approximately 50 DTS per day and 25 DTS per day, respectively (D. Tolbert, personal communication, November 18, 2003). Hence, given there are 320 DTS per day, the flow to the on-site contractor should be the minimum of $\{\frac{25}{320}, \frac{50}{320}\}$ or $\frac{25}{320} = 0.078 \approx 0.08$. In other words, F_d^k should be at least 0.08 or zero otherwise. Constraints (5.14) – (5.19) together with Constraint (4.12) enforce the above requirements.

$$\forall d \in D;$$

$$C_d^k \leq 2 \times y_d^c \tag{5.14}$$

$$-F_d^k + 0.08 \leq 2 \times (1 - y_d^c) \tag{5.15}$$

$$y_d^c \in \{0, 1\} \tag{5.16}$$

$$G_d^k \leq 7 \times y_d^g \tag{5.17}$$

$$-F_d^k + 0.08 \leq 7 \times (1 - y_d^g) \tag{5.18}$$

$$y_d^g \in \{0, 1\} \tag{5.19}$$

More specifically, when $F_d^k < 0.08$, y_d^c and y_d^g must be zero (Constraints (5.15) and (5.18)) and consequently, C_d^k and G_d^k must be zero (Constraints (5.14) and (5.17)). In addition, when C_d^k and G_d^k equal zero, Constraint (4.12) forces F_d^k to be zero. On the other hand, when $F_d^k \geq 0.08$, y_d^c and y_d^g can be either zero or one as long as Constraint (4.12) is satisfied. In addition, C_d^k is bounded by its availability of two when y_d^c equals one. Similarly, when y_d^g equals one, G_d^k is bounded by seven.

Next, we consider the term (4.23), DCWASA's centrifuge cost $= 196 \sum_{d=1}^{|D|} \lceil \frac{B_d F_d^{dc}}{50} \rceil$ in the total cost objective function (4.19). In Chapter 4, we use the term $\lceil \frac{B_d F_d^{dc}}{50} \rceil$ to represent the number of DCWASA centrifuges needed to dewater assigned biosolids.

Now, that F_d^{dc} is determined endogenously, the ceiling operator $\lceil \, \rceil$ is no longer applicable and we need a new variable to represent the number of DCWASA centrifuges in service. Defining C_d^{dc} as the number of DCWASA centrifuge in service on day d, the following constraints determine C_d^{dc}.

$$\forall d \in D;$$

$$C_d^{dc} \geq \frac{B_d F_d^{dc}}{50} \tag{5.20}$$

$$C_d^{dc} \leq 7 \tag{5.21}$$

$$C_d^{dc} \in \mathbb{Z}_+ \tag{5.22}$$

The number 50 in the denominator of (5.20) represents the capacity in DTS of one centrifuge and, given that C_d^{dc} is an integer, (5.20)–(5.22) find the minimum number of centrifuges needed. In addition, since we are minimizing the total cost, C_d^{dc} will always be the smallest number that satisfies (5.20). Subsequently, 4.23 needs to be modified as follows:

$$\text{DCWASA's centrifuge cost} = 196 \sum_{d=1}^{|D|} C_d^{dc} \tag{5.23}$$

Next, let's consider (4.27),

$$\text{on-site contractor's dewatering cost} = \sum_{d=1}^{|D|} \begin{cases} 71.75 F_d^k B_d & \text{if } F_d^k B_d \leq 150 \\ 3225 + 50.25 F_d^k B_d & \text{if } F_d^k B_d > 150 \end{cases}.$$

Similarly to the previous case, $F_d^k B_d$ or the amount of biosolids assigned to the on-site contractor cannot be determined exogenously. Nevertheless, it can be seen that the on-site contractor's dewatering cost is a piecewise-linear function, which consequently can be determined endogenously by employing SOS2 variables. Considering

Figure 5.2 displaying the piecewise linear cost function, there are three breakpoints to be considered for the amount of biosolids assigned to the on-site contractor: 0, 150, and 320 DTS. Consequently, the three breakpoints for the dewatering cost are

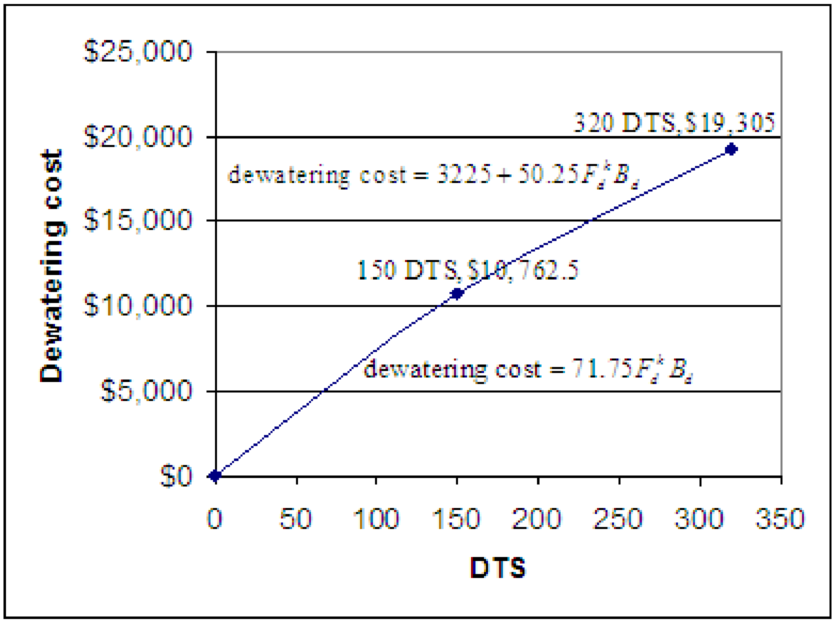


Figure 5.2: The on-site contractor piecewise linear dewatering cost function

$0, $10,762.5, and $19,305 determined from the corresponding linear cost functions. Next, considering SOS2 variables z_d^1, z_d^2, and z_d^3, the on-site contractor's dewatering cost can be determined as:

$$\text{on-site contractor's dewatering cost} = \sum_{d=1}^{|D|} 0z_d^1 + 10762.5z_d^2 + 19305z_d^3 \tag{5.24}$$

with the following constraints.

$$\forall d \in D;$$

$$B_d F_d^k = 0z_d^1 + 150z_d^2 + 320z_d^3 \tag{5.25}$$

$$z_d^1 \leq y_d^1 \tag{5.26}$$

$$z_d^2 \leq y_d^1 + y_d^2 \tag{5.27}$$

$$z_d^3 \leq y_d^2 \tag{5.28}$$

$$z_d^1 + z_d^2 + z_d^3 = 1 \tag{5.29}$$

$$y_d^1 + y_d^2 = 1 \tag{5.30}$$

$$y_d^1, y_d^2 \in \{0, 1\} \tag{5.31}$$

Now, the full optimization model with Schur's decomposition and SOS2 variables is as follows.

$$\begin{aligned} &\text{minimize} && \text{(total odor)} \\ &\text{minimize} && \text{(total cost)} \end{aligned} \tag{5.32}$$

$$\begin{aligned} \text{subject to} \quad & (4.1)\text{–}(4.19), (4.21), (4.24)\text{–}(4.26), (4.28), (4.30), \\ & (5.1)\text{–}(5.31) \\ & \text{and} \end{aligned}$$

$$\text{DCWASA's dewatering cost} = 70 \sum_{d=1}^{|D|} B_d F_d^{dc} \tag{5.33}$$

According to (5.33), it is shown that the DCWASA dewatering cost was set to \$70/DTS. This value corresponds to our discussion at the end of Chapter 4. In particular, we were interested in solving for optimal F_d^{dc} endogenously when the DCWASA dewatering cost equals \$70/DTS. Next, we applied the constraint method (see Section 2.1.1) to Problem (5.32) in order to find Pareto optimal points. The resulting optimization problem is:

$$\text{minimize} \quad \text{(total cost)} \tag{5.34}$$

$$\text{subject to} \quad (4.1)\text{–}(4.19), (4.21), (4.24)\text{–}(4.26), (4.28), (4.30),$$

$$(5.1)\text{–}(5.31), (5.33)$$

and

$$\sum_{d=1}^{|D|} O_d \leq \tau \tag{5.35}$$

It is noted here that the total cost objective function could also be used as an objective constraint and total odor could be the objective function to be minimized. However, based on our experience, when we minimized total cost and constrained total odor, the computational time was minimized.

Before solving Problem (5.34), we need to decide how many breakpoints $|I|$ and $|J|$ to use in order to approximate the bilinear term $F_d^{dc} L_d^{dc}$ (see (5.1)–(5.13)). Next, we analyze the number of breakpoints to use.

5.2.1 Analysis on Number of Breakpoints

Beale [4] has suggested using four to six points for each curve. However, more accuracy can also be achieved by reducing the spacing of the points with a significant increase in computing time [4, 33]. Moreover, after the first solution has been found, a more refined solution can also be achieved by redefining the breakpoints about the solution point and resolving [33]. Nevertheless, the number of breakpoints to use depends on the acceptable error and the function to be approximated [33].

As computing memory and solution algorithms have significantly improved since the work by Beale [4], more breakpoints could be employed than was initially suggested. Therefore, in order to find the right number of breakpoints, we solved a test problem. Unfortunately, the test could not be performed for our intended 31-day planning horizon as the compute time was extensive and will be described later in Section 5.2.2 (see Table 5.2) when we review the computational results. Thus, we ran the test problem for 1-day time horizon. In addition, we arbitrarily assigned $\tau = -3$. Table 5.1 presents the number of breakpoints tried, optimal solutions for L_d^{dc} and F_d^{dc}, approximated DCWASA's lime cost reported by the model, and DCWASA's lime cost calculated from the optimal L_d^{dc}. In particular, the approximated lime cost reported by the model was determined from (5.13); DCWASA's lime cost = $0.06 \sum_{d=1}^{|D|} B_d \left(\sum_{i=1}^{|I|} z_{id}^u (b_{id}^u)^2 - \sum_{j=1}^{|J|} z_{jd}^v (b_{jd}^v)^2 \right)$. On the other hand, given optimal solutions for L_d^{dc} and F_d^{dc}, denoted $\tilde{L}_d^{dc}$ and $\tilde{F}_d^{dc}$ respectively, the lime cost can be determined as DCWASA's lime cost = $\tilde{L}_d^{dc} \times \tilde{F}_d^{dc} \times B_d \times 0.06$.

Table 5.1: Number of breakpoints analysis

# Breakpoints	L_d^{dc}	F_d^{dc}	$\tilde{L}_d^{dc} \times \tilde{F}_d^{dc} \times B_d \times 0.06$	Approximated DCWASA's lime cost	Error (%)
5	310.4	0.92	5482.9056	5448.4320	0.63
10	310.4	0.92	5482.9056	5465.3440	0.32
15	310.4	0.92	5482.9056	5470.1760	0.23
20	310.4	0.92	5482.9056	5472.4648	0.19
25	310.4	0.92	5482.9056	5473.8000	0.17
30	310.4	0.92	5482.9056	5474.6748	0.15
40	310.4	0.92	5482.9056	5475.7514	0.13
50	310.4	0.92	5482.9056	5476.3886	0.12
60	310.4	0.92	5482.9056	5476.8098	0.11
70	310.4	0.92	5482.9056	5478.0467	0.09
80	310.4	0.92	5482.9056	5480.1845	0.05
90	310.4	0.92	5482.9056	5481.4364	0.03
100	310.4	0.92	5482.9056	5482.1813	0.01

Figure 5.3 depicts the percentage error as a function of the number of breakpoints used. It can be seen that the greater the number of breakpoints the more

accurate the approximation of DCWASA's lime cost. Moreover, when the number

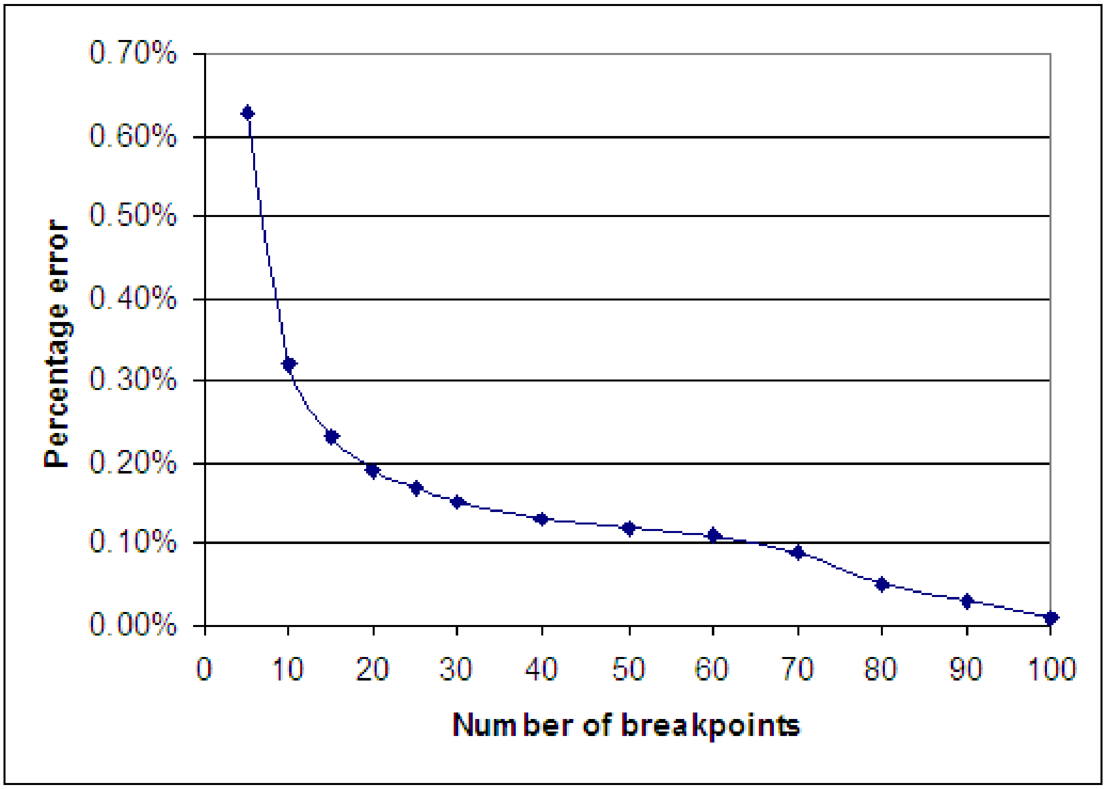


Figure 5.3: Percentage errors and number of breakpoints used

of breakpoints changed from five to ten the percentage error dropped by half and they gradually dropped for succeeding breakpoints. However, interestingly, all numbers of breakpoints yielded the same optimal solutions for L_d^{dc} and F_d^{dc} as indicated by Table 5.1. To this end, although the higher number of breakpoints may yield a better approximation of the term $L_d^{dc}F_d^{dc}$, the lower number of breakpoints may still yield the correct optimal solutions for L_d^{dc} and F_d^{dc}. Therefore, given the limited solution times and the reasons stated above, we chose 10 number of breakpoints to implement. In the next section, we present an algorithm to solve a series of the optimization problems and discuss the computational results.

5.2.2 Algorithm and Computational Results

In order to produce several Pareto optimal points, we need to solve a series of problems (5.34) by varying the τ value in (5.35) and use the criteria discussed in Section 2.1.1 to verify if the solution is Pareto optimal. There are many choices for τ values. The method by which we chose our τ values follows. Given the data for the 31 day-time horizon in question and the lower and upper bounds of variables, we substituted these data and corresponding bound in the odor prediction equation (3.2) as to obtain possible maximum and minimum odor value on each day. We then summe these maximum or minimum values to get the corresponding possible maximum and minimum total odor over the 31-day time horizon. Subsequently, the possible minimum and maximum total odor score was -85.856 and 216.084, respectively. The first problem similar to (5.34) was then set up with $\tau = -85.856$. The next problem was set up by increasing τ value by five each time until $\tau \geq 216.084$.

