DETERMINISTIC GLOBAL OPTIMIZATION APPROACH TO BILINEAR PROCESS NETWORK SYNTHESIS

DANAN SURYO WICAKSONO

NATIONAL UNIVERSITY OF SINGAPORE

2007

DETERMINISTIC GLOBAL OPTIMIZATION APPROACH TO BILINEAR PROCESS NETWORK SYNTHESIS

DANAN SURYO WICAKSONO

(B.Sc., BANDUNG INSTITUTE OF TECHNOLOGY, INDONESIA)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

DETERMINISTIC GLOBAL OPTIMIZATION APPROACH TO BILINEAR PROCESS NETWORK SYNTHESIS

DANAN SURYO WICAKSONO

2007

ACKNOWLEDGEMENTS

I express my most sincere gratitude to **Prof. I. A. Karimi** for providing the opportunity and freedom to explore a variety of exciting topics from Liquefied Natural Gas (LNG) technology and process network synthesis to mixed-integer programming and global optimization. I also genuinely appreciate his guidance through research ideas brainstorming, manuscripts writings and presentations as well as his constant encouragement to be productive, active, and competitive.

I wish to thank **Dr. Hassan Alfadala**, **Qatar University**, **Mr. Omar I. Al-Hatou**, and **Qatargas Operating Company Ltd.** for providing the opportunity to learn many industrial aspects of LNG plant operations.

I wish to thank **Dr. Lakshminarayanan Samavedham** and **Prof. Tan Thiam Chye** for strengthening the foundation of my basic chemical engineering knowledge in numerical methods and reaction engineering.

I wish to thank **A/P Chiu Min-Sen**, **A/P Rajagopalan Srinivasan**, and **Prof. Neal Chung** for broadening my chemical engineering perspective with advanced topics in multivariable controller design, artificial intelligence, and membrane technology.

I wish to thank **Prof. Gade Pandu Rangaiah** for providing the opportunity to tutor an undergraduate course in process design.

I deeply indebted to **National University of Singapore**, **Japan International Cooperation Agency**, and **ASEAN University Network / South East Asia Engineering Development Network** for facilitating a life-long beneficial quality higher education.

I would like to thank all my labmates, especially **Mr. Li Jie**, **Mr. Liu Yu**, and **Mr. Selvarasu Suresh** who created an inspirational thought-provoking working place in the lab.

TABLE OF CONTENTS

SUMMARY

Deterministic global optimization approach to bilinear process network synthesis is the focal point of this work. Process synthesis addresses the problem of finding the optimal arrangement of the chemical process flowsheet which is often represented as nonconvex programming problem exhibiting multiple local optimal solutions. Deterministic global optimization is required to obtain a guaranteed global optimal solution of such problems. Process synthesis problems which can be posed as bilinear programs, a class of nonconvex programs, are called as bilinear process network synthesis problems.

The first section of this work addresses the practical application of deterministic global optimization approach in solving industrial bilinear process network problems. In this section, the optimal operation problem on an existing fuel gas network in a natural gas liquefaction plant is presented. A superstructure and a corresponding mathematical programming model are proposed to model the possible structural alternatives for the fuel gas network. Efficient representation of the superstructure enables the use of a commercial solver to locate the global optimal solution of such problem. The deterministic global optimization approach leads to the reduction in fuel-from-feed consumption. Further reduction is obtained through the integration of jetty boil-off gas as an additional fuel which is solved using the same procedure.

The second section concentrates on the theoretical-algorithmic study of the deterministic global optimization technique in solving bilinear programs. The idea of using *ab inito* partitioning of the search domain to improve the relaxation quality is discussed. Such idea relies on piecewise under- and overestimators. It produces tighter relaxation as compared to conventional technique based on continuous linear programming which is often weak and thus slows down the convergence rate of the global optimization algorithm. Several novel modeling strategies for piecewise under- and overestimators via mixed-integer linear programming are proposed. They are evaluated using a variety of process network synthesis problems arising in the area of integrated water system design and non-sharp distillation column sequencing. Metrics are defined to measure the effectiveness of such technique along with some valuable insights on properties. Several theoretical results are presented as well.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1. Process Design and Synthesis

Chemical process design is one of the most classic yet evergreen topics for chemical engineers. It often embodies the archetypal ultimate goal for many other chemical engineering activities. It is complex, requiring the use of numerous science and engineering know-how in an integrated manner to devise processing systems transforming raw materials into products that best achieve the desired objective. Chemical processes distinguish themselves from other engineering objects in the sense that they are typically designed for very long lifetimes while simultaneously capital and operating cost intensive. Thus, the prospective of having many years of continuing incurred costs emphasizes the importance of a good process design. It is well known that process design, an activity that may only account for around two to three percent of the project cost, determines significant percentages of capital and operating costs of the final process plant as well as its profitability. While empirical judgment is imperative, good process design is not a trivial task in the absence of systematic procedures.

The preliminary phase for chemical process design is the flowsheet synthesis activity, also called as *process synthesis*. It poses a problem of arranging a set of processing equipments in the availability of a set of raw materials and energy sources to produce a set of desired products under certain performance criteria. It includes several steps. The first is to gather required information to uncover existing alternatives. Next, the process alternatives need to be represented in a concise manner for decision making. In order to do this, several criteria to asses and evaluate are required the value of a certain design. These criteria are typically related with technical and economic performances. Due to the extensive amount of possible alternatives, a systematic procedure is required to generate and search among these alternatives.