When we increase the τ value each time, the new problem contains a larger feasible region. Hence, the optimal total cost objective function value is expected to be smaller for the successor. Subsequently, when solving the next problem, we can inform the optimizer that the optimal objective function value should be at least as good as or smaller than the objective function value of the predecessor. This permits the optimizer to ignore solving any nodes that can only provide worse objective function values and therefore the run time is minimized [24]. XPRESS-MP provides a control parameter called "MIPABSCUTOFF" to communicate such the information to the optimizer. Figure 5.4 displays a flowchart of the procedure used to solve the multiobjective optimization model for wastewater treatment plant (MOPWTP).

Using the above algorithm, we started our first problem with $\tau = -85.856$. However, we only continued for the next 12 values of τ since the optimal solutions were not promising after extensive run times. The first optimization problem with $\tau = -85.856$ yielded an integer solution with 0% optimality gap within 36 seconds. However, for the next 12 problems, no integer solutions with 0% optimality gap could be obtained within several time limits we tried. Table 5.2 displays the computational results.

Table 5.2: Computational results from Problem 5.34

τ	#solutions found	Best Cost Bound ($)	Best Cost Solution ($)	Run Time (sec)	Optimality Gap%
-85.856	2	3,939,520	3,939,592	36	0
-80.856	4	3,799,486	3,932,085	12,992	3.49
-75.856	5	3,577,028	3,927,536	43,233	9.80
-70.856	3	3,642,585	3,926,077	43,233	7.78
-65.856	3	3,506,823	3,925,113	43,228	11.93
-60.856	7	3,602,905	3,921,547	43,236	8.84
-55.856	0	3,557,231	N/A	43,200	N/A
-50.856	9	3,459,374	3,919,705	43,233	13.31
-45.856	6	3,489,627	3,917,437	43,228	12.26
-40.856	0	3,486,870	N/A	43,200	N/A
-35.856	6	3,504,889	3,913,754	43,154	11.67
-30.856	0	3,489,558	N/A	43,200	N/A
-25.856	6	3,466,085	3,909,966	7,232	12.81

Although extensive compute times ($\geq$ 12 hours for most problems) were allowed, Table 5.2 shows that only the first problem where $\tau = -85.856$ yielded a zero percent optimality gap. The other problems, with at least one integer solution

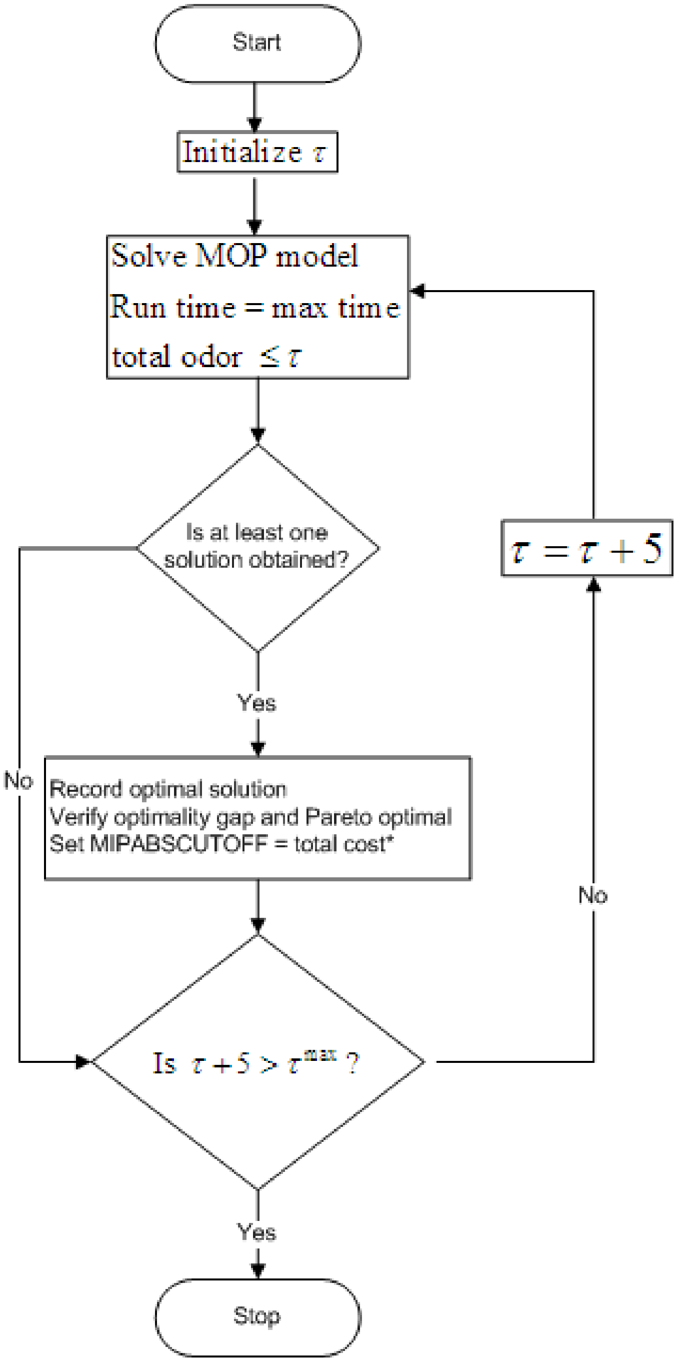

Start
Initialize τ
Solve MOP model
Run time = max time
total odor $\leq \tau$
Is at least one solution obtained?
$\tau = \tau + 5$
Yes
No
Record optimal solution
Verify optimality gap and Pareto optimal
Set MIPABSCUTOFF = total cost*
No
Is $\tau + 5 > \tau^{\max}$?
Yes
Stop

Figure 5.4: Algorithm to solve multiobjective program for wastewater treatment plant (MOPWTP)

found, yielded optimality gaps between 3.49% and 13.31%. Moreover, when $\tau = -55.856$, -40.856, and -30.856, the solver could not find any integer feasible solutions within the given time limits. As our illustrative example, figure 5.5 displays the progress of the branch and bound search when solving Problem (5.34) with $\tau = -60.856$. In addition, Figure 5.6 depicts the number of integer solutions found

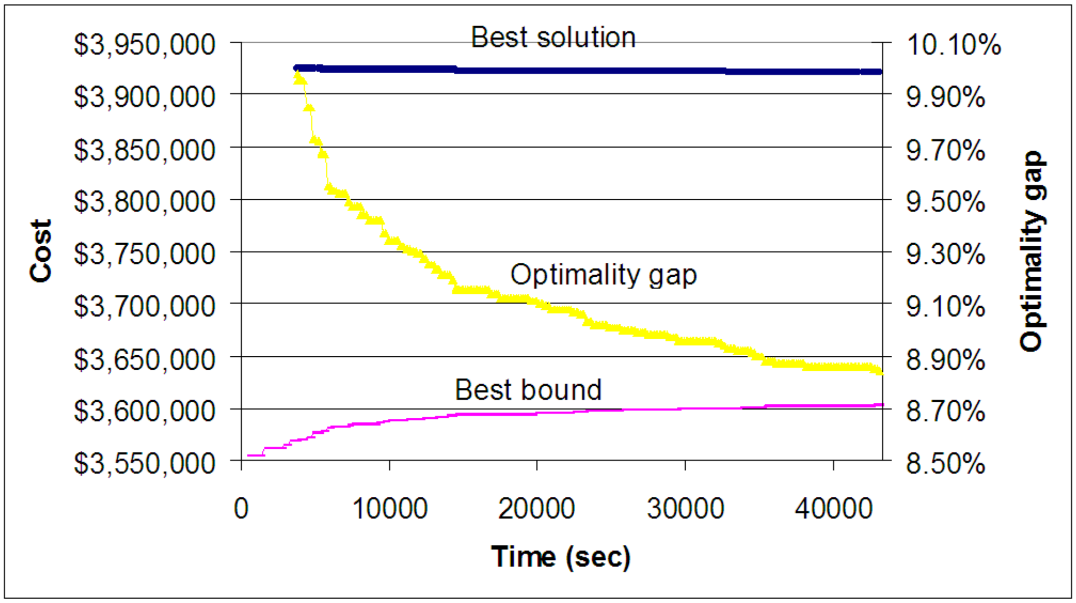


Figure 5.5: Best bound, best solution, and optimality gap for Problem (5.34) with $\tau = -60.856$

and optimality gaps for the 13 runs we have tried.

How large an optimality gap that is acceptable is subjective and depends on many factors (e.g., level of accuracy needed, type of application, and time permitted). One of the criteria we can use to decide what level of accuracy we need is the type and/or order of magnitude of the objective function value. For example, if the objective is to minimize the chance of failures of a space shuttle, then the optimality gap should be zero or near zero. On the other hand, if we want to minimize the cost

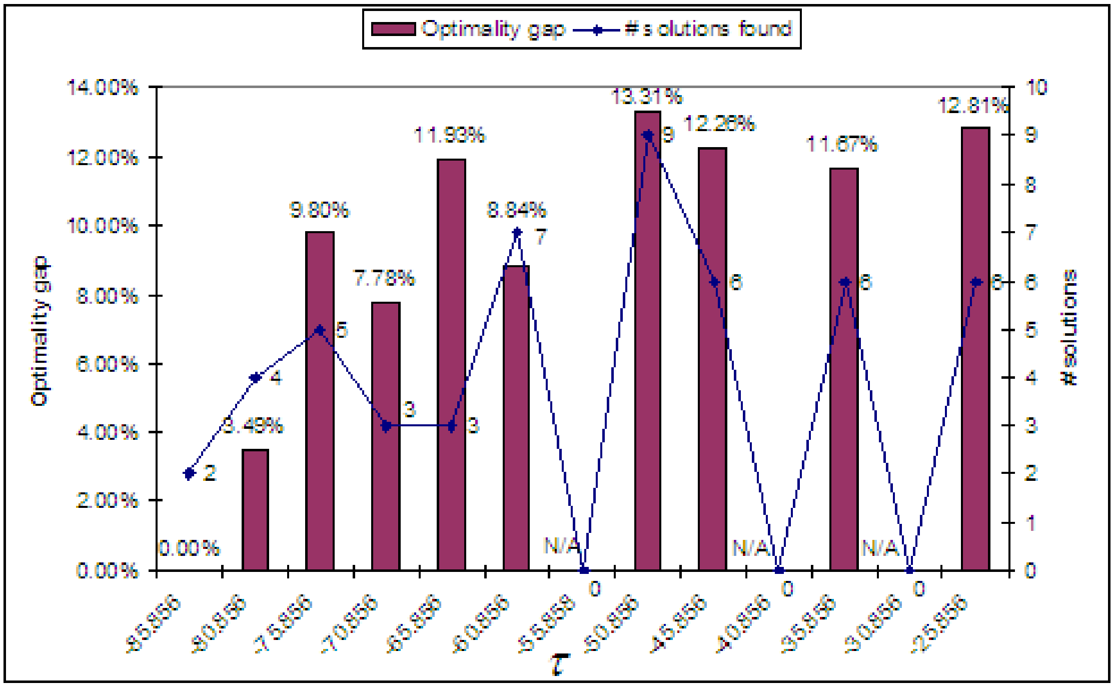


Figure 5.6: Number of solutions found and optimality gap for Problem (5.34) with $\tau = -85.856$–25.856

of a \$3,000,000 construction project, being able to obtain optimal solutions within 0.1% optimality gap means that the solutions can be off only \$3,000 and is probably acceptable. Now, let's consider the 13 problems we have tried. Table 5.3 indicates the possible amount of dollars off from the optimal solutions given the obtained optimality gaps and 0.1% assumed optimality gaps. Excluding the problem with $\tau = -85.856$, we see that the optimality gap in dollars was between \$132,600 and \$460,332. While, given the assumed 0.1% optimality gap, the possible amount of dollars off could have been between \$3,459 and \$3799, significantly less than the previous values. Nevertheless, these numbers are determined according to a worst-case scenario. In other words, it is also possible that the integer solutions obtained were already within less than the reported optimality gap, given that the better bounds

Table 5.3: Possible amount of dollars off from optimal solutions given optimality gaps

τ	Best bound ($)	Obtained optimality gap	Dollars off ($)	Dollar off (if 0.1% optimality gap, $)
−85.856	3,939,520	0	0	N/A
−80.856	3,799,486	3.49	132,600	3,799
−75.856	3,577,028	9.8	350,508	3,577
−70.856	3,642,585	7.78	283,492	3,643
−65.856	3,506,823	11.93	418,290	3,507
−60.856	3,602,905	8.84	318,642	3,603
−55.856	3,557,231	N/A	N/A	3,557
−50.856	3,459,374	13.31	460,332	3,459
−45.856	3,489,627	12.26	427,811	3,490
−40.856	3,486,870	N/A	N/A	3,487
−35.856	3,504,889	11.67	408,866	3,505
−30.856	3,489,558	N/A	N/A	3,490
−25.856	3,466,085	12.81	443,881	3,466

could be obtained.

Given the previous discussions, there are two choices to make at this point. First, we permit more computing time in order to get better integer solutions and better bounds and consequently optimality gaps are reduced. Second, we algorithmically find better bounds using the Dantzig-Wolfe decomposition algorithm discussed in Section 2.2 and numerically prove that the obtained integer solutions already yielded lower optimality gaps than the reported ones. Figure 5.7 displays the improvements in optimality gaps versus run times for selected problems. It can be seen that after allowing more time, the improvements in optimality gaps for all problems were very gradual. Moreover, all optimality gaps were more than 7.7% in spite of extensive computing times of 43,200 seconds. Therefore, the second choice

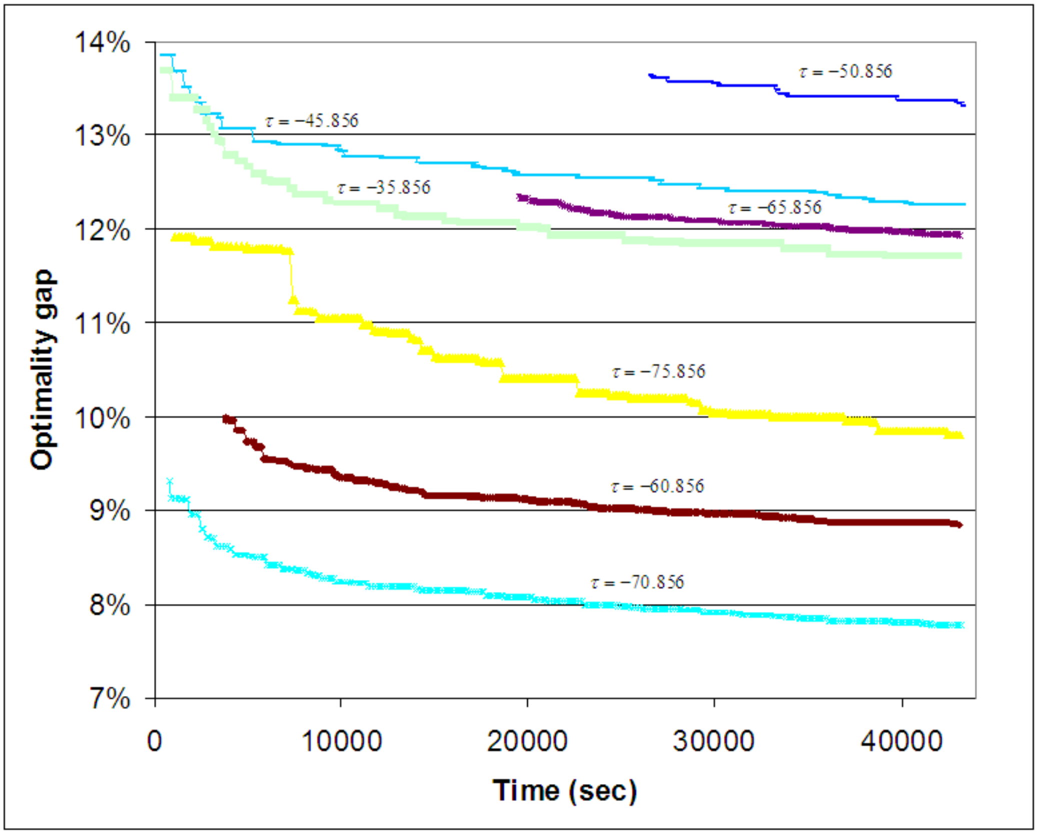


Figure 5.7: Optimality gap progressions for selected problems

of algorithmically finding better bounds for each problem is preferable to the first one. To this end, within this section, the only optimal solution obtained was when $\tau = -85.856$ and the corresponding total cost and total odor scores were \$3,939,520 and -85.856, respectively. It can be seen that the total odor $= \tau = -85.856$ and, according to Section 2.1.1 discussing the verification of Pareto optimal solution, this solution is also Pareto optimal.