1.2. Superstructure

The need to develop a systematic procedure for process design results in the birth of the so-called *superstructure* (Smith, 1995, Biegler *et al*., 1997). In a superstructure, several possible design alternatives are represented in a set of arcs and graphs. Typically arcs represent inteconnection in spatial, temporal, or logical domain of nodes symbolizing the resources (e.g. raw materials, energy utilities, processing equipments). This representation is later transformed in an optimization problem, which are typically a mathematical programming problem (Edgar *et al*., 2001). The objective function contains the technical and economic criteria that measure the performance of a proposed design such as maximizing profits, product yields, or minimizing costs, consumption of raw materials, consumption of energy. The constraints capture the physical nature of the design alternatives (e.g. total mass balance, component mass balance, and energy balance) as well as resource restrictions (availability of raw material and utilities) and quality specifications (product purity and environmental regulations). Equations involved in the objective function and constraints can be linear or nonlinear. Variables involved can be continuous and discrete. Continuous variables represent process variables such as flow rates, compositions, temperatures, and pressure. Discrete variables represent the logic of the process such as the existence of a certain stream and processing sequence recipe.

A mathematical program which contains only linear equations and continuous variables is called as *Linear Programming* (LP) problem. If at least one integer variable is added, the mathematical program becomes a *Mixed-integer Linear Programming* (MILP) problem. If at least one equation is nonlinear, the mathematical program becomes a *Nonlinear Programming* (NLP) problem. *Mixed-integer Nonlinear Programming* (MINLP) problem represents a situation where integer and continuous variables as well as nonlinear and linear constraints exist simultaneously.

1.3. Nonconvex Programming and Deterministic Global Optimization

Several process synthesis problems lead to a *nonconvex programming* problem which exhibits multiple local optimal solutions. Such a feature imposes difficulty, since obtaining the best of the best solutions (i.e. global optimal solution) is desirable in many process synthesis problems. Global optimization approach is required to obtain the global optimal solution of a nonconvex programming problem. While such approach may be attempted via heuristic methods such as genetic algorithm and simulated annealing, the obtained solution is not guaranteed to be the true global optimal solution. Another approach called as *deterministic global optimization* approach can provide such a guarantee. In addition, the deterministic approach can asses the solution quality by measuring the gap between the upper and lower bounds of the global optimal solution.

Several nonconvex programming problems can be found in the field of blending and pooling problem, integrated water systems design, heat exchanger network design, and non-sharp distillation sequencing. For such problem, nonconvexities arise from the product of two different continuous variables: stream flow rates and compositions or steam flow rates and temperatures. Thus, the problem can be classified as *bilinear programming* problem (BLP). Such problem is important because it represents an omnipresent situation in most chemical process plants.

Moreover, bilinear term is one of the building blocks for a wider class of factorable nonconvex programming problem in which the nonconvex terms can be broken down into recursive sums and products of univariate terms. Factorable nonconvex programming is a powerful tool for a vast range of science and applications in chemical engineering and other fields. Throughout this thesis, process network synthesis problems which are modeled using BLP are termed as *bilinear process network synthesis*.

1.4. Research Objective

This work focuses on deterministic global optimization approach in solving bilinear process network synthesis. The objectives of this work are to: (1) develop a systematic methodology based on an industrial application of deterministic global optimization of bilinear process network, which is chosen to be a fuel gas network in a natural gas liquefaction plant (2) develop a novel strategy to improve the algorithm of deterministic global optimization approach in solving BLPs together with some theoretical and computational studies.

1.5. Thesis Outline

This thesis is divided into two main sections. The first section consists of Chapter 2 and 3. It discusses the practical importance of deterministic global optimization approach in solving BLPs. In this section a problem on a fuel gas network in a natural gas liquefaction plant is described. The problem is later represented using a superstructure which then transformed into a MINLP with bilinear terms. Efficient superstructure representation makes available the use of commercial solver BARON to locate the global optimal solution. Significant amount of improvement is achieved in the form of fuel-to-feed consumption reduction.

The second section consists of chapter 4, 5, and 6. This section focuses on a novel technique to obtain the bound of the global optimal solution. The novel technique is capable of locating tighter bound as compared to the conventional one. It relies on *ab inito* partitioning of the search domain, called as piecewise relaxation. Several novel modeling strategies for piecewise under- and overestimators are proposed in the frame of mixed-integer linear programming invoking a two-level relaxation hierarchy. These novel strategies are based on three systematic approaches (i.e. Big-M, Convex Combination, and Incremental Cost) and two segmentation schemes (i.e. arbitrary and identical). Computational and theoretical studies are performed on the models developed in the second part. The studies employ a variety of problems from process network synthesis (i.e. integrated water system design and non-sharp distillation column sequencing). Computational study favors the novel models over the exisiting models based on disjunctive programming. Several properties of the models are observed and theoretically studied. Metrics to define the effectiveness of such model is introduced along with the theoretical background.

Eventually, Chapter 7 summarizes the advances obtained from these works.

SECTION I:

INDUSTRIAL APPLICATION

(In collaboration with Dr. Hassan Alfadala from Qatar University and

Mr. Omar I. Al-Hatou from Qatargas Operating Company;

Data and models related to this work are the property of

Qatargas Operating Company**)**

Chapter 2

A REVIEW ON LIQUEFIED NATURAL GAS (LNG)

2.1. Natural Gas

Natural gas comes from reservoirs beneath the earth's surface. Sometimes it occurs naturally, sometimes it comes to the surface with crude oil (associated gas), and sometimes it is being produced constantly such as in landfill gas. Natural gas is a fossil fuel, meaning that it is derived from organic material deposited and buried in the earth millions of years ago. Other fossil fuels are coal and crude oil. Together crude oil and gas constitute a type of fossil fuel known as "hydrocarbons" because the molecules in these fuels are combinations of hydrogen and carbon atoms.

Natural gas is a highly combustible odorless and colorless hydrocarbon gas largely composed of methane (Figure 2.1). The other components in natural gas are ethane, propane and butane with trace amounts of nitrogen and carbon dioxide. Natural gas is the most environmentally friendly (Table 2.1) and one of the most abundant fossil fuels in the world, thus it is the economic and environmental fuel of choice. The demand for natural gas has been growing rapidly in recent years and is expected to grow at a much faster pace than crude oil.

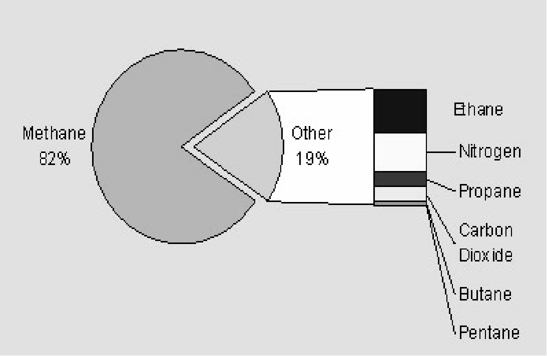

Figure 2.1. Typical Natural Gas Composition

Table 2.1. Comparison of air pollutant emissions between hydrocarbon fuels (http://www.eia.doe.gov/pub/oil_gas/natural_gas/analysis_publications/natural_gas_1998_issues_trends/pdf/chapter2.pdf)

Pollutant (Lb / 10^6 Btu of energy input)	**Natural Gas**	**Oil**	**Coal**
Carbon Dioxide	117,000	164,000	208,000
Carbon Monoxide	40	33	208
Nitrogen Oxides	92	448	457
Sulfur Dioxide	1	1,122	2,591
Particulates	7	84	2,744
Mercury	0	0.007	0.016

2.2. Liquefied Natural Gas

Liquefied natural gas (LNG) is natural gas that has been processed to remove impurities and cooled to the point that it condenses to a liquid (Flynn, 2005; Timmerhaus and Reed, 2007), which occurs at a temperature of approximately -161°C at atmospheric pressure. Liquefaction reduces the volume by approximately 600 times and thus making it more economical to transport between continents in specially designed ocean vessels, whereas traditional pipeline transportation systems would be less economically attractive and could be technically or politically infeasible (Greenwald, 1998). Thus, LNG technology makes natural gas available throughout the world.

The growing popularity of LNG is due to two reasons. First, there is a continuous and growing demand for fuel from the key markets of Asia, Europe and North America to meet the ever growing energy requirements. These end-user markets are thousand of miles from countries where there are vast resources of natural gas in countries such as the Middle East and South America. Second, it will be more economical to transport the natural gas for long distance by ship as compared to via long pipelines. Furthermore, the geographical location of the importing and exporting countries prevents the use of long pipelines as the main transportation means.

2.3. LNG Supply Chain

In order to deliver natural gas in the form of LNG, several huge companies have to invest in a number of operations that is highly linked and dependent to each other called as *LNG supply chain*. The typical LNG supply chain consists of: exploration, production, liquefaction, shipping, regasification and distribution.

The aim of the exploration stage is to find in the earth crust. Search for natural gas deposits begins with geologists and geophysicists using their knowledge of the earth to locate the geographical areas. Geologists survey, map the surface & sub-surface characteristics and extrapolate which areas are more likely to contain a natural gas reservoir. Geophysicists conduct further more tests to get more detailed data and uses the technology to find and map under rock formations.

Production involves extraction and processing. Extraction deals with the withdrawal of natural gas from its sources inside earth's crust. Later, natural gas undergoes some processing steps to satisfy pipeline requirements. These requirements include oil, water, and condensate removal. Processed natural gas is transported to liquefaction plant by pipeline.

Liquefaction is to transform the natural gas feed into LNG which is then transported by a special ship from the exporting terminal to the importing terminal. LNG stored in tanks is vaporized or regasified to gas state (natural gas) before its connected to the transmission system. Regasification involves pressuring the LNG above the transmission system pressure and then warmed by passing it through pipes heated by direct-fired heaters, seawater or through pipes that are in heated water. The vaporized gas is then regulated for pressure and enter the pipeline system for distribution.

2.4. Natural Gas Liquefaction Plant

A natural gas liquefaction facility is typically consists of several parallel units called as *trains* (Flynn, 2005). Each train is designed using similar technology and consists of similar processing parts. However, as the facility expands, it is possible that trains which were built earlier may have different technology and capacity as compared to the newly built trains. In each train, the natural gas feed typically undergoes several treatment processes to remove impurities (e.g. CO_2, H_2S, water), recover heavier hydrocarbon (e.g. propane and butane sold as different products or used as refrigerant), liquefaction to LNG, upgrading of methane content through N_2 rejection, and helium recovery. These trains are supported by utility plants assisting their operational needs such as steam, cooling water, and fuel.

2.5. Fuel Gas Network in a Natural Gas Liquefaction Plant

A natural gas liquefaction process is highly energy-intensive. Thus, efficient use of energy is very important. A key facility of natural gas liquefaction plant is the fuel gas system which is part of the plant utilities section. The function of this facility is to satisfy the plant energy demands. It is unique because the sources of fuel are coming from the plant itself. The fuel itself is used for generating power in the form of both electricity and steam to support plant operations in onsite and offsite area. Fuel gas system is designed considering the availability of tail gas in the plant, equipment design requirements as the user of fuel gases and these have to be balanced in such manner that no flaring occur.

Chapter 3

OPTIMIZATION OF FUEL GAS NETWORK IN A NATURAL GAS LIQUEFACTION PLANT

3.1. The Fuel Gas Network

The fuel gas network which is the focus of this study has several distinct components as discussed further (Qatargas operating manual).