In the next section, we apply the Dantzig-Wolfe decomposition technique discussed in Section 2.2 and are able to verify that several obtained integer solutions are already within 0.1% optimality gap.

5.3 Dantzig-Wolfe Decomposition Algorithm as Applying to Multi-objective Optimization Model for Distributing Biosolids to Reuse Fields

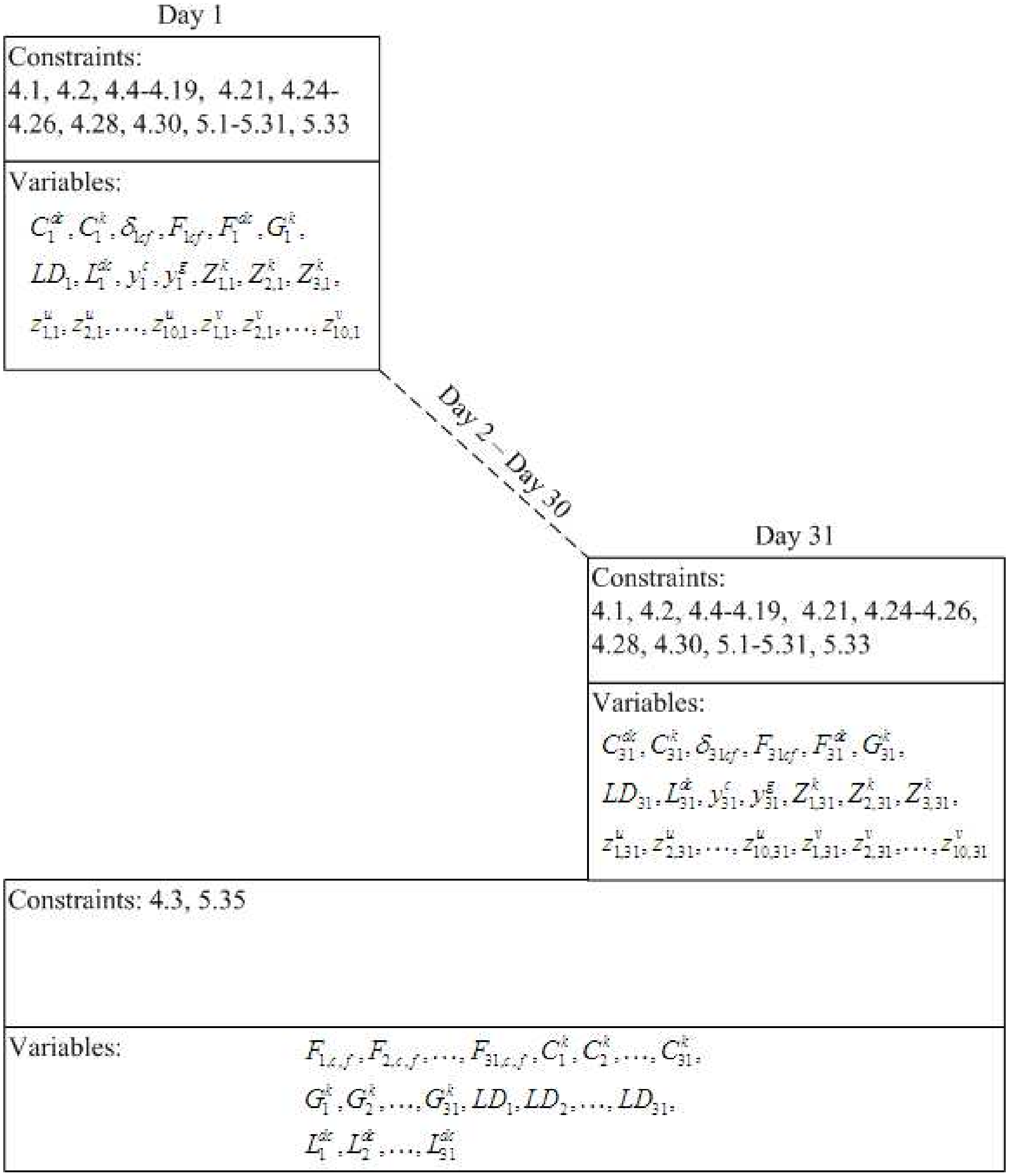
First, we present here the structure of our optimization problem and how it can be decomposed into several small problems. Figure 5.8 indicates if constraints (4.3) and (5.35) are dropped then problem 5.34 can be solved separately by day. In particular, constraints (4.3)[1] and (5.35)[2] are the joint constraints and will be included in the master problem. Other constraints shown in the daily block will be included in the pricing subproblems. Next, we give notation to some variables that

[1] The amount of biosolids distributed over the time horizon to each field cannot exceed the field's given capacity (in tons)

[2] Summation of biosolids odor score over the time horizon cannot exceed the given specified value

Minimize (total cost) (5.34)

Subject to



Day 1

Constraints:
4.1, 4.2, 4.4-4.19, 4.21, 4.24-4.26, 4.28, 4.30, 5.1-5.31, 5.33

Variables:
$C_1^{dk}, C_1^{k}, \delta_{1kf}, F_{1kf}, F_1^{dk}, G_1^{k},$
$LD_1, L_1^{dk}, y_1^{c}, y_1^{\xi}, Z_{1,1}^{k}, Z_{2,1}^{k}, Z_{3,1}^{k},$
$z_{1,1}^{u}, z_{2,1}^{u}, \ldots, z_{10,1}^{u}, z_{1,1}^{v}, z_{2,1}^{v}, \ldots, z_{10,1}^{v}$

Day 2 – Day 30

Day 31

Constraints:
4.1, 4.2, 4.4-4.19, 4.21, 4.24-4.26, 4.28, 4.30, 5.1-5.31, 5.33

Variables:
$C_{31}^{dk}, C_{31}^{k}, \delta_{31kf}, F_{31kf}, F_{31}^{dk}, G_{31}^{k},$
$LD_{31}, L_{31}^{dk}, y_{31}^{c}, y_{31}^{\xi}, Z_{1,31}^{k}, Z_{2,31}^{k}, Z_{3,31}^{k},$
$z_{1,31}^{u}, z_{2,31}^{u}, \ldots, z_{10,31}^{u}, z_{1,31}^{v}, z_{2,31}^{v}, \ldots, z_{10,31}^{v}$

Constraints: 4.3, 5.35

Variables:
$F_{1,c,f}, F_{2,c,f}, \ldots, F_{31,c,f}, C_1^{k}, C_2^{k}, \ldots, C_{31}^{k},$
$G_1^{k}, G_2^{k}, \ldots, G_{31}^{k}, LD_1, LD_2, \ldots, LD_{31},$
$L_1^{dk}, L_2^{dk}, \ldots, L_{31}^{dk}$

Figure 5.8: Structure of MOPWTP problem presented by day

have not yet been defined and the mathematical formulations for master problem and pricing subproblems follow.

Notations

F_{dcfi} = Amount of biosolids applied to field f by contractor c proposed by pricing subproblem d, proposal i

O_{di} = Biosolids odor proposed by pricing subproblem d, proposal i

total cost$_{di}$ = Optimal objective function value proposed by pricing subproblem d, proposal i

u_{di} = Weight variable corresponding to pricing subproblem d, proposal i

λ_f = Dual variable value corresponding to (5.37)

σ = Dual variable values corresponding to (5.38)

γ_d = Dual variable value corresponding to (5.39)

Master Problem:

$$\text{minimize} \sum_{d=1}^{|D|}\sum_{i=1}^{|I^d|} (\text{total cost})_{di} u_{di} \tag{5.36}$$

subject to

$$\forall f \in F; \sum_{d=1}^{|D|}\sum_{c=1}^{|C|}\sum_{i=1}^{|I^d|} F_{dcfi} u_{di} \le F_f^{\text{up}} \tag{5.37}$$

$$\sum_{d=1}^{|D|}\sum_{i=1}^{|I^d|} O_{di} u_{di} \le \tau \tag{5.38}$$

$$\forall d \in D; \sum_{i=1}^{|I^d|} u_{di} = 1 \tag{5.39}$$

Pricing Subproblem

$$\text{minimize} \qquad (\text{total cost}) - \sum_{c=1}^{|C|}\sum_{f=1}^{|F|} \lambda_f F_{dcf} - \sigma O_d - \gamma_d \tag{5.40}$$

$$\text{subject to} \qquad (4.1)\text{–}(4.2), (4.4)\text{–}(4.19), (4.21), (4.24)\text{–}(4.26), (4.28), (4.30),$$

$$(5.1)\text{–}(5.31), (5.33)$$

It is noted here that the subscript d in objective function or constraints of problem (5.40) represents only a single day in question. In other words, if the pricing subproblem being solved is for day 1, then data and variables correspond to day 1 only; for example, $O_d = O_1, \gamma_d = \gamma_1, F_{dcf} = F_{1cf}$.

Having shown the master problem and pricing subproblem, next we apply the DWIP algorithm as discussed in Section 2.2 and present the results.

5.3.1 Dantzig-Wolfe Decomposition Run Results

In this section, we set up 62 problems each by varying the maximum τ values for Problem (5.34). We then applied the Dantzig-Wolfe algorithm to these problems. In general, if the optimal solution vector $\tilde{\lambda} = (\tilde{\lambda}^1, \ldots, \tilde{\lambda}^K)$ of any LPM is not integer, then the original Problem (5.34) corresponding to that LPM is not yet solved [70]. Nevertheless, the optimal objective function values obtained from the Dantzig-Wolfe algorithm can be utilized as the lower bounds to their original problems.

The 62 τ values were picked as follows. We took the possible maximum and minimum total odor values found earlier in Section 5.2.2. The first problem similar to (5.34) was then set up with $\tau = -85.856$. The next problem was set up by increasing τ value by five each time until $\tau \geq 216.084$. Following this process, 62 problems were set up and the DWIP algorithm was applied to these problems. It turned out that none of these problems was able to be solved. That is, for every problem, the optimal solution vector $\tilde{\lambda} = (\tilde{\lambda}^1, \ldots, \tilde{\lambda}^K)$ was not integer when the DWIP algorithm terminated. Hence, the objective function values obtained from the DWIP algorithm were recorded and later used as lower bounds for the optimal objective function values of the original problems (see Table 5.4). We see that the maximum run time out of 62 problems was 11,673 seconds or approximately 3.24 hours and this was the case when $\tau = -75.856$. In addition, this problem also required the maximum number of iterations (32 iterations in total starting from Phase 1 until Phase 4 of DWIP algorithm). The second and third longest run times needed occurred when $\tau = -80.856$ and -85.856, where the run times were 7008 (≈ 1.95 hours) and 3226 seconds (≈ 53.77 minutes), respectively and the number of iterations needed were 26 and 30 iterations. The remaining problems required roughly the same amount of time with the average run time of 14 minutes. In addition, all of them required three iterations until the algorithm terminated.

As we mentioned earlier that the objective function values obtained from this section serve as lower bounds for Problem 5.34, in the next section, we numerically prove that the solutions obtained from Section 5.2.2, where shown to have large optimality gaps, are already within less than 0.1% optimality gap or lower.

Table 5.4: Objective function values obtained from DWIP algorithm

τ	Total Cost ($)	Run Time (Sec)	#iterations
-85.856	3,939,592.21698	3,226	30
-80.856	3,931,409.81972	7,008	26
-75.856	3,927,053.31700	11,673	32
-70.856	3,924,209.43764	851	3
-65.856	3,921,616.4313	841	3
-60.856	3,919,299.94231	849	3
-55.856	3,917,172.28273	848	3
-50.856	3,915,050.96558	837	3
-45.856	3,912,994.79164	855	3
-40.856	3,911,644.78919	863	3
-35.856	3,910,402.66125	848	3
-30.856	3,909,057.21065	872	3
-25.856	3,908,024.85254	848	3
-20.856	3,906,790.21157	856	3
-15.856	3,905,709.05505	858	3
-10.856	3,904,682.21824	857	3
-5.856	3,903,695.62302	853	3
-0.856	3,903,336.61651	853	3
4.144	3,902,564.07543	856	3
9.144	3,901,999.36590	863	3
14.144	3,901,423.17583	854	3
19.144	3,900,920.05560	858	3
24.144	3,900,416.77554	856	3
29.144	3,899,920.39914	864	3
34.144	3,899,420.05093	861	3
39.144	3,898,922.71977	883	3
44.144	3,898,433.83673	865	3
49.144	3,897,930.51021	877	3
54.144	3,897,484.99628	862	3
59.144	3,897,277.23006	865	3
64.144	3,896,990.69094	858	3
69.144	3,896,606.04516	864	3
74.144	3,896,409.92344	863	3
79.144, 84.144–114.144	3,896,339.11127	860–876	3
119.144	3,896,328.49690	859	3
124.144	3,896,326.29318	862	3
129.144, 134.144–214.144	3,896,326.26337	863–886	3
219.144	N/A (Problem is infeasible)		

5.4 Modified Optimality Gaps with Updated Lower Bounds

In this section, we revisit the results from Section 5.2.2. In particular, the objective function values obtained from Dantzig-Wolfe decomposition algorithm in the previous section will be used as the updated lower bounds to the solutions obtained from Section 5.2.2. Table 5.5 indicates the optimality gap when the new lower bounds are utilized. In addition, for the sake of comparison, we also include the original lower bounds and optimality gaps. Table 5.5 indicates that all modified optimality gaps are significantly lower than the original optimality gaps. In addition, most of the objective function values have less than a 0.1% optimality gap with the exception when $\tau = -50.856$ and -45.856. For all objective function values with optimality gaps less than or equal to 0.1%, we then compare their corresponding total odor and τ value in order to verify if they are Pareto optimal. More specifically, for each problem, if total odor $= \tau$ then the

Table 5.5: Modified bounds and optimality gaps

τ	MOPWTP algorithm				DWIP algorithm		Pareto optimal
	Total odor	Best bound ($)	Best solution ($)	Optimality gap (%)	Modified bound ($)	Optimality gap (%)	
-80.856	-80.856	3,799,486	3,932,085	3.49	3,931,410	0.02	yes
-75.856	-75.856	3,577,028	3,927,536	9.8	3,927,053	0.01	yes
-70.856	-70.856	3,642,585	3,926,077	7.78	3,924,209	0.05	yes
-65.856	-65.856	3,506,823	3,925,113	11.93	3,921,616	0.09	yes
-60.856	-60.998	3,602,905	3,921,547	8.84	3,919,300	0.06	no
-50.856	-50.856	3,459,374	3,919,705	13.31	3,915,051	0.12	no
-45.856	-45.864	3,489,627	3,917,437	12.26	3,912,995	0.11	no
-35.856	-36.778	3,504,889	3,913,754	11.67	3,910,403	0.09	no
-25.856	-25.856	3,466,085	3,909,966	12.81	3,908,025	0.05	yes

solution is Pareto optimal. To this end, Table 5.5 indicates five more Pareto optimal points; (−80.856, \$3,932,085.22101), (−75.856, \$3,927,535.95305), (−70.856, \$3,926,077.47746), (−65.856, \$3,925,113.48639), and (−25.856, \$3,909,966.37768) corresponding to the problem where $\tau = -80.856, -75.856, -70.856, -65.856$, and -25.856, respectively. Including the one Pareto optimal point obtained in Section 5.2.2, Figure 5.9 displays six Pareto optimal points we have obtained so far.