3.1.1. Fuel Sources

Fuel sources are located upstream in a fuel gas network. They are gases which can be utilized as fuel. There are two major sources of fuel: tail gases and feed gases. Tail gases are leftover gases which are neither nor product or recyclable. These gases correspond to production losses and therefore should be minimized by using them fully as fuel gases if possible. Excess tail gases which cannot be used as fuel are burned in flare. Tail gases are produced before and after the purification units. Tail gases produced before the purification units typically has low methane content and therefore low Wobbe Index (WI), while tail gases produced after the purification units typically has high methane content and high WI.

Fuel gases taken from feed are used to fill the gap between plant energy demand and the amount of energy which can be provided by tail gases. However, the usage of feed as fuel decreases the quantity of LNG produced and hence should be minimized.

During emergency event where the amount of tail gases and feed are not sufficient, fuel may be supplied by feedstock gases coming from the natural gas wells. However, these gases are rich in impurities which may be harmful to the fuel sinks.

3.1.2. Fuel Sinks

Fuel sinks are located downstream of the fuel gas network. They transform potential energy contained by fuel into more practically useful form. Typical fuel consumers are process driver turbines, power generator turbines, boilers, and incinerators. Process turbines drive the refrigerant compressors. Power turbines and boilers provide the plant with necessary electricity and steam, respectively. For the sake of complicity, flare may also be included as one of the sinks although it does not produce energy and causes negative environmental effect.

3.1.3. Fuel Source - Sink Compatibility

Every sink has different fuel requirements based on its design while each fuel source has its own characteristic such as LHV (Lower Heating Value) and composition. The interchangeability between these various fuels is measured by Wobbe Index (WI). Thus, each sink must be fed by fuel which satisfies a certain range of Wobbe Index. In order to achieve the desired WI specification, some operations such as mixing required.

3.2. Problem Statement

Here, we present two different problems. The first one is optimizing the operation of fuel gas network under the current conditions of fuel sources and sinks. The second one considers the integration of an additional fuel source named jetty boil-off gas (BOG).

3.2.1. Optimal Operation of the Existing Fuel Gas Network

We consider the optimal configuration of the fuel gas network. The network consists of fuel gas sources, sinks, mixers, fuel sinks, and connecting pipelines. The objective of this study is to design a network which gives minimum fuel consumption.

The decisions which have to be determined are mixing and distribution scenarios. No chemical reactions, separations, and phase changes involved. Conditions of fuel sources, such as flow rate and composition are determined by the operating mode. The requirements imposed by fuel sinks are allowable WI range, and fuel energy content. Our problem can be summarized as follow:

given:

1. sources and sinks (existing and additional) and their characteristics
2. fuel supply and demand, including quality requirements

determine:

1. optimal fuel mixing and distribution scenario
2. minimum fuel consumption

3.2.2. Integrating Recovered Jetty Boil-off Gas as an Additional Fuel

In addition, we consider an additional fuel source in the form of jetty BOG which is vapors generated during the loading of LNG into delivery ships. Hence, it is not produced continuously. For the purpose of this study, we use the average jetty BOG rate throughout the year which is a deterministic value based on the ship arrival schedule.

It is desirable to integrate this additional fuel into the existing fuel gas network. However, integrating this additional fuel source optimally and satisfactorily within the existing fuel gas network is not a trivial task, as extra piping and/or equipment may be needed to accommodate this modification. Furthermore, this should be done without affecting the fuel quality requirements of existing equipments.

3.3. Solution Methodology

In this work, we consider all possible scenarios in one superstructure and then formulate the selection of the best structure as an optimization problem. The problem is then solved to global optimality. The proposed approach is general in that it can be extended to any numbers of sources and sinks.

3.3.1. Superstructure

Figure 3.1 shows the proposed superstructure for this problem. Nodes *i*, *m*, and *o* represent fuel sources, mixers, and sinks, respectively while arcs represent interconnection between fuel sources, mixers, and sinks. It should be noted that the number of mixers in the superstructure is equal to the number of sinks concerned. One source node does not necessarily correspond to one physical source. Sources which have identical properties can be lumped into a single node. Similar concepts can also be applied to sinks. Using this strategy called *reduced superstructure*, the size of the problem is reduced and so does the computational effort required.

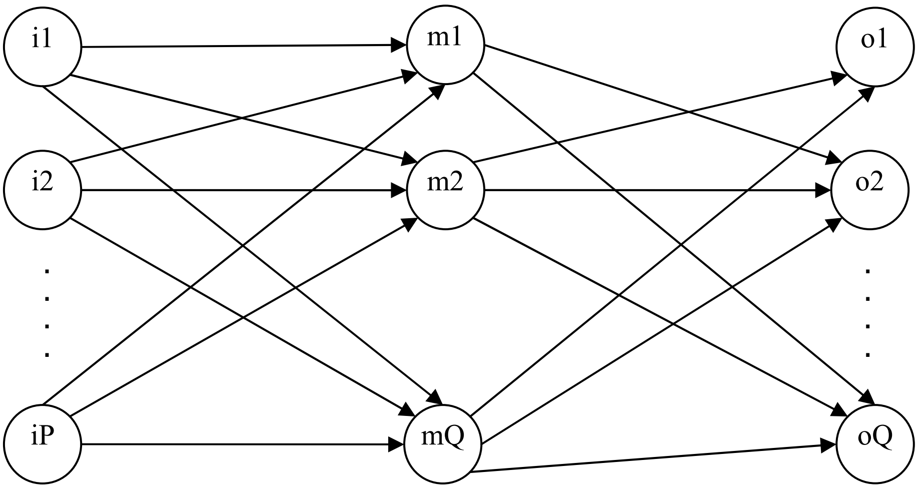

Figure 3.1. Fuel gas network superstructure with P sources and Q sinks

3.3.2. Mathematical Programming Model

Mathematical formulation is developed based on the given superstructure in such manner that nonlinearities are minimized. The model incorporates overall and component material balance as well as energy balance. The resulting formulation is a mixed-integer nonlinear programming (MINLP) problem with bilinear terms.