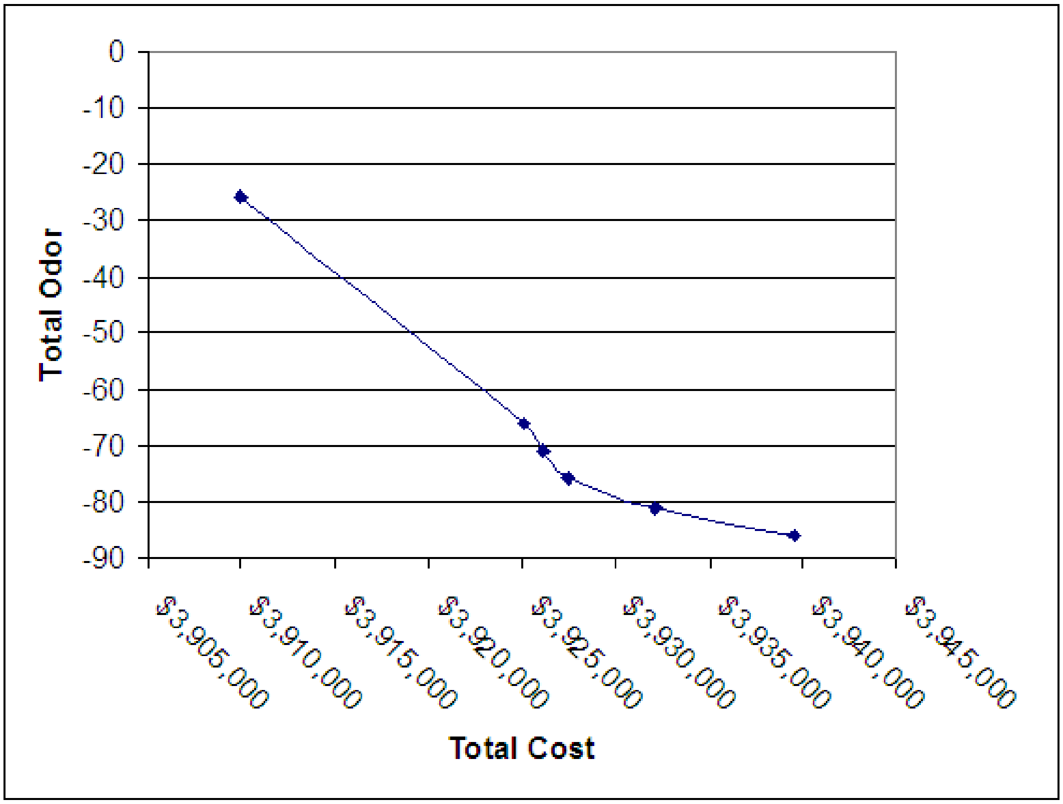


Figure 5.9: Pareto optimal points

Our current approach require three steps. First, in Section 5.2.2 we solved the original problems without the Dantzig-Wolfe decomposition. Although we allowed more than 12 hours of computing time for most problems, only one problem converged with 0.1% optimality gap. Second, in Section 5.3.1 we applied the Dantzig-

Wolfe decomposition to the original problem and obtained better lower bounds. Third, in this section, we numerically proved that the solutions obtained from Section 5.2.2 were in fact already within 0% optimality gap. Finally, we were able to verify five more Pareto optimal points. Although most of the problems required only 14 minutes in solution times for the second step, the first step required substantially large solution times of 12 hours for most problems. Consequently, with our current approach, extensive computing times are still required. Therefore, in the next section, the lower bounds obtained from DWIP algorithm are incorporated into the MOPWTP algorithm in order to reduce the overall solution times.

5.5 Modified Multiobjective Optimization Model for Wastewater Treatment Plant (MMOPWTP) Algorithm and Computational Results

In this section, we incorporate the total cost objective function values obtained from DWIP algorithm into the MOPWTP algorithm as shown in Figure 5.10. In particular, we solve a series of our multiobjective optimization problems as follows:

1. Initialize the τ value and the lower bound[3] associated to the τ value, denoted ρ^{τ}, obtained from Section 5.3.1.
2. Solve Problem 5.34 with the following setup. Set total odor $\leq \tau$ and MAXTIME= $T^{\max}$. MAXTIME is a parameter in XPRESS-MP used to set maximum al-

[3] Or the optimal objective function value obtain from DWIP algorithm

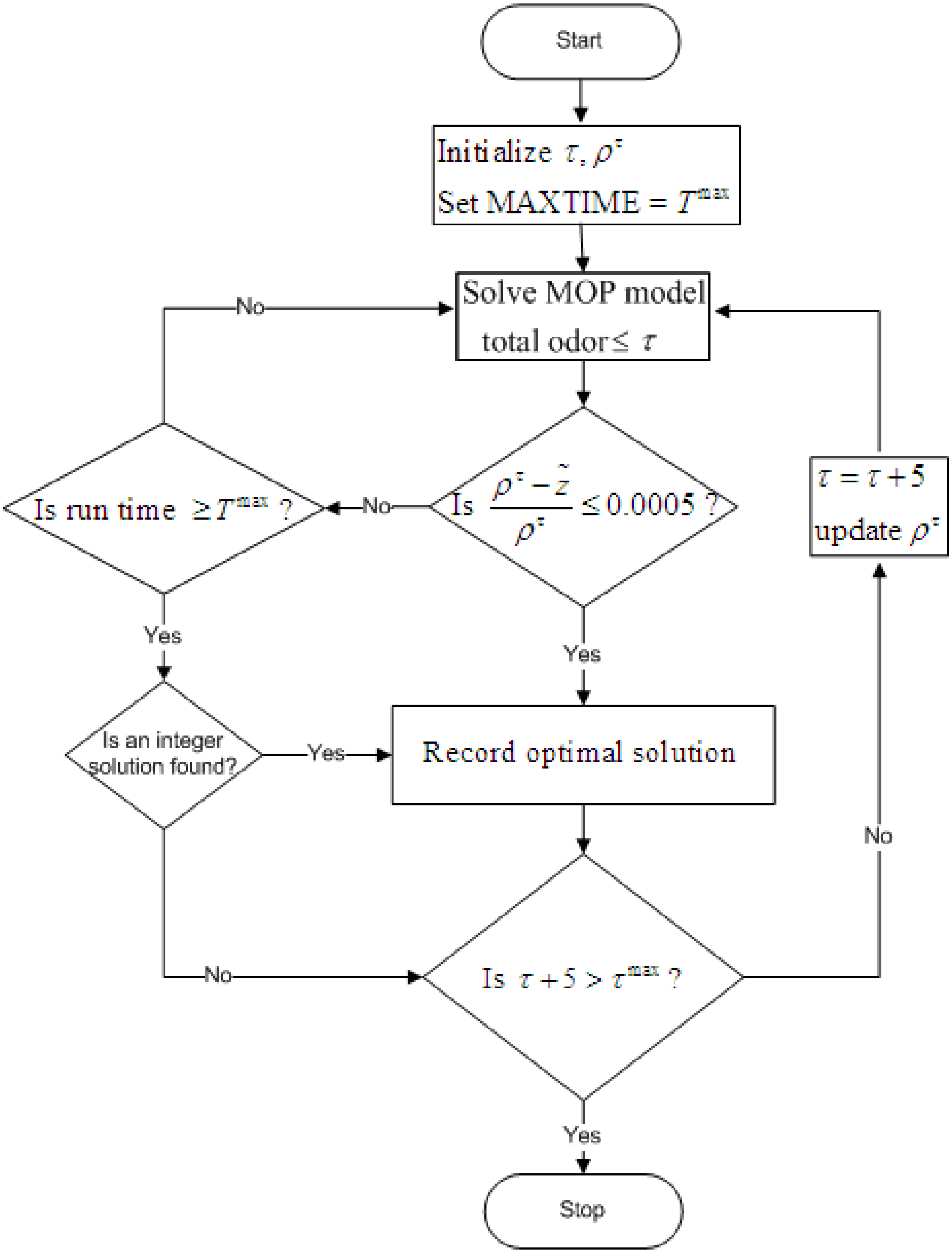

Start
Initialize τ, ρ^{τ}
Set MAXTIME = T^{max}
Solve MOP model
total odor $\leq \tau$
No
Is run time $\geq T^{max}$?
No
Is $\frac{\rho^{\tau} - \tilde{z}}{\rho^{\tau}} \leq 0.0005$?
$\tau = \tau + 5$
update ρ^{τ}
Yes
Yes
Is an integer solution found?
Yes
Record optimal solution
No
No
Is $\tau + 5 > \tau^{max}$?
Yes
Stop

Figure 5.10: Modified multiobjective optimization model for wastewater treatment plant algorithm

lowable run time. The optimizer will terminate if it does not converge within this preset maximum run time and then the program continues to the next applicable command.

3. If an integer solution is found, calculate its objective function value, denoted $\tilde{z}$. If $\frac{\rho^\tau - \tilde{z}}{\rho^\tau} \leq 0.0005$, go to 6. Otherwise, continue to 4.

4. If run time $\geq T^{\max}$, go to 5 . Otherwise, go to 2.

5. If at least one integer solution has been found, go to 6. Otherwise, go to 7.

6. Record optimal solution.

7. If $\tau + 5 > \tau^{\max}$, terminate model run. Otherwise, set $\tau \leftarrow \tau + 5$, update ρ^τ and go to 2.

This modified algorithm can reduce the overall solution time by allowing for better lower bounds resulting in faster convergence.

5.5.1 Computational Results

In this section, we utilize the objective function values obtained from Section 5.3.1. In particular, each of them was assigned to its associated ρ^τ value in MMOPWTP algorithm presented above. However, we do not need to resolve the problem where $\tau = -85.856 - -25.856$, since they were already solved in Section 5.2.2. In addition, their optimality gaps were already recalculated in Section 5.4, where the lower bounds obtained from Dantzig-Wolfe algorithm were already utilized. Thus, in this section, we only need to perform the MMOPWTP algorithm for the problems where $\tau = -20.856$ – 214.144. In addition, we also include three more reported total odor values: −60.998, −45.864, and −36.778 from Table 5.5. Each of these reported total odor values corresponds to an integer solution and, according to our computational experience, there is a chance that they are Pareto optimal. Therefore, we used these reported total odor values as the τ values and needed to find their corresponding lower bounds. As a result, we applied the DWIP algorithm to Problem 5.34 with $\tau = -60.998$, −45.864, and −36.778. Table 5.6 displays the corresponding results. Next, the MMOPWTP algorithm was implemented with τ = −60.998, −45.864, −36.778, and −20.856 – 214.144. In particular, for the τ values form −20.856 – 214.144, there was an increment of five as indicated by the algorithm. On the other hand, when we ran the algorithm for the τ values equal −60.998, −45.864, −36.778, and −20.856, Step 7 was omitted and each of the τ values was individually assigned. Table 5.7 and Table 5.8 displays the results for all τ values.

Table 5.6: Additional objective function values obtained from DWIP algorithm

τ	Total Cost ($)	Run Time (Sec)	#iterations
−60.998	$3,919,360.35933	123,393	31
−45.864	$3,913,311.52200	7,643	3
−36.778	$3,910,998.02508	7,792	3

Table 5.7: Best bounds, modified bounds, best solutions, and run time from MMOPWTP algorithm

τ	Best bound ($)	Modified bound ($)	Best solution ($)	Run time (sec)
−60.998	3,5809,968	3,9193,360	3,9213,317	29,692

Continued on next page

Table 5.7 – continued from previous page

τ	Best bound ($)	Modified bound ($)	Best solution ($)	Run time (sec)
−45.864	3,4843,323	3,9133,312	3,9169,977	36,033
−36.778	3,5046,662	3,9109,998	3,9130,039	36,033
−20.856	3,4626,689	3,9067,790	3,9089,940	7,234
−15.856	3,4358,883	3,9057,709	3,9083,352	7,228
−10.856	3,4804,484	3,9046,682	3,9082,203	7,229
−5.856	3,4267,761	3,9036,696	3,9075,588	7,234
−0.856	3,4466,650	3,9033,337	3,9082,212	7,229
4.144	3,4325,541	3,9025,564	3,9069,987	7,229
9.144	3,4223,331	3,9019,999	3,9074,421	7,235
14.144	3,4382,281	3,9014,423	3,9068,863	7,234
19.144	3,4392,261	3,9009,920	3,9087,732	7,228
24.144	3,4353,342	3,9004,417	3,9069,909	7,230
29.144	3,4619,938	3,8999,920	3,9077,709	7,228
34.144	3,4431,112	3,8994,420	3,9079,935	7,229
39.144	3,4420,043	3,8989,923	3,9070,012	7,229
44.144	3,4426,677	3,8984,434	3,9080,039	7,230
49.144	3,4313,387	3,8979,931	3,9081,158	7,228
54.144	3,4449,965	3,8974,485	3,9085,566	7,228
59.144	3,4202,281	3,8972,277	3,9095,550	7,233
64.144	3,4346,659	3,8969,991	3,9165,590	7,229
69.144	3,4316,621	3,8966,606	3,9086,614	7,231
74.144	3,4377,746	3,8964,410	3,9070,048	7,232
79.144	3,4452,258	3,8963,339	3,9074,490	7,228
84.144	3,4636,678	3,8963,339	3,9089,906	7,233
89.144	3,4613,313	3,8963,339	3,9088,854	7,232
94.144	3,4730,014	3,8963,339	3,9082,261	7,228
99.144	3,4409,957	3,8963,339	3,9102,238	7,229
104.144	3,4510,056	3,8963,339	3,9091,178	7,228
109.144	3,4379,962	3,8963,339	3,9078,879	7,228
114.144	3,4411,151	3,8963,339	3,9079,903	7,231
119.144	3,4294,427	3,8963,328	3,9082,284	7,230
124.144	3,4536,647	3,8963,326	3,9098,838	7,229
129.144	3,4325,548	3,8963,326	3,9092,217	7,229
134.144	3,4508,802	3,8963,326	3,9085,503	7,228
139.144	3,4505,592	3,8963,326	3,9082,264	7,229
144.144	3,4344,492	3,8963,326	3,9113,366	7,236
149.144	3,4730,013	3,8963,326	3,9075,534	7,232
154.144	3,4489,938	3,8963,326	3,9088,818	7,228
159.144	3,4722,261	3,8963,326	3,9084,403	7,237
164.144	3,4548,830	3,8963,326	3,9105,554	7,236
169.144	3,4322,260	3,8963,326	3,9054,462	7,231

Continued on next page

Table 5.7 – continued from previous page

τ	Best bound ($)	Modified bound ($)	Best solution ($)	Run time (sec)
174.144	3,4826,628	3,8963,326	3,9069,927	7,233
179.144	3,4407,784	3,8963,326	3,9134,459	7,236
184.144	3,4588,861	3,8963,326	3,9079,920	7,230
189.144	3,4729,981	3,8963,326	3,9088,845	7,230
194.144	3,4383,322	3,8963,326	3,9137,772	7,229
199.144	3,4725,501	3,8963,326	3,9078,819	7,228
204.144	3,4879,989	3,8963,326	3,9094,451	7,234
209.144	3,4503,377	3,8963,326	3,9079,994	7,229
214.144	3,4989,932	3,8963,326	3,9078,833	7,230

Table 5.8: Solver and modified optimality gaps from MMOPWTP algorithm

τ	Total odor	Solver optimality gap (%)	Modified optimality gap (%)	Pareto optimal
−60.998	−60.998	9.5	0.05	yes
−45.864	−46.183	12.42	0.09	no
−36.778	−36.778	11.65	0.05	yes
−20.856	−21.232	12.89	0.06	no
−15.856	−16.415	13.75	0.07	no
−10.856	−11.063	12.29	0.09	no
−5.856	−6.942	14.03	0.1	no
−0.856	−1.583	13.39	0.12	no
4.144	4.041	13.82	0.11	no
9.144	8.495	14.17	0.14	no
14.144	13.567	13.63	0.14	no
19.144	17.7	13.65	0.2	no
24.144	23.183	13.73	0.17	no
29.144	28.687	12.88	0.2	no
34.144	33.774	13.5	0.22	no
39.144	38.387	13.51	0.21	no
44.144	38.556	13.52	0.25	no
49.144	46.461	13.89	0.26	no
54.144	1.098	13.46	0.28	no
59.144	46.763	14.3	0.31	no
64.144	15.65	14.03	0.5	no
69.144	44.983	13.9	0.31	no
74.144	47.245	13.65	0.27	no

Continued on next page

Table 5.8 – continued from previous page

τ	Total odor	Solver optimality gap (%)	Modified optimality gap (%)	Pareto optimal
79.144	56.987	13.42	0.29	no
84.144	43.503	12.85	0.32	no
89.144	44.318	12.93	0.32	no
94.144	43.665	12.53	0.31	no
99.144	42.832	13.64	0.36	no
104.144	38.054	13.27	0.33	no
109.144	45.158	13.67	0.3	no
114.144	42.869	13.56	0.3	no
119.144	43.98	13.96	0.31	no
124.144	41.099	13.21	0.35	no
129.144	41.691	13.89	0.33	no
134.144	31.929	13.26	0.31	no
139.144	38.891	13.26	0.31	no
144.144	45.53	13.88	0.39	no
149.144	32.508	12.51	0.29	no
154.144	12.233	13.33	0.32	no
159.144	39.398	12.56	0.31	no
164.144	21.37	13.19	0.37	no
169.144	39.027	13.79	0.23	no
174.144	39.487	12.18	0.27	no
179.144	35.22	13.74	0.44	no
184.144	11.232	12.98	0.3	no
189.144	35.8	12.55	0.32	no
194.144	10.424	13.83	0.45	no
199.144	33.377	12.54	0.29	no
204.144	31.155	12.08	0.34	no
209.144	45.937	13.26	0.3	no
214.144	34.552	11.69	0.3	no

It is shown that the modified optimality gaps were significantly less than the optimality gap obtained internally by the solver. In addition, Table 5.8 indicates that two more Pareto optimal points were obtained. Therefore, including one Pareto optimal point obtained in Section 5.2.2, five Pareto optimal points obtained in Section 5.4 and finally two Pareto optimal points obtained in this section, we have obtained eight Pareto optimal points in total. Table 5.9 displays the eight Pareto optimal points we have obtained and Figure 5.11 depicts the Pareto optimal curve.