Sets

i	fuel sources
m	mixers
o	fuel sinks
c	components

Parameters

supply and demand

S(i)	fuel supply of fuel source i
D(o)	energy demand of fuel sink o

fixed operation costs

FCP(i,m)	fixed construction and operation cost for stream p(i,m)
FCQ(m,o)	fixed construction and operation cost for stream q(m,o)

variable operation costs

VCI(i) variable operation cost for using fuel source i

[VCI(i) > 0 if fuel source i is tail gas, VCI(i) < 0 if fuel source i is feed gas]

VCO(o) variable operation cost for using fuel sink o

[VCO(o) > 0 if fuel sink o is a flare, VCO(o) = 0 if fuel sink o is not a flare]

VCP(i,m)	variable operation cost for stream p(i,m)
VCQ(m,o)	variable operation cost for stream q(m,o)

fuel characteristics

x(i,c)	composition of component c fuel source i
f(i)	quality (Wobbe index) of fuel source i
H(c)	individual lower heating value of component c

sink composition requirements

hU(o)	upper bound for quality (Wobbe index) of fuel entering sink o
hL(o)	lower bound for quality (Wobbe index) of fuel entering sink o

bounds for flow rates

pU(i,m)	upper bound for stream p(i,m)
pL(i,m)	lower bound for stream p(i,m)
qU(i,m)	upper bound for stream q(m,o)
qL(i,m)	lower bound for stream q(m,o)

Binary Variables

zp(i,m)	1 if stream p(i,m) exists in the optimal solution, 0 otherwise
zq(m,o)	1 if stream q(m,o) exists in the optimal solution, 0 otherwise

Continuous Variables

p(i,m)	fuel flow rate from source i to mixer m
q(m,o)	fuel flow rate from mixer m to sink o

y(m,c)	fuel composition exiting mixer m
z(o,c)	fuel composition entering sink o
g(m)	fuel quality exiting mixer m
T	total costs

$$\begin{aligned} T = \min \Big\langle & \sum_{i}\sum_{m}\Big[\big(VCI(i) + VCP(i,m)\big)\cdot p(i,m) + FCP(i,m)\cdot zp(i,m)\Big] \\ & + \sum_{m}\sum_{o}\Big[\big(VCQ(m,o) + VCO(o)\big)\cdot q(m,o) + FCQ(m,o)\cdot zq(m,o)\Big]\Big\rangle \end{aligned} \qquad (3.1)$$

Equation (3.1) evaluates the operational costs of the system and hence is the objective function. The first, second, fourth, and fifth terms describe the variable operating costs related to the usage of fuel source *i*, stream *p*, stream *q*, and the usage of fuel sink *o*, respectively. The third and sixth terms describe the fixed operating costs related to the existence of stream *p* and stream *q*, respectively.

Equations (3.2) are the total balance at mixer *m*.

$$\sum_{i} p(i,m) = \sum_{o} q(m,o) \qquad \forall m \qquad (3.2)$$

Equations (3.3) are the component balance at mixer *m*.

$$\sum_{i}\left[p(i,m)\cdot x(i,c)\right] = y(m,c)\cdot\sum_{o} q(m,o) \qquad \forall m\ \forall c \qquad (3.3)$$

Equations (3.4) are the component balance at sink *o*.

$$\sum_{m}\left[y(m,c)\cdot zq(m,o)\right] = z(o,c) \qquad \forall m\ \forall c \qquad (3.4)$$

Equations (3.5) are the quality balance at mixer *m*. Quality of fuel gas is assessed using Wobbe Index (WI). In this study, WI change due to mixing is assumed to be linear.

$$\sum_i \left[p(i,m)\cdot f(i)\right] = g(m)\cdot \sum_o q(m,o) \qquad \forall m \qquad (3.5)$$

Equations (3.6) are the quality balance at sink *o*.

$$hL(o,c) \le \sum_m \left[g(m,c)\cdot zq(m,o)\right] \le hU(o,c) \qquad \forall m\ \forall c \qquad (3.6)$$

Equations (3.7) ensure that fuel usage is not exceeding the supply by fuel sources.

$$\sum_m p(i,m) \le S(i) \qquad \forall i \qquad (3.7)$$

Equations (3.8) ensure that fuel going into fuel sink *j* satisfies the energy demand of the corresponding fuel sink.

$$\sum_m \sum_c \left(z(o,c)\cdot q(m,o)\cdot H(c)\right) \ge D(o) \qquad \forall o \qquad (3.8)$$

Equation (3.9) ensure that only a single layer mixing exists in the network.

$$\sum_m zq(m,o) \le 1 \qquad \forall o \qquad (3.9)$$

Binary variable *zq*(*m*,*o*) models the interconnection between mixer *m* and sink *o*. Therefore, nonconvex bilinear terms in the component material balance can be exactly linearized. This reduction in nonlinearities significantly improves the computational performance of the MINLP.