Next, we compare the Pareto optimal curve obtained in this section with the ones obtained in Chapter 4, where F_d^{dc} was exogenously determined.

Table 5.9: Eight Pareto optimal points

Total Odor	Total cost
-85.856	\$3,939,520
-80.856	\$3,932,085
-75.856	\$3,927,536
-70.856	\$3,926,077
-60.998	\$3,921,317
-65.856	\$3,925,113
-36.778	\$3,913,039
-25.856	\$3,909,966

5.6 Compare Results with Base Case

In Section 4.4.3: Sensitivity Analysis on DCWASA's Operating Costs, we found that when DCWASA's operating cost equaled \$70/DTS, the non-dominated Pareto optimal points correspond to the Pareto optimal points when $F_d^{dc} \in \{0.2, 0.3, 0.9\}$ as indicated in Figure 4.14. Nevertheless, for the F_d^{dc} we tried; 0.1, 0.2,. . .,0.9 were only a subset of all possible $0 \leq F_d^{dc} \leq 1$. In addition, the setup we had in Chapter 4 did not allow F_d^{dc} to vary across all days. In this chapter, F_d^{dc} was defined as a decision variable and was determined endogenously, however, with some additional constraints and computational difficulties we have previously discussed. Next, we compare the Pareto optimal points obtained in this chapter with the ones obtained

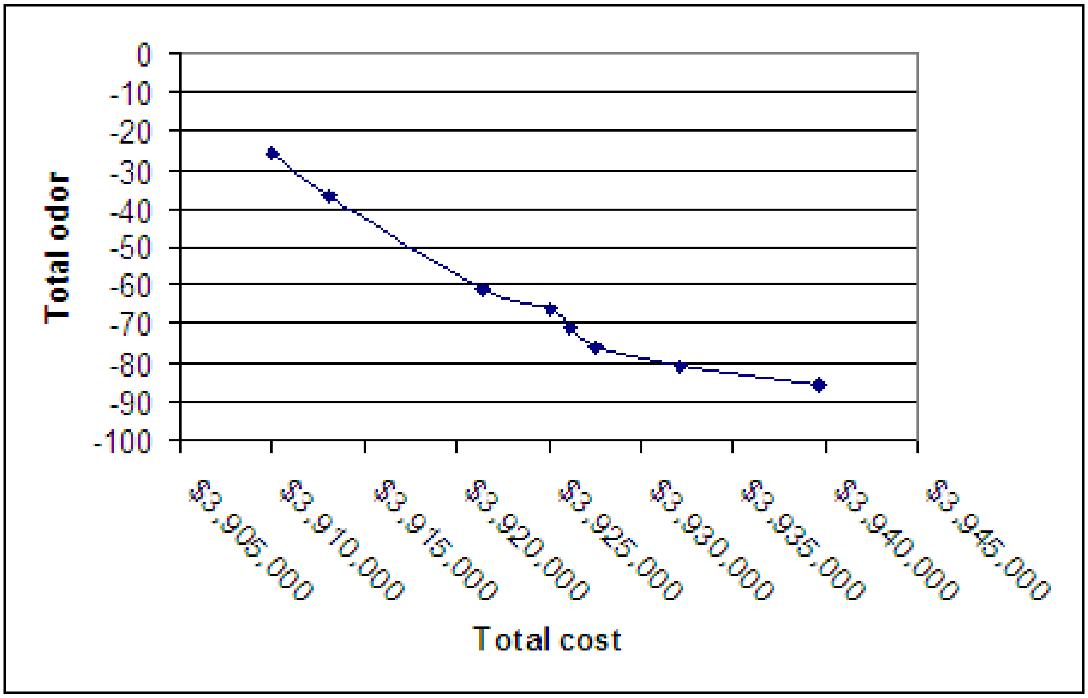


Figure 5.11: Eight Pareto optimal points

from Section 4.4.3.

Figure 5.12 indicates that the set of Pareto optimal points obtained when F_d^{dc} is determined endogenously dominates other sets of Pareto optimal points obtained exogenously in Chapter 4. Although, this is intuitive, the computational results are necessary to be carried out. As one of our major objectives in this chapter is to determine F_d^{dc} endogenously, our analysis on optimal solutions focuses on the optimal F_d^{dc} in comparison to ones exogenously determined in Chapter 4. Table 5.10 indicates that the optimal solutions for F_d^{dc} vary across all days for the time period in question. In addition, Figure 5.13 displays a bar chart summarizing the number of days for each value of F_d^{dc}.

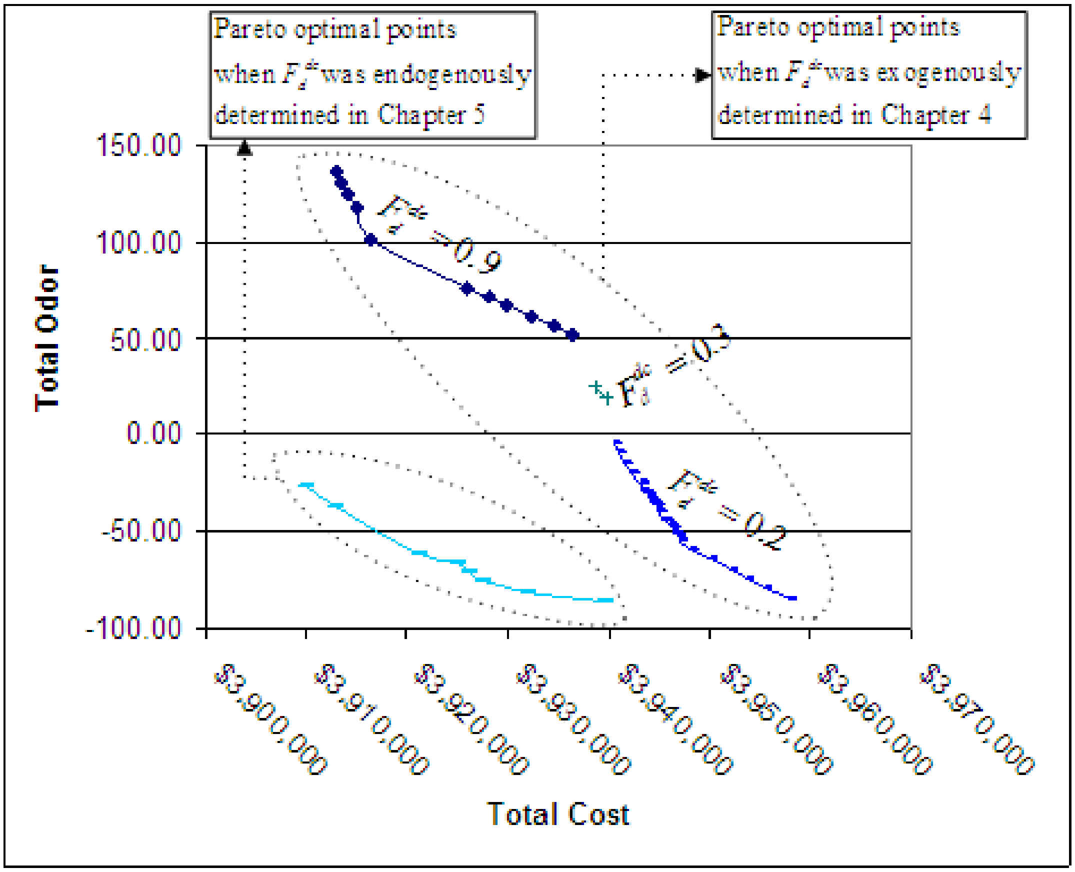


Figure 5.12: Comparison of Pareto optimal points when F_d^{dc} was determined endogenously and exogenously

Table 5.10: Daily optimal F_d^{dc} values for the eight Pareto optimal points

Day	F_d^{dc}							
	$\tau =$ -85.856	$\tau =$ -80.856	$\tau =$ -75.856	$\tau =$ -70.856	$\tau =$ -65.856	$\tau =$ -60.998	$\tau =$ -36.778	$\tau =$ -25.856
1	0.92	0.92	0.92	0.92	0.92	0.92	0.92	1
2	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
3	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
4	0.14	0.92	0.14	0.92	0.14	0.92	0.92	0.92
5	0.92	0.92	0.92	0.92	0.92	0.92	0.92	1
6	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
7	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
8	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
9	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
10	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
11	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
12	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
13	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
15	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
16	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
17	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
18	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
19	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
20	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
21	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
22	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
23	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
24	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
25	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
26	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
27	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
28	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
29	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
30	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
31	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14

It is shown that the optimal solutions to F_d^{dc} for all Pareto optimal points are in the set $\{1, 0.92, 0.14\}$. The choice of optimal F_d^{dc} relates directly to F_d^k as

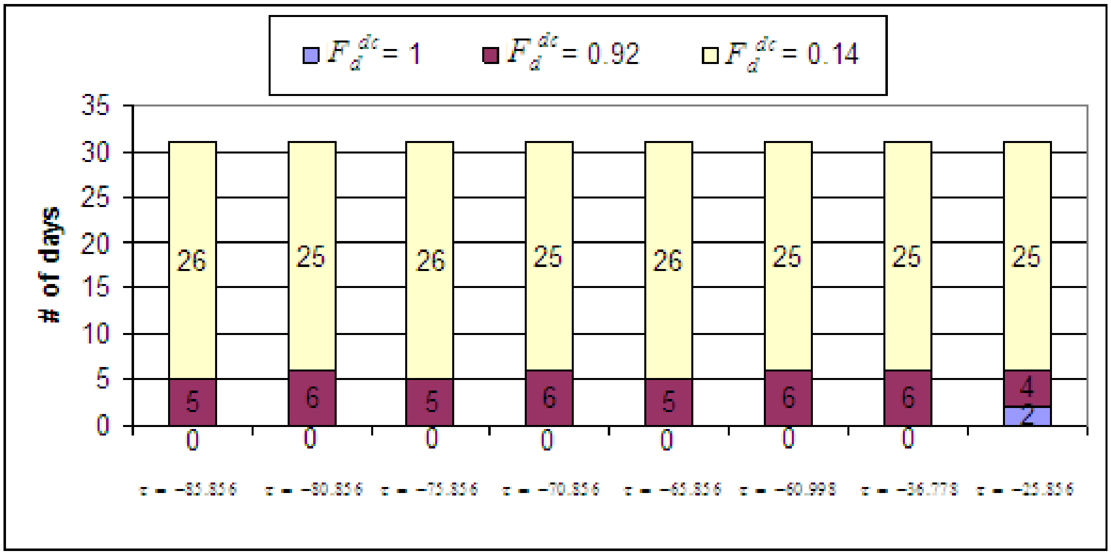


Figure 5.13: Optimal F_d^{dc} for eight Pareto optimal points

shown earlier in (4.1) where, on each day, the summation of flows to DCWASA and the on-site contractor must equal one ($\forall d \in D$; $F_d^{dc} + F_d^k = 1$). Moreover, F_d^k is constrained by (5.14) – (5.19) and (4.12), where F_d^k must be at least 0.08 or zero otherwise, as discussed earlier in Section (5.2). Thus, the choice of optimal F_d^{dc} is connected to (5.14) – (5.19) and (4.12) via (4.1). Consequently, in analyzing the optimal F_d^{dc} value, we examine optimal solutions to F_d^k, G^k and C^k, where G^k and C^k are two other decision variables appearing in (5.14) – (5.19) and (4.12). Table 5.11 displays optimal solutions to F_d^{dc}, F_d^k, G^k and C^k for the Pareto optimal point: (total odor $= -25.856$, total cost = \$3,909,966).

Table 5.11: Daily optimal solutions to F_d^{dc}, F_d^k, G_d^k, and C_d^k for Pareto optimal point: (total odor $= -25.856$, total cost = \$3,909,966)

Day	F_d^{dc}	F_d^k	G_d^k	C_d^k
1	1	0	0	0
2	0.92	0.08	0	2

Continued on next page

Table 5.11 – continued from previous page

Day	F_d^{dc}	F_d^k	G_d^k	C_d^k
3	0.92	0.08	0	1
4	0.92	0.08	0	2
5	1	0	0	0
6	0.14	0.86	7	2
7	0.14	0.86	7	2
8	0.14	0.86	7	2
9	0.14	0.86	7	2
10	0.14	0.86	7	2
11	0.14	0.86	7	2
12	0.14	0.86	7	2
13	0.14	0.86	7	2
14	0.14	0.86	7	2
15	0.14	0.86	7	2
16	0.14	0.86	7	2
17	0.14	0.86	7	2
18	0.14	0.86	7	2
19	0.14	0.86	7	2
20	0.14	0.86	7	2
21	0.14	0.86	7	2
22	0.14	0.86	7	2
23	0.14	0.86	7	2
24	0.14	0.86	7	2
25	0.14	0.86	7	2
26	0.14	0.86	7	2
27	0.14	0.86	7	2
28	0.14	0.86	7	2
29	0.14	0.86	7	2
30	0.92	0.08	0	2
31	0.14	0.86	7	2

We proceed by analyzing the optimal solutions to F_d^{dc} in three cases: $F_d^{dc} = 1$, $F_d^{dc} = 0.92$, and $F_d^{dc} = 0.14$.

First, we analyze the case when $F_d^{dc} = 1$. According to Table 5.11, it is shown that when both G_d^k and C_d^k equal zero, F_d^k equals zero and therefore F_d^{dc} equal one (see for example Day 1 and Day 5). In other words, when the on-site contractor belt filter presses and centrifuges were not needed in odor reduction, there would be no flow to the on-site contractor or $F_d^k = 0$ (cf. (4.12) and therefore F_d^{dc} must equal 1 (cf. (4.1)).

Second, the case when $F_d^{dc} = 0.92$ or, equivalently, $F_d^k = 0.08$ is analyzed. This case happened when only the on-site contractor centrifuges were needed for odor reduction. However, in order to have at least one centrifuge in service, at least eight percent of the flow must be assigned to the on-site contractor. These restrictions come from (5.14) – (5.19).