Equations (3.10) and (3.11) connect the logical relationship between continuous variable *p* and *q* representing stream flowrate and binary variable *zp* and *zq*, respectively.

$$zp(i,m) \cdot pL(i,m) \le p(i,m) \le zp(i,m) \cdot pU(i,m) \qquad \forall i \ \forall m \tag{3.10}$$

$$zq(m,o) \cdot qL(m,o) \le q(m,o) \le zq(m,o) \cdot qU(m,o) \qquad \forall m \ \forall o \tag{3.11}$$

3.4. Case Study

An industrial fuel gas network in an LNG plant comprising three trains as depicted in Figure 3.2 was considered in this work. Later on, we integrate one additional fuel source which is jetty BOG. It consists of four major fuel sources and four major fuel sinks. Several sources and sinks belong to a certain train. The four major sources for fuel gas are: tankage boil off gas (BOG), fuel from feed (FFF), end flash gas (EFG), and high pressure (HP) flash gas. Tankage BOG are gases generated in the storage tanks due to heat leaks. FFF is part of the feed gases taken from the mercury removal unit outlet stream in each train. EFG comes from the top product of Nitrogen Rejection Unit (NRU) and HP flash gases are sour gas obtained from the acid gas removal unit in each train. Hence, the first source comes from the offsite facilities while the other three sources come from the process train itself.

BOG, EFG, and HP flash gas usage corresponds to the production losses and called as tail gases. Therefore, they are expected to be fully consumed by the fuel gas system. Excess of these three sources are sent to the flare facilities. In the other hand, FFF usage is only to fill the gap between the plant power requirements and the amount of power which can be extracted from the other three sources (i.e. BOG, EFG, and HP flash gas). FFF is unwanted source of fuel since increasing FFF usage decreases the amount of feed gas flowing to the main cryogenic heat exchanger (MCHE) causing reduced LNG production. Therefore, FFF consumption should be minimized. Thus, a positive cost is associated with the use of FFF and flaring.

3.5. Results and Discussion

The proposed model was implemented in GAMS 22.2 (Brooke et al., 2005) and solved using BARON 7.5 (Sahinidis, 1996) on a Dell Optiplex GX620 with Windows XP Professional operating system, Pentium IV HT 3 GHz processor, and 2 GB RAM.

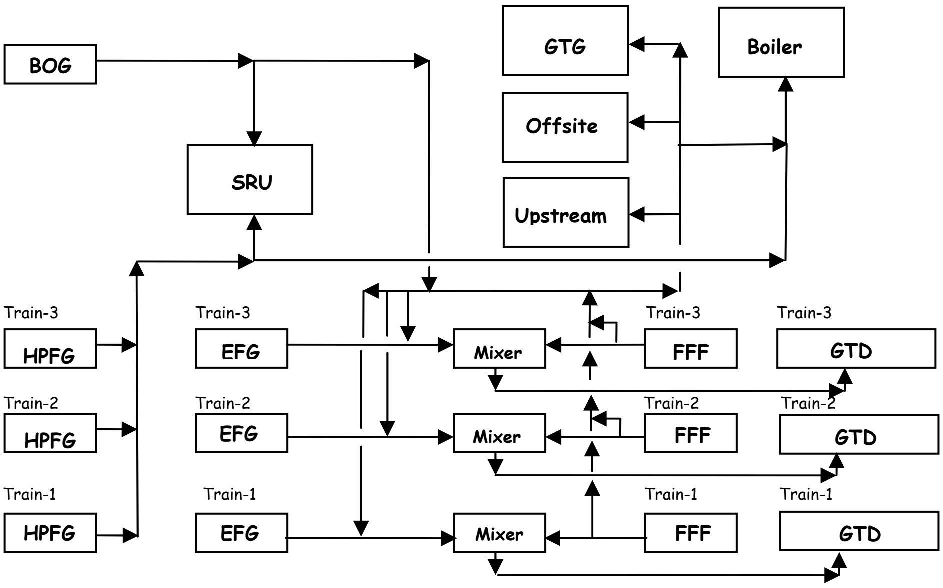

Figure 3.2. Existing fuel gas network

The guaranteed best optimal solution suggests a significant FFF consumption reduction. Note that the BARON is able to locate the global optimal solution due to manageable size of our superstructure representation. BARON guarantees the global optimality of the solution since through the course of “branching” and “bounding” (in the context of BARON is “reducing”) the gap between the upper and lower bound is closed. In a global minimization problem, the upper bound is any feasible solution of the original problem and the lower bound is obtained from the relaxation problem. This enhancement corresponds to increasing LNG production rate and thus plant operation profitability. In the case of jetty BOG integration, the comparison between the fuel gas consumption before and after jetty BOG integration is shown in Table 3.1. It is shown that by integrating jetty BOG as additional fuel, the FFF consumption decreases by about 15% overall. This reduction further increases the plant efficiency by reducing the use of FFF.

Table 3.1. Fuel consumption before and after jetty BOG integration (flow unit)

Fuel source	Before	After
FFF	53.62	45.77
Jetty BOG	0	50.21

SECTION II:

THEORETICAL-ALGORITHMIC STUDY

Chapter 4

A REVIEW ON DETERMINISTIC GLOBAL OPTIMIZATION ALGORITHM FOR BILINEAR PROGRAMS

4.1. Introduction

Many practical problems of interest in chemical engineering and other fields can be formulated as optimization problems involving bilinear functions of continuous decision variables. For instance, the mathematical programming formulations for the pooling problem (Haverly, 1978), integrated water systems synthesis (Takama *et al*., 1980), process network synthesis (Quesada and Grossmann, 1995), crude oil operations scheduling (Reddy *et al*., 2004; Reddy *et al*., 2004), as well as fuel gas network design and management in Liquefied Natural Gas (LNG) plants (Wicaksono *et al*., 2006; Wicaksono *et al*., 2007) all involve bilinear products of continuous decision variables such as stream flows and compositions. The optimization formulations involving such bilinear functions, called bilinear programs (BLPs), belong to the class of nonconvex nonlinear programming problems that exhibit multiple local optima. For such problems, a local nonlinear programming (NLP) solver often provides a sub-optimal solution or even fails to locate a feasible one. However, the need for obtaining a guaranteed globally optimal solution is real, essential, and often critical, in many practical problems mentioned above. Understandably, this has led to a flurry of research activities (Biegler and Grossmann, 2004; Floudas *et al*., 2005) in the last two decades on global optimization, which involves obtaining a theoretically guaranteed globally optimal solution to a nonconvex mathematical program.