Lastly, the case where $F_d^{dc} = 0.14$ is analyzed. This case is equivalent to having $F_d^k = 0.86$ and can happen when all of the available on-site contractor belt filter presses and centrifuges were needed for odor reduction. Hence, the flow was assigned to the on-site contractor at their full capacity. In particular, seven belt filter presses at a capacity of 25 DTS/belt filter press and two centrifuges at a capacity of 50 DTS/centrifuge[4] can process $25 \times 7 + 50 \times 2 = 275$ DTS or $\frac{275}{320} = 86\%$ of biosolids per day.

5.7 Conclusion

In this chapter, we defined the percentage flow from the blend tank to DCWASA (F_d^{dc}) as decision variables. Therefore, F_d^{dc} were endogenously determined as opposed to being exogenously determined as explained in Chapter 4. However, the computational challenge was imposed by bilinear terms arising from the product of F_d^{dc} and

[4] See our assumption in Section 4.1: Problem Statement.

lime addition (L_d^{dc}). Later, we used Schur's decomposition and SOS2 variables to approximate the bilinear term. The resulting approximation problem became a linear integer program and was able to be solved with an existing optimization solver. However, with an extensive number of integer variables and constraints resulting from the approximation procedures, the approximation problem could not be solved to optimality within a reasonable computational time. This called for decomposition techniques. We have tried both Benders and Dantzig-Wolfe decomposition methods. However, only Dantzig-Wolfe decomposition technique proved to be useful. In particular, we successfully employed the Dantzig-Wolfe decomposition technique to significantly improve the lower bounds of the optimization problem. Finally, we significantly reduced the computing time by incorporating Dantzig-Wolfe bounds into our optimization model.

As F_d^{dc} were endogenously determined, a larger feasible region was available. As one could expect when the feasible region became larger, the Pareto optimal curve obtained in this chapter dominated the Pareto optimal curves obtained in Chapter 4. In addition, we found that the optimal percentage flows from blend tank to DCWASA were in the set $\{100\%, 92\%, 14\%\}$, where 14% selected the majority of the time. Nevertheless, it is very important to note that although most of biosolids were assigned to the on-site contractor, it did not necessarily imply that the on-site contractor would be more cost-efficient than DCWASA. In fact, it should be taken into account that the model was constructed from the odor prediction equation $O_d = 3.89 + .03T_d^{min} + .98B_d^e - .47G_d^k - 1.91C_d^k - .01L_d^{dc} + .56PD_d + 1.13LD_d$ (see equation (4.30)). It can be seen that besides DCWASA lime addition (L_d^{dc}), the on-site contractor belt filter presses and centrifuges (G_d^k and C_d^k) play very important roles in odor reduction. In addition, according to our analysis in Section 4.3.2, belt filter presses and centrifuges were preferable to lime additions most of the time. As (4.30) does not contain DCWASA centrifuge or belt filter press variables [5], when the model preferred belt filter presses or centrifuges over lime addition, more flows were thus assigned to the on-site contractor. To support our discussion, Figures 5.14 and 5.15 display the number of the on-site contractor centrifuges and belt filter presses needed, respectively, for each of the Pareto optimal points. It can be seen that when lower total odors were desired, a greater number of the on-site contractor centrifuges and belt filter presses were needed and thus more flows were assigned to the on-site contractor accordingly.

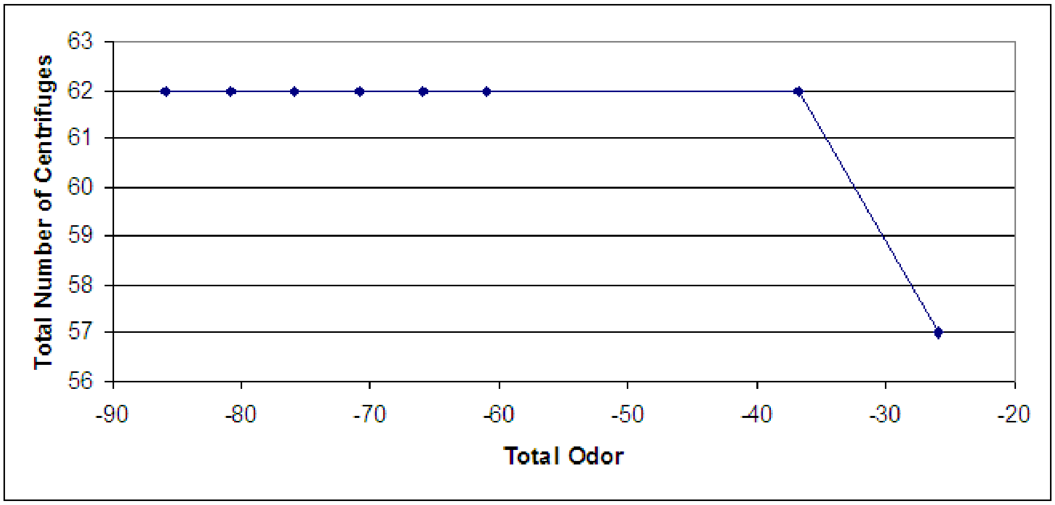


Figure 5.14: Total number of the on-site contractor centrifuges and total odor for the eight Pareto optimal points

[5] In fact, DCWASA does not operate belt filter presses.

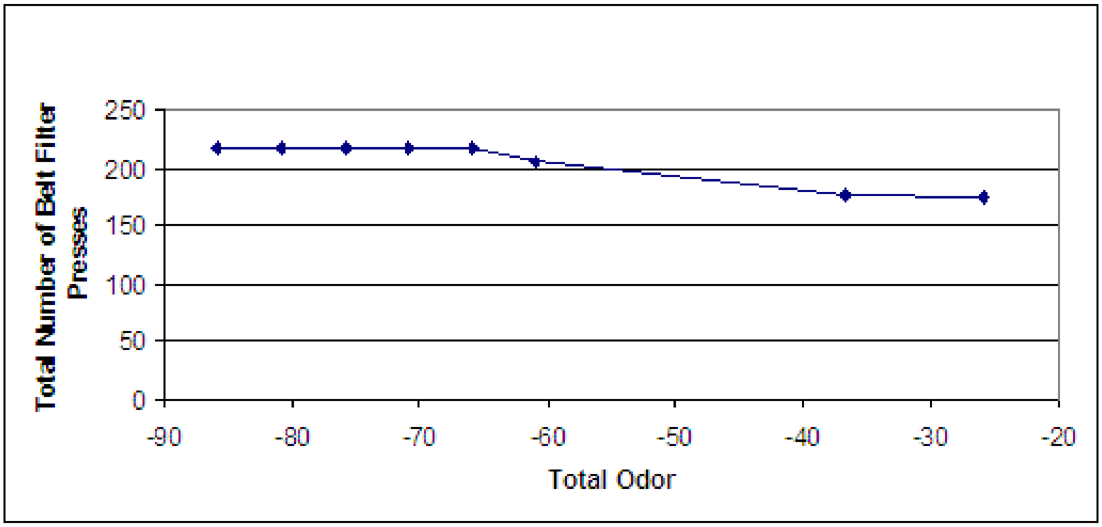


Figure 5.15: Total number of the on-site contractor belt filter presses and total odor for the eight Pareto optimal points

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this dissertation, we developed several novel multiobjective optimization models to simultaneously minimize biosolids odors as well as wastewater treatment process and biosolids distribution costs. With an increasing trend in biosolids land application in industry, we anticipate that typical wastewater treatment facilities could benefit from these models. In particular, the models can be used proactively to find optimal wastewater treatment process and biosolids distribution strategies in such a way that lessen odor impact to surrounding communities and is cost-effective at the same time. Some of our major findings and novel work are summarized below.

In Chapter 3, we developed novel odor threshold calculations to rank the odor sensitivities for reuse fields. (see Section 3.4). Later, we develop several optimization models, which may be divided into two major categories: one when the percentage flow from the blend tank to DCWASA (F_d^{dc}) was defined as a parameter or, in other words, exogenously determined; the other when F_d^{dc} was defined as a decision variable or endogenously determined. The former category was analyzed in Chapter 4, while, the latter was analyzed in Chapter 5.

In Chapter 4, we found that marginal activities (e.g., lime addition) needed to be considered in order to remain cost-effective when trying to reduce biosolids odor. Our analysis indicated that DCWASA's actual costs could be improved by using a Pareto optimal strategy by \$26,669 as well as reducing odors as shown in Figure 4.6. We found that when lower total odors were desired (consequently, higher total costs), biosolids would be hauled to reuse fields with generally lower associated odor thresholds. A sensitivity analysis on the percentage flows has shown that, given DCWASA operating costs = \$90 and \$80 per DTS, the Pareto optimal curve obtained when $F_d^{dc} \in 0.2$ dominated other Pareto optimal curve obtained when $F_d^{dc} \in \{0.3, \dots, 0.9\}$. However, when DCWASA operating cost = \$70/DTS, the Pareto optimal points were obtained when $F_d^{dc} \in \{0.2, 0.3, 0.9\}$. This led to the work in Chapter 5 where F_d^{dc} was endogenously determined.

In Chapter 5, we encountered computational challenges imposed by the bilinear terms arising from the product of F_d^{dc} and L_d^{dc}. Using Schur's decomposition [36, 29] and SOS2 variables [6, 5, 27], we approximated the bilinear terms. The resulting approximation problem became a linear integer program and was solvable with an existing optimization solver. However, with an extensive number of integer variables and constraints resulting from the approximation procedures, we were not able to solve the approximation problem to optimality within a reasonable computing time for a 31-day time horizon. Both Benders [34] and Dantzig-Wolfe [21] decomposition techniques were tried to decompose the computationally large optimization model. However, only Dantzig-Wolfe decomposition technique was proved to be useful for our optimization problem. In particular, we successfully employed Dantzig-Wolfe decomposition technique and significantly improved the lower bounds of the optimization problem. Finally, we considerably reduced the computing time by incorporating these bounds into our optimization model. Later, we found that the Pareto optimal points were obtained when $F_d^{dc} \in \{1.0, 0.92, 0.14\}$. In addition, the optimal solutions indicated that $F_d^{dc} = 0.14$ were optimal most of the time. Nevertheless, we concluded that the solution did not imply that the on-site contractor was more cost-effective than DCWASA. In fact, it'd rather be the case that the optimal solutions were influenced by the odor statistical equation used to predict biosolids odors in our optimization model.

Having mentioned that, one of limitations of our optimization models is that the applications of the models are specific to the odor prediction equations being used. Therefore, when more valid odor prediction equations are available, the optimization models need to be modified. Other limitations of the models include:

1. Since most of the odor reducing variables corresponded to the on-site contractor, the marginal activities were mostly associated to the on-site contractor. In particular, only one marginal activity (lime addition) was associated to DCWASA. While, two marginal activities (number of contractor centrifuges and belt filter presses) were corresponding to the on-site contractor.

2. We assumed that no storage of biosolids at the wastewater treatment facility was allowed. Therefore, when distribution of all biosolids from the plant is not possible, the developed optimization models are not valid

3. Our optimization models were constructed on the assumption that things are deterministic.

The limitations of our optimization models as well as other possible improvements lead to future work discussed in the next section.

6.2 Future Work

We summarize the list of potential future work as follows.

1. As mentioned above, other odor prediction equations containing a more completed set of wastewater treatment processing variables from both the on-site contractor and DCWASA should be tried as the input to the optimization model.

2. In Section 5.3, the Dantzig-Wolfe decomposition technique is applied to our problem and yields the solutions that are not integer. According to [70], this situation can be handled by the branch and price algorithm. Therefore, one should consider incorporating the branch and price algorithm into the MMOPWTP algorithm presented in Section 5.5.

3. Our optimization models should be modified such that they allowed storages of biosolids at the wastewater treatment facility. However, it is anticipated that the model will become much more complicated in the sense that biosolids odor would be changed over time. In addition, the mixture of biosolids produced from each day would require complicated odor prediction equations.

4. In most real-world problems, there are always some uncertainties corresponding to some parameters. The mathematical programming that takes into account these parameter uncertainties, where probability distributions of these

parameters are known, is called stochastic or probabilistic programming. As an extension to the deterministic models that we have done, the uncertainties of some parameters could be incorporated into the model (e.g., temperatures). Two well-known stochastic programming methods are two-stage programs and chance constraints [65].

5. In order to facilitate the prospective end-user (e.g., wastewater treatment facility) or the decision-maker, a friendly user interface to depict the optimization output could be designed. The interface can involve but not limited to the optimization package (e.g., XPRESS-MP), the database software (e.g., MS ACCESS), and the GIS software (ARCVIEW-GIS).