4.2. Spatial Branch-and-Bound

While several global optimization algorithms (Grossmann, 1996; Floudas, 2000; Tawarmalani and Sahinidis, 2002; Floudas and Pardalos, 2004) exist today, the most common ones use the so-called spatial branch-and-bound framework (Horst and Tuy, 1993; Tuy, 1998). This framework is similar to the standard branch-and-bound algorithm widely used in combinatorial optimization (Nemhauser and Wolsey, 1988). The main difference is that the spatial branch-and-bound branches in continuous rather than discrete variables. Tight lower and upper bounds, efficient procedures for obtaining them, and clever strategies for branching are the main challenges in this scheme. For a minimization (maximization) problem, any feasible solution acts as a valid upper (lower) bound and can be obtained by means of a local NLP solver (e.g. CONOPT, MINOS, SNOPT). For lower (upper) bounds, however, the common approach is to solve a good convex (concave), linear or nonlinear, relaxation of the original problem to global optimality using a standard LP solver (e.g. CPLEX, OSL, LINDO, XA) or a local NLP solver. If the gap between the lower and upper bounds exceeds a pre-specified tolerance for any partition of the search space, that partition is branched further, until the gap reduces below the tolerance.

The development of this branch-and-bound approach has been the focus of much research during the last decade. BARON (*Branch-And-Reduce Optimization Navigator*), a commercial implementation of this framework, by Sahinidis (1996) has been a significant development. Ryoo and Sahinidis (1996) introduced a *branch-and-reduce* approach with a range-reduction test based on Lagrangian multipliers. Zamora and Grossmann (1999) proposed a *branch-and-contract* global optimization algorithm for univariate concave, bilinear, and linear fractional functions. The emphasis was on reducing the number of nodes in the branch-and-bound tree through the proper use of a contraction operator. This involved maximizing and minimizing each variable within a linear relaxation problem. Neumaier *et al*. (2005) presented test results for the software performing complete search to solve global optimization problems and concluded that BARON is the fastest and most robust.

The success of a spatial branch-and-bound scheme depends critically on the rate at which the gap between the lower and upper bounds reduces. For faster convergence, this gap must decrease quickly and monotonically, as the search space reduces. In other words, devising efficient procedures for obtaining tight bounds is a key challenge in global optimization, as both the quality of bounds and the time required to obtain them strongly influence the overall effectiveness and efficiency of a global optimization algorithm. As stated earlier, relaxation of the original problem is the most widely used procedure, so the quality of relaxation and the effort required for its solution are extremely critical.

4.3. Convex Relaxation

Much research has focused on constructing a convex relaxation for factorable nonconvex NLP problems. This class of problems exclusively involves factorable functions, which are the ones that can be expressed as recursive sums and products of univariate functions (McCormick, 1976). Several researchers (Kearfott, 1991; Smith and Pantelides, 1999) proposed symbolic reformulation techniques to transform an arbitrary factorable nonconvex program into an equivalent standard form in which all nonconvex terms are expressed as special nonlinear terms such as bilinear and concave univariate terms. This approach employs the fact that all factorable algebraic functions involve one or more unary and/or binary operations. Transcendental functions, such as the exponential and logarithm of a single variable, are examples of the former and five basic arithmetic operations of addition, subtraction, multiplication, division, and exponentiation form the latter. Therefore, these special nonlinear terms form the building blocks for factorable nonconvex problems that abound in a wide range of disciplines including chemical engineering. In addition to those mentioned earlier, many problems in process systems engineering such as process design, operation, and control fall within this scope. Thus, by addressing bilinear programs in this work, we are essentially addressing the much wider class of factorable nonconvex programs.

LP relaxation is the most widely used technique for obtaining lower bounds for a factorable nonconvex program. McCormick (1976) was the first to present convex underestimators and concave overestimators for the bilinear term on a rectangle. Later, Al-Khayyal and Falk (1983) theoretically characterized these under- and overestimators as the convex envelope for a bilinear term. Foulds *et al*. (1992) utilized the bilinear envelope embedded inside a branch-and-bound framework to solve a bilinear program for the single-component pooling problem based on total flow formulation. Tawarmalani *et al*. (2002) showed that tighter LP relaxations can be produced by disaggregating the products of a single continuous variable and a sum of several continuous variables. LP relaxation, however, is often weak, and thus other forms of relaxation have also been proposed.

Androulakis *et al*. (1995) proposed a convex quadratic NLP relaxation, named *αBB underestimator*, which can be applied to general twice continuously differentiable functions. However, the tightness of such a relaxation for specific problems involving bilinear terms is inferior compared to its LP counterpart. Meyer and Floudas (2005) attempted to improve the tightness of the classical αBB underestimator via a smooth piecewise quadratic, perturbation function.

Sherali and Alameddine (1992) introduced a novel technique, called *Reformulation-Linearization Technique* (RLT), to improve the relaxation of a bilinear program by creating redundant constraints. Ben-Tal *et al*. (1994) proposed an alternative formulation for a bilinear program for the multicomponent pooling problem based on individual flow formulation and employed a Lagrangian relaxation to solve it within a branch-and-bound framework. Adhya *et al*. (1999) proposed another Lagrangian approach for generating valid relaxations for the pooling problem that are tighter than LP relaxations. Tawarmalani and Sahinidis (2002) showed that the combined total and individual flow formulation for the bilinear programs of multicomponent pooling and related problems proposed by Quesada and Grossmann (1995) produces a tighter LP relaxation compared to either the Lagrangian relaxation or the LP relaxation based on either the total or individual flow formulations alone. While the formulation of Quesada and Grossmann (1995) can be derived using the RLT, no theoretical and/or systematic framework exists to date for deriving RLT formulations with predictably efficient performance for general nonconvex programs.