Appendix A

Avenue Script

Avenue script is an object-oriented programming language and development environment that is part of ArcView. There are many benefits for employing Avenue script in ArcView. It can be used to customize functions and tools in ArcView. A specific task performed by ArcView can also be guided by Avenue script. Moreover, Avenue script can be used to develop an application that is embedded ArcView's graphical user interface. In this thesis dissertation, we programmed Avenue script to determine numbers of people, schools, and hospitals and length of streets in specified areas. The script was embedded and run in ArcView GIS version 3.2a. Although a brief description of the purpose of each section of the codes was given, more specific meanings and uses of command lines are beyond the scope of this thesis dissertation. Readers with more interest in Avenue script should consult the help section: "Customizing and programming ArcView with Avenue" embedded in the ArcView application. The Avenue script used in this thesis dissertation is as follows:

```
    'create a new VTab to record total population from each field
myFileName = "c:\Prawat\ArcView3\OdorThresholdCalculation.dbf".AsFileName
popVTab = VTab.MakeNew(myFileName,dBase)
fld1 = Field.Make("FieldID",# FIELD_LONG, 15, 0)
```

```
fld2 = Field.Make(“TotalPop1”,#FIELD_DOUBLE, 15, 2)
fld3 = Field.Make(“TotalPop2”,#FIELD_DOUBLE, 15, 2)
fld4 = Field.Make(“TotalPop3”,#FIELD_DOUBLE, 15, 2)
fld5 = Field.Make(“PopCone1”,#FIELD_DOUBLE, 15, 2)
fld6 = Field.Make(“PopCone2”,#FIELD_DOUBLE, 15, 2)
fld7 = Field.Make(“PopCone3”,#FIELD_DOUBLE, 15, 2)
fld8 = Field.Make(“TotalSch1”,#FIELD_DOUBLE, 15, 2)
fld9 = Field.Make(“TotalSch2”,#FIELD_DOUBLE, 15, 2)
fld10 = Field.Make(“TotalSch3”,#FIELD_DOUBLE, 15, 2)
fld11 = Field.Make(“SchCone1”,#FIELD_DOUBLE, 15, 2)
fld12 = Field.Make(“SchCone2”,#FIELD_DOUBLE, 15, 2)
fld13 = Field.Make(“SchCone3”,#FIELD_DOUBLE, 15, 2)
fld14 = Field.Make(“TotalHos1”,#FIELD_DOUBLE, 15, 2)
fld15 = Field.Make(“TotalHos2”,#FIELD_DOUBLE, 15, 2)
fld16 = Field.Make(“TotalHos3”,#FIELD_DOUBLE, 15, 2)
fld17 = Field.Make(“HosCone1”,#FIELD_DOUBLE, 15, 2)
fld18 = Field.Make(“HosCone2”,#FIELD_DOUBLE, 15, 2)
fld19 = Field.Make(“HosCone3”,#FIELD_DOUBLE, 15, 2)
fld20 = Field.Make(“TotalLen1”,#FIELD_DOUBLE, 15, 2)
fld21 = Field.Make(“TotalLen2”,#FIELD_DOUBLE, 15, 2)
fld22 = Field.Make(“TotalLen3”,#FIELD_DOUBLE, 15, 2)
fld23 = Field.Make(“LenCone1”,#FIELD_DOUBLE, 15, 2)
fld24 = Field.Make(“LenCone2”,#FIELD_DOUBLE, 15, 2)
```

```
fld25 = Field.Make(“LenCone3”,#FIELD_DOUBLE, 15, 2)
popVTab.AddFields({fld1, fld2, fld3, fld4, fld5, fld6, fld7, fld8, fld9, fld10, fld11,
fld12, fld13, fld14, fld15, fld16, fld17, fld18, fld19, fld20, fld21, fld22, fld23, fld24,
fld25})
popVTab.SetEditable(True) ’get a document on which we will be working
projectedView = av.finddoc(“Projected”)
PopDen = projectedView.findtheme (“dc_va_md_blkgrp_
3milesbuffer_MD_NAD83_feet.shp”)
SchoolTheme = projectedView.findtheme (“School_3milesbuffer_
md_nad83_feet_nohistorical.shp”)
HosTheme = projectedView.findtheme (“DC_VA_MD_Hospital_
3milesBuffer_MD_NAD83_Feet.shp”)
StreetTheme = projectedView.findtheme (“dc_va_md_street_
3milesBuffer_MD_NAD83_Feet.shp”)
    ’get reuse field table and use getX and getY method to retrieve its x,y coordi-
nate in xy plane
ptTheme = projectedView.findtheme(“ReusefieldsJan05Oct06_MD_NAD83_Feet.shp”)
myPointFtab = ptTheme.getFtab
myFieldIDField = myPointFtab.FindField(“Fieldid”)
myShapeField = myPointFtab.FindField(“Shape”)
myWindField = myPointFtab.FindField(“Wind_Direc”)
pi = number.getpi
    ’loop through each record in reuse field table. For each loop, the following
```

```
tasks will be done
'1. Get x coordinate and y coordiate from each reuse field
'2. Create a plume from the obtained coordinate
'3. Select population density features that fall within a plume
'4. Export selected features to a new FTab
'5. Create a new FTheme from the new FTab
'6. Select all features of the new Ftheme
'7. Use “ClipSelected” method to clip the Ftheme with the plume polygon
'8. Save the new edited FTheme
'9. Loop through each feature of the edited FTheme
'9.1 Multiply population density with the area
'9.2 Total sum to get total population
'10. record each the total effected population for each reuse field to a VTab “popTable.dbf”
for each i in 1..myPointFtab.getnumrecords
'for each i in 536..536 generate new record for each reuse field
myNewRec = popVTab.AddRecord
    'get each record of point feature in Reusefieldsinlandonlyprojectednad1983.shp point theme
myPointName = myPointFtab.returnvalue(myFieldIDField,i-1)
myPoint = myPointFtab.returnvalue(myShapeField,i-1)
myWind = myPointFtab.returnvalue(myWindField,i-1).AsString
    'get the origin of each point (reuse field)
```

```
xpt = myPoint.getX
ypt = myPoint.getY
        'get location of the plume polygon
mya = 5280*3
myb = 5280
if (myWind = “W”) then
WindDir = 0
elseif (myWind = “WSW”) then
WindDir = pi/8
elseif (myWind = “SW”) then
WindDir = 2*pi/8
elseif (myWind = “SSW”) then
WindDir = 3*pi/8
elseif (myWind = “S”) then
WindDir = 4*pi/8
elseif (myWind = “SSE”) then
WindDir = 5*pi/8
elseif (myWind = “SE”) then
WindDir = 6*pi/8
elseif (myWind = “ESE”) then
WindDir = 7*pi/8
elseif (myWind = “E”) then
WindDir = 8*pi/8
```

```
elseif (myWind = “ENE”) then
WindDir = 9*pi/8
elseif (myWind = “NE”) then
WindDir = 10*pi/8
elseif (myWind = “NNE”) then
WindDir = 11*pi/8
elseif (myWind = “N”) then
WindDir = 12*pi/8
elseif (myWind = “NNW”) then
WindDir = 13*pi/8
elseif (myWind = “NW”) then
WindDir = 14*pi/8
elseif (myWind = “WNW”) then
WindDir = 15*pi/8
else
MsgBox.info(“Wind Direction at field ” + i.AsString + “ is not valid”,“”)
return nil
end
if ((WindDir ≥ 0) and (WindDir ≤ (pi/2))) then
theta = WindDir
alpha = (myb/mya).Atan
gamma = theta-alpha
myc = mya/alpha.cos
```

```
myd = myc*gamma.cos

mye = myc*gamma.sin

myf = myc*(theta+alpha).cos

myg = myc*(theta+alpha).sin

myh = mya*theta.cos

myi = mya*theta.sin

p1 = (xpt+myd)@(ypt+mye)

p2 = (xpt+myf)@(ypt+myg)

p3 = (xpt+myh)@(ypt+myi)

elseif ((WindDir > (pi/2)) and (WindDir ≤ pi)) then

theta = WindDir - (pi/2)

alpha = (myb/mya).Atan

gamma = theta-alpha

myc = mya/alpha.cos

myd = myc*gamma.sin

mye = myc*gamma.cos

myf = myc*(theta+alpha).sin

myg = myc*(theta+alpha).cos

myh = mya*theta.sin

myi = mya*theta.cos

p1 = (xpt-myd)@(ypt+mye)

p2 = (xpt-myf)@(ypt+myg)

p3 = (xpt-myh)@(ypt+myi)
```

```
elseif ((WindDir > pi) and (WindDir ≤ (3*pi/2))) then
theta = WindDir - pi
alpha = (myb/mya).Atan
gamma = theta-alpha
myc = mya/alpha.cos
myd = myc*gamma.cos
mye = myc*gamma.sin
myf = myc*(theta+alpha).cos
myg = myc*(theta+alpha).sin
myh = mya*theta.cos
myi = mya*theta.sin
p1 = (xpt-myd)@(ypt-mye)
p2 = (xpt-myf)@(ypt-myg)
p3 = (xpt-myh)@(ypt-myi)
elseif ((WindDir > (3*pi/2)) and (WindDir < (2*pi))) then
theta = WindDir - (3*pi/2)
alpha = (myb/mya).Atan
gamma = theta-alpha
myc = mya/alpha.cos
myd = myc*gamma.sin
mye = myc*gamma.cos
myf = myc*(theta+alpha).sin
myg = myc*(theta+alpha).cos
```

```
myh = mya*theta.sin
myi = mya*theta.cos
p1 = (xpt+myd)@(ypt-mye)
p2 = (xpt+myf)@(ypt-myg)
p3 = (xpt+myh)@(ypt-myi)
else
msgbox.info(“Wind Direction at field ” + i.AsString + “ is not valid”,“”,“”)
return nil
end
’make new polygon out of three points
pointlist1 = xpt@ypt,p1,p2
’make a circle polygon from each field location
conepoly = polygon.make(pointlist1)
’make center point of the circle
circlecenter = point.make(xpt,ypt)
r = 0
’what follows we loop three times, one for each radius
for each m in 1..3
r = r + 5280
myCirclePoly = circle.make(circlecenter, r).AsPolygon
        ’now we select Dc_va_md_nad1983_av32.shp Ftheme based on a circle polygon
’select features that intersect with the circle
PopDen.SelectByPolygon(myCirclePoly,#VTAB_SELTYPE_NEW)
```

```
PopDenTab = PopDen.GetFTab
    'make a new shape file out of selected popden features and then make it a theme
thePopDenCircle = “c:\Prawat\ArcView3\working\popDenCircle”+
myPointName.AsString+“_”+m.AsString+“.shp” 'export selected features to a new FTab
mySelectPopDen = PopDenTab.Export(thePopDenCircle.asFileName, Shape, TRUE)
myTheme = FTheme.Make(mySelectPopDen)
projectedView.AddTheme(myTheme)
projectedView.SetEditableTheme(myTheme)
    'select all features of mySelectPopDen theme
myTheme.SelectByPolygon(myCirclePoly,#VTAB_SELTYPE_NEW)
    'clip mySelectPopDen with myCirclePoly and save edit
myTheme.ClipSelected(myCirclePoly)
myTheme.StopEditing(True)
myTheme.ClearSelection
    'clear selection of PopDen
PopDen.ClearSelection
    '********************************************
'next we calculate number of people within r mile-radius
'loop through each feature in new edited mySelectPopden to calculate population
myThemeFTab = myTheme.GetFTab
myPopDenBlock = myThemeFTab.FindField(“Shape”)
```

```
myDenField = myThemeFTab.FindField(“Pop00_sqmi”)
myTotalPop=0
for each k in 1..myThemeFTab.GetNumRecords
myPolygon = myThemeFTab.ReturnValue(myPopDenBlock,k-1)
myArea = myPolygon.ReturnArea ’area in sq.ft.
myDen = myThemeFTab.ReturnValue(myDenField,k-1) ’population density in people/sq.mi.
myPop = myArea*myDen/27878400 ’27878400 is to convert the area in sq.ft. to sq.mi.
myTotalPop = myTotalPop + myPop
end
    ’next we add record total population into popVTab.dbf
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField(“FieldID”),myNewRec,myPointName)
popVTab.SetValue(popVTab.FindField(“TotalPop”+
m.AsString),myNewRec,myTotalPop)
else
MsgBox.info(“popVTab is not editable”,“Warning!”)
end
    ’************************************
’Next we count only people within r mile-radius and intersect with conepoly
’Next we select myTheme (edited thePopDenCircle) based on conepoly
thePopDenCone = “c:\Prawat\ArcView3\working\popDenCone”+
myPointName.AsString+“_”+m.AsString+“.shp”
myTheme.SelectByPolygon(conepoly,#VTAB_SELTYPE_NEW)
mySelectPopDenCone = myTheme.GetFTab.Export(
thePopDenCone.asFileName,shape,true)
myPopDenConeTheme = FTheme.Make(mySelectPopDenCone)
projectedView.AddTheme(myPopDenConeTheme)
projectedView.SetEditableTheme(myPopDenConeTheme)
myTheme.ClearSelection
        ’Select all features of myPopDenConeTheme
myPopDenConeTheme.SelectByPolygon(conepoly,#VTAB_SELTYPE_NEW)
        ’Clip myPopDenConeTheme with conepoly and save edit
myPopDenConeTheme.ClipSelected(conepoly)
myPopDenConeTheme.StopEditing(True)
myPopDenConeTheme.ClearSelection
        ’next we loop through each feature in a new cliped myPopDenConeTheme and
do a sum product of area and population density fields
myThemeFTab = myPopDenConeTheme.GetFTab
myPopDenBlock = myThemeFTab.FindField(“Shape”)
myDenField = myThemeFTab.FindField(“Pop00_sqmi”)
myTotalPopCone=0
for each j in 1..myThemeFTab.GetNumRecords
myPolygon = myThemeFTab.ReturnValue(myPopDenBlock,j-1)
myArea = myPolygon.ReturnArea ’area in sq.ft.
```

```
myDen = myThemeFTab.ReturnValue(myDenField,j-1) 'population density in people/sq.mi.
myPop = myArea*myDen/27878400 '27878400 is to convert the area in sq.ft. to sq.mi.
myTotalPopCone = myTotalPopCone + myPop
end
projectedView.DeleteTheme(myTheme)
projectedView.DeleteTheme(myPopDenConeTheme)
        'next we add record total population into popVTab.dbf
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField(“PopCone”+
m.AsString),myNewRec,myTotalPopCone)
else
MsgBox.info(“popVTab is not editable”,“Warning!”)
end
end
        'stopworking with population density theme here
'*************************************************************
'*************************************************************
        'Next we work with school theme
'make a circle polygon from the each field location
r = 0
'what follows we loop three times, one for each radius
```

```
for each m in 1..3
r = r + 5280
myCirclePoly = circle.make(circlecenter, r).AsPolygon 'make a circle
    'calculate total number of school within r feet radius
'now we select school theme based on a circle polygon
'select features that intersect with the plume
SchoolTheme.SelectByPolygon(myCirclePoly,#VTAB_SELTYPE_NEW)
SchoolFTab = SchoolTheme.GetFTab
    'make a new shape file out of selected school theme features and then make it
a theme
theSchoolCircle = “c:\Prawat\ArcView3\working\SchoolCircle”+
myPointName.AsString+“_”+m.AsString+“.shp” 'export selected features to a new
FTab
mySelectSchool = SchoolFTab.Export(theSchoolCircle.asFileName, Shape, TRUE)
myTheme = FTheme.Make(mySelectSchool)
projectedView.AddTheme(myTheme)
    'clear selection of SchoolTheme
SchoolTheme.ClearSelection
    '*****************************************************************
'next we calculate total school within r radius
myThemeFTab = myTheme.GetFTab
TotalSchool = myThemeFTab.GetNumRecords
    'next we record total school into popVTab.dbf
```

```
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField(“TotalSch”+m.AsString),myNewRec,TotalSchool)
else
MsgBox.info(“popVTab is not editable”,“Warning!”)
end
    ’***************************************************************
’Next we determine total school within r radius and intersect with conepoly
’Next we select myTheme (edited theSchoolCircle) based on conepoly
theSchoolCone = “c:\Prawat\ArcView3\working\SchoolCone”+
myPointName.AsString+“_”+m.AsString+“.shp”
myTheme.SelectByPolygon(conepoly,#VTAB_SELTYPE_NEW)
mySelectSchoolCone = myTheme.GetFTab.Export(theSchoolCone.asFileName,shape,true)
mySchoolConeTheme = FTheme.Make(mySelectSchoolCone)
projectedView.AddTheme(mySchoolConeTheme)
    ’next we calculate total school
myThemeFTab = mySchoolConeTheme.GetFTab
TotalSchool = myThemeFTab.GetNumRecords
    projectedView.DeleteTheme(myTheme)
projectedView.DeleteTheme(mySchoolConeTheme)
    ’next we record total school into popVTab.dbf
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField(“SchCone”+m.AsString),myNewRec,TotalSchool)
else
```

```
MsgBox.info(“popVTab is not editable”,“Warning!”)
end
end
’Stop working with school theme here
’*********************************************
’*********************************************
    ’next we work with hospital theme
’make a circle polygon from each field location
r = 0
’what follows we loop three times, one for each radius
for each m in 1..3
r = r + 5280
myCirclePoly = circle.make(circlecenter, r).AsPolygon ’make a circle
    ’calculate total number of hospital within r feet radius
’now we select hospital theme based on a circle polygon
HosTheme.SelectByPolygon(myCirclePoly,#VTAB_SELTYPE_NEW)
’select features that intersect with the plume
HosFTab = HosTheme.GetFTab
    ’make a new shape file out of selected hospital theme features and then make it a theme
theHosCircle = “c:\Prawat\ArcView3\working\HosCircle”+ myPointName.AsString+“_”+m.AsString+“.shp” ’export selected features to a new FTab
```

```
mySelectHos = HosFTab.Export(theHosCircle.asFileName, Shape, TRUE)
myTheme = FTheme.Make(mySelectHos)
projectedView.AddTheme(myTheme)
        'clear selection of HosTheme
HosTheme.ClearSelection
        '*******************************************************************
'next we calculate total hospital within r radius
myThemeFTab = myTheme.GetFTab
TotalHos = myThemeFTab.GetNumRecords
        'next we record total hospital into popVTab.dbf
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField("TotalHos"+m.AsString),myNewRec,TotalHos)
else
MsgBox.info("popVTab is not editable","Warning!")
end
        '******************************************************************
'Next we determine total hospital within r radius and intersect with conepoly
'Next we select myTheme (edited theHosCircle) based on conepoly
theHosCone = "c:\Prawat\ArcView3\working\HosCone"
+myPointName.AsString+"_"+m.AsString+".shp"
myTheme.SelectByPolygon(conepoly,#VTAB_SELTYPE_NEW)
mySelectHosCone = myTheme.GetFTab.Export(theHosCone.asFileName,shape,true)
myHosConeTheme = FTheme.Make(mySelectHosCone)
```

```
projectedView.AddTheme(myHosConeTheme)
    'next we calculate total hospital
myThemeFTab = myHosConeTheme.GetFTab
TotalHos = myThemeFTab.GetNumRecords
    projectedView.DeleteTheme(myTheme)
projectedView.DeleteTheme(myHosConeTheme)
    'next we record total hospital into popVTab.dbf
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField("HosCone"+m.AsString),myNewRec,TotalHos)
else
MsgBox.info("popVTab is not editable","Warning!")
end
end
    'Stop working with hospital theme here
'*********************************************************************
'*********************************************************************
    'next we work with street theme to get effected lenght of street in a 1,2,3 mile radius from each reuse field
'In addition we also calculate effected length of street in a 1,2,3 mile radius but within conepoly
'for each loop we calculate total length of street in a specify radius.
    'make a circle polygon from the each field location
r = 0
```

```
'what follows we loop three times, one for each radius
for each m in 1..3
r = r + 5280
myCirclePoly = circle.make(circlecenter, r).AsPolygon 'make a circle
    'calculate total length of street within r feet radius
'now we select street theme based on a circle polygon
'select features that intersect with the plume
StreetTheme.SelectByPolygon(myCirclePoly,#VTAB_SELTYPE_NEW)
StreetFTab = StreetTheme.GetFTab
    'make a new shape file out of selected street theme features and then make it
a theme
theStreetCircle = “c:\Prawat\ArcView3\working\StreetCircle”+
myPointName.AsString+“_”+m.AsString+“.shp” 'export selected features to a new
FTab
mySelectStreet = StreetFTab.Export(theStreetCircle.asFileName, Shape, TRUE)
myTheme = FTheme.Make(mySelectStreet)
projectedView.AddTheme(myTheme)
projectedView.SetEditableTheme(myTheme)
    'select all features of mySelectStreet theme
myTheme.SelectByPolygon(myCirclePoly,#VTAB_SELTYPE_NEW)
    'clip mySelectStreet with myCirclePoly and save edit
myTheme.ClipSelected(myCirclePoly)
myTheme.StopEditing(True)
myTheme.ClearSelection
     'clear selection of StreetTheme
StreetTheme.ClearSelection
     '****************************************************************
'next we calculate total lenght of street within r radius
'loop through each feature in new edited mySelectStreet to get street lenght
myThemeFTab = myTheme.GetFTab
myShape = myThemeFTab.FindField("Shape")
myTotalLength = 0
if (myThemeFTab.GetNumRecords = 0) then
myTotalLength = 0
else
for each k in 1..myThemeFTab.GetNumRecords
myPolyline = myThemeFTab.ReturnValue(myShape,k-1)
myLength = myPolyline.ReturnLength/5280 'lenght in mile.
myTotalLength = myTotalLength + myLength
end
end
'next we record total lenght into popVTab.dbf
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField("TotalLen"+m.AsString),
myNewRec,myTotalLength)
else
```

```
MsgBox.info(“popVTab is not editable”,“Warning!”)
end
    ’*************************************************************************
’Next we determine street lenght within r radius and intersect with conepoly
’Next we select myTheme (edited thePopDenCircle) based on conepoly
theStreetCone = “c:\Prawat\ArcView3\working\StreetCone”+
myPointName.AsString+“_”+m.AsString+“.shp”
myTheme.SelectByPolygon(conepoly,#VTAB_SELTYPE_NEW)
mySelectStreetCone= myTheme.GetFTab.Export (theStreetCone.asFileName,shape,true)
myStreetConeTheme = FTheme.Make(mySelectStreetCone)
projectedView.AddTheme(myStreetConeTheme)
projectedView.SetEditableTheme(myStreetConeTheme)
myTheme.ClearSelection
    ’Select all features of myStreetConeTheme
myStreetConeTheme.SelectByPolygon(conepoly,#VTAB_SELTYPE_NEW)
    ’Clip myStreetConeTheme with conepoly and save edit
myStreetConeTheme.ClipSelected(conepoly)
myStreetConeTheme.StopEditing(True)
myStreetConeTheme.ClearSelection
    ’next we loop through each feature in a new cliped myStreetConeTheme
myThemeFTab = myStreetConeTheme.GetFTab
myShape = myThemeFTab.FindField(“Shape”)
myTotalLength = 0
```

```
if (myThemeFTab.GetNumRecords = 0) then
myTotalLength = 0
else
for each k in 1..myThemeFTab.GetNumRecords
myPolyline = myThemeFTab.ReturnValue(myShape,k-1)
myLength = myPolyline.ReturnLength/5280 'lenght in mile.
myTotalLength = myTotalLength + myLength
end
end
projectedView.DeleteTheme(myTheme)
projectedView.DeleteTheme(myStreetConeTheme)
    'next we record total lenght into popVTab.dbf
if (popVTab.IsEditable) then
popVTab.SetValue(popVTab.FindField(“LenCone”+
m.AsString),myNewRec,myTotalLength)
else
MsgBox.info(“popVTab is not editable”,“Warning!”)
end
end
'Stop working with street theme here
'********************************************************
'********************************************************
end popVTab.SetEditable(False)
```

BIBLIOGRAPHY

[1] Arispe, S. (2005). *Measuring and Developing a Control Strategy for Odorous Gases from Solids Handling Processes of a Large Wastewater Treatment Plant.* M.S. Thesis, Department of Civil and Environmental Engineering, University of Maryland, College Park, MD.

[2] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-price: column generation for solving huge integer programs. *Operations Research* **46**(3): 316-329.

[3] Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. (1990). *Linear Programming and Network Flows* second ed. John Wiley & Sons: New York.

[4] Beale, E. M. L. (1968). *Mathematical Programming in Practice.* John Wiley & Sons: New York.

[5] Beale, E. M. L., and Forrest, J. J. H. (1976). Global optimization using special ordered sets. *Mathematical Programming* **10**:52-69.

[6] Beale, E. M. L., and Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In: Lawrence J. (ed). *Proceedings of the Fifth IFORS Conference.* Tavistock, London, pp 447-454.

[7] Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4**: 238-252.

[8] Bertsimas, D., and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization.* Athena Scientific: Belmont, MA.

[9] Bhattacharya, B., Lobbrecht, A. H., and Solomatine, D. P. (2003). Neural networks and reinforcement learning in control of water systems. *Journal of Water Resources Planning and Management* **129**(6): 458-465.

[10] Chang, N. B., Chen, H. W., Shaw, D. G., and Yang, C. H. (1997). Water pollution control in river basin by interactive fuzzy interval multiobjective programming. *Journal of Environmental Engineering* **123**(12): 1208-1216.

[11] Chang, N. B., Lu, H. Y., and Wei, Y. L. (1997). GIS technology for vehicle routing and scheduling in solid waste collection systems. *Journal of Environmental Engineering* **123**(9): 901-910.

[12] Chang, N. B., Schuler, R. E., and Shoemaker, C. A. (1993). Environmental and economic optimization of an integrated solid waste management system. *Journal of Resource Management & Technology* **21**(2): 87-100.

[13] Chang, N. B., and Wang, S. F. (1997). A fuzzy goal programming approach for the optimal planning of metropolitan solid waste management systems. *European Journal of Operational Research* **99**(2):303-321.

[14] Chang, N. B., Wen, C. G., and Chen, Y. L. (1997). A fuzzy multi-objective programming approach for optimal management of the reservoir watershed. *European Journal of Operational Research* **99**: 289-302.

[15] Chvatal, V. (1983). *Linear Programming.* W. H. Freeman and Company: New York\San Francisco.

[16] Ciric, A. R., and Huchette, S. G. (1993). Multiobjective optimization approach to sensitivity analysis: waste treatment costs in discrete process synthesis and optimization problems. *Industrial & Engineering Chemistry Research* **32**(11): 2636-2646.

[17] Cohon, J. L. (1978). *Multiobjective Programming and Planning.* Academic Press: New York.

[18] Colombani, Y., and Heipcke, S. (2005). *Multiple models and parallel solving with MOSEL.* Dash Optimization: UK.

[19] Conejo, A. J., Castillo, E., Minquez, R., and Bertrand, R. G. (2006). *Decomposition Techniques in Mathematical Programming, Engineering, and Science Applications.* Springer: Berlin/New York.

[20] Crohn, D. M., and Thomas, A. C. (1998). Mixed-integer programming approach for designing land application systems at a regional scale. *Journal of Environmental Engineering* **124**(2): 170-177

[21] Dantzig, G. B. (1998). *Linear Programming and Extensions.* Princeton University Press: Princeton, New Jercey.

[22] Dantzig, G. B., and Thapa, M. N., *Linear Programmin 1: Introduction* Springer, New York, 1997.

[23] Dantzig, G. B., and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research* **8**(1): 101-111.

[24] Dash Optimization *Xpress-Optimizer Reference Manual.* Retrieved August 8, 2007, from http://www.dashoptimization.com/home//secure/documentation/optimizer/optimizer.pdf.

[25] DCWASA *Biosolids Management Program.* Retrieved December 16, 2004, from http://www.dcwasa.com/education/biosolids-brochure.pdf.

[26] Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons: New York.

[27] Escudero, L. (1978). A comparative analysis of linear fitting for non-linear functions on optimization, a case study: air pollution problems. *European Journal of Operational Research* **2**: 398-408.

[28] Federal Register (1993). Standards for the use or disposal of sewage sludge: final rules. *Federal Register* **58**: 9248-9415.

[29] Gabriel, S. A., Bertrand, R. G., Sahakij, P., and Conejo, A. J. (2006). A practical approach to approximate bilinear functions in mathematical programming problems by using Schurs decomposition and SOS type 2 variables. *Journal of the Operational Research Society* **57**: 9951004.

[30] Gabriel, S. A., Sahakij, P., Ramirez, M., and Peot, C. (2007). A multiobjective optimization model for processing and distributing biosolids to reuse fields. *Journal of the Operational Research Society* **58**: 850-864.

[31] Gabriel, S. A., Vilalai, S., Arispe, S., Kim, H., McConnell, L. L., Torrents, A., Peot, C., and Ramirez, M. (2005). Prediction of dimethyl disulfide levels from biosolids using statistical modeling. *Journal of Environmental Science & Health* Part A: Toxic/Hazardous Substances & Environmental Engineering **40**(11): 2009-2025.

[32] Gabriel, S. A., Vilalai, S., Peot, C., and Ramirez, M. (2006). Statistical modeling to forecast odor levels of biosolids applied to reuse sites. *Journal of Environmental Engineering* **132**: 479-488.

[33] Gass, S. L. (1985). *Linear Programming: Methods and Applications*. McGraw-Hill: New York.

[34] Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications* **10**(4): 237-260.

[35] Heipcke, S. (2002) *Applications of optimization with XpressMP*. Dash Optimization Ltd.: Northants, UK.

[36] Horn, R. A., and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press: Cambridge.

[37] Intarakosit, E. (2006). *GIS-Based Odor Disperstion Modeling for Measuring the Effect of DCWASA Biosolids in Reuse Fields*. M.S. Thesis, Department of Civil and Environmental Engineering, University of Maryland, College Park, MD.

[38] Janpengpen, A. (2006). *Real-Time Monitoring and Forecasting of Odor after Dewatering in Wastewater Treatment*. M.S. Thesis, Department of Civil and Environmental Engineering, University of Maryland, College Park, MD.

[39] Kansakar, B. R., and Polprasert, C. (1983). Integrated Wastewater Management. *Journal of Environmental Engineering* **109**(3): 619-630.

[40] Kim, H., Murthy, S. N., McConnell, L. L., Peot, C., Ramirez, M., and Strawn, M. (2002). Characterization of wastewater and solids odors using solid phase microextraction at a large wastewater treatment plant. *Water Science and Technology* **46**(10): 9-16.

[41] Kim, H., Murthy, S. N., McConnell, L., Peot, C., Ramirez, M., and Strawn, M. (2003). Examination of mechanisms for odor compound generation during lime stabilization. *Water Environment Research: a Research Publication of the Water Environment Federation* **75**(2): 121125.

[42] Marseguerra, M., Zio, E., and Podofillini, L. (2004). A multiobjective genetic algorithm approach to the optimization of the technical specifications of a nuclear safety system. *Reliability Engineering and System Safety* **84**: 87-99.

[43] Miettinen, K. (1999). *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers: Boston/London/Dordrecht.

[44] Minor, S. D., and Jacobs, T. L. (1994). Optimal land allocation for solid-and-hazardous waste landfill siting. *Journal of Environmental Engineering* **120**(5): 1095-1108.

[45] Murthy, S. N., Sadick, T., Baily, W., Peot, C., Tolbert, D., and Strawn, M. (2001). Mitigation of odors from lime stabilized biosolids. *WEF Residuals and Biosolids Management Conference*, San Diego, CA (CD-ROM).

[46] Nash, S. G., and Sofer, A. (1996). *Linear and Nonlinear Programming.* McGraw-Hill: Singapore.

[47] Nema, A. K., and Gupta, S. K. (2003). Multiobjective risk analysis and optimization of regional hazardous waste management system. *Practice Periodical of Hazardous, Toxic, and Radioactive Waste Management* **7**(2): 69-77.

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

[49] ReVelle, C., and McGarity, A. E. (eds) (1997). *Design and Operation of Civil and Environmental Engineering Systems.* John Wiley & Sons: New York.

[50] Sherali, H. D., and Alameddine, A. (1992). A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization* **2**: 379-410.

[51] Shih, L. H., and Cheng, K. J. (2001). Multiobjective transportation planning for waste hauling. *Journal of Environmental Engineering* **127**(5): 450-455.

[52] Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation, and Application.* John Wiley & Sons: New York/Chichester/Brisbane/Toronto/Singapore.

[53] Taha, H. A. (1975). *Integer Programming: Theory, Appications, and Computations.* Academic Press: New York/San Francisco/London.

[54] Tebboth, J. R. (2001). *A Computational Study of Dantzig-Wolfe Decomposition.* Ph.D. Thesis, University of Buckingham.

[55] U.S. Environmental Protection Agency (EPA) (1994a). *Biosolids recycling: beneficial technology for a better environment* (Report no. EPA 832-R-94-009). Office of Water, EPA, Washington, DC.

[56] U.S. Environmental Protection Agency (EPA) (1994b). *Land Application of Sewage Sludge: A Guide for Land Appliers on the Requirements of the Federal Standards for the Use or Disposal of Sewage Sludge, 40 CFR Part 503* (Report no. EPA 831-B-93-002b). Office of Enforcement and Compliance Assurance, EPA, Washington, DC.

[57] U.S. Environmental Protection Agency (EPA) (1994c). *Plain English guide to the EPA part 503 biosolids rule* (Report no. EPA 832-R-93-003). Office of Wastewater Manangement, EPA, Washington, DC.

[58] U.S. Environmental Protection Agency (EPA) (1999). *Biosolids Generation, Use, and Disposal in The United States* (Report no. EPA530-R-99-009). Office of Solid Waste, EPA, Washington, DC.

[59] U.S. Environmental Protection Agency (EPA) (2000a). *Biosolids Technology Fact Sheet: Belt Filter Press* (Report no. EPA 832-F-00-057). Office of Water, EPA, Washington, DC.

[60] U.S. Environmental Protection Agency (EPA) (2000b). *Biosolids Technology Fact Sheet: Centrifuge Thickening and Dewatering* (Report no. EPA 832-F-00-053). Office of Water, EPA, Washington, DC.

[61] U.S. Environmental Protection Agency (EPA) (2000c). *Biosolids Technology Fact Sheet: Land Application of Biosolids* (Report no. EPA 832-F-00-064). Office of Water, EPA, Washingtion, DC.

[62] U.S. Environmental Protection Agency (EPA) (2000d). *Guide To Field Storage of Biosolids and Other Organic By-Products Used in Agriculture and for Soil Resource Management* (Report no. EPA 832-B-00-007). Office of Wastewater Management, EPA, Washingtion, DC.

[63] U.S. Environmental Protection Agency (EPA) (2002). *Status Report: Land Application of Biosolids* (Report no. 2002-S-000004). Office of Inspector General, EPA, Washington, DC.

[64] U.S. Environmental Protection Agency (EPA) (2006). *Emerging Technologies for Biosolids Management* (Report no. EPA 832-R-06-005). Office of Wastewater Management, EPA, Washington, DC.

[65] Vajda, S. (1972). *Probabilistic Programming.* Academic Press: New York/London.

[66] Venderbeck, F. (2000). On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research* **48**(1): 111-128.

[67] Vanderbeck, F., and Wolsey, L. A. (1996). An exact algorithm for IP column generation. *Operations Research Letters* **19**(4): 151-159.

[68] Vanderbei, R. J. (1996). *Linear Programming Foundations and Extensions.* Kluwer Academics Publishers: Boston/London/Dordrecht.

[69] Winston, W. L. (1993). *Operation Research Applications and Algorithm.* Duxbury Press: Belmont, CA.

[70] Wolsey, L. A. (1998). *Integer Programming.* John Wiley & Sons: New York.