4.4. Piecewise Relaxation

An interesting recent development is the idea of *ab initio* partitioning of the search domain, which results in a relaxation problem that is a mixed-integer linear program (MILP) rather than LP, called as *piecewise MILP relaxation*. Some recent work has shown the promise of such an approach in accelerating the convergence rate in several important applications such as process network synthesis (Bergamini *et al*., 2005), integrated water systems synthesis (Karuppiah and Grossmann, 2006), and generalized pooling problem (Meyer and Floudas, 2006). However, much work is in order to fully exploit the potential of such an approach. All previous works have reported that the lower bounding problem in global minimization based on piecewise MILP relaxation is the most time consuming step. Moreover, it is solved repeatedly inside a global optimization framework (e.g. spatial branch-and-bound, outer approximation, or RLT) and thus many issues such as the quality and efficiency of piecewise MILP relaxation demand further attention. In this work, we develop, analyze, compare, and improve several novel and existing formulations for piecewise MILP under- and overestimators for BLPs that may arise solely or within some Mixed-integer Bilinear Programming (MIBLP) problems. We demonstrate the superiority of our under- and overestimators as well as corresponding formulations using a variety of examples.

Chapter 5

MODELING PIECEWISE UNDER- AND OVERESTIMATORS FOR BILINEAR PROGRAMS VIA MIXED-INTEGER LINEAR PROGRAMMING

5.1. Problem Statement

Our ultimate goal is to solve the following global optimization problem by employing piecewise mixed-integer relaxation.

$$\mathbf{P} = \left\{ \underset{\boldsymbol{x}^L \le \boldsymbol{x} \le \boldsymbol{x}^U}{\text{Min}} f(\boldsymbol{x}) \text{ subject to } \mathbf{g}(\boldsymbol{x}) \le 0 \text{ and } \mathbf{h}(\boldsymbol{x}) = 0 \right\}$$

where $\boldsymbol{x} \in \Re^{\mathrm{n}}$ is a vector of continuous variables with bound vectors $\boldsymbol{x}^L$ and $\boldsymbol{x}^U$, $f(\boldsymbol{x})$ is an $\Re^{\mathrm{n}} \to \Re$ scalar objective function, and $\mathbf{g}(\boldsymbol{x})$ and $\mathbf{h}(\boldsymbol{x})$ are vectors of $\Re^{\mathrm{n}} \to \Re$ scalar functions representing the inequality and equality constraints. All functions are twice continuously differentiable and involve linear and bilinear terms only.

To achieve the above goal, we focus on developing several novel piecewise MILP under- and overestimators for the following nonconvex feasible region (S).

$$S = \{(x, y, z) \mid z = xy, x \in \Re, y \in \Re, x^L \le x \le x^U, y^L \le y \le y^U \}$$

5.2. The Role of Relaxation in Solving Optimization Problem

Relaxation involves outer-approximating the feasible region of a given problem and underestimating (overestimating) the objective function of a minimization (maximization) problem. A relaxation does not fully replace the original problem, but provides guaranteed bounds on its solutions. In a minimization (maximization) problem, the optimal solution of the relaxation problem provides a lower (upper) bound on the optimal objective function value of the original problem. Typically, a relaxation is achieved by bounding the complicating variables, terms, or functions in the original problem by means of under-, over-, and/or outer-estimating variables, terms, or functions.

Several forms of relaxation exist in the literature. One form is the *discrete-to-continuous relaxation* employed for solving discrete optimization problems, where discrete variables are treated as continuous variables. For instance, binary variables in a MILP are relaxed to be 0-1 continuous (Nemhauser and Wolsey, 1988). Another form is the continuous *nonconvex-to-convex relaxation* employed for solving nonconvex NLP. For example, the bilinear envelope suggested by McCormick (1976) and Al-Khayyal and Falk (1983) is widely used to relax bilinear terms in nonconvex programs. This relaxation involves replacing every occurrence of *S* in the original program by the following linear (convex) underestimators (Eqs. R1 and R2) and linear (concave) overestimators (Eqs. R3 and R4).

$$z \geq xy^L + x^L y - x^L y^L \tag{R1}$$

$$z \geq xy^U + x^U y - x^U y^U \tag{R2}$$

$$z \leq xy^L + x^U y - x^U y^L \tag{R3}$$

$$z \leq xy^U + x^L y - x^L y^U \tag{R4}$$

Since the resulting relaxation is linear and continuous, it is called as *LP relaxation* (Figure 5.1).

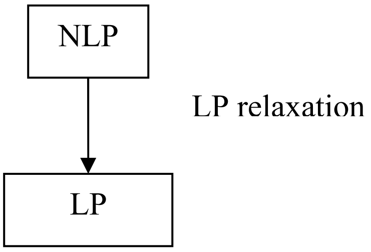

Figure 5.1. LP relaxation (McCormick, 1976) [one-level-relaxation] for bilinear programs

The quality of a relaxation is the accuracy with which a relaxation approximates the original problem and/or its solution. The closer the approximation, the *tighter* is the relaxation. An important consideration in relaxation is the size of the relaxation problem. This can be measured in terms of the numbers of variables, constraints, and nonzeros involved in the formulation. Typically, a larger problem size is needed to achieve a tighter relaxation. While solving MILPs in a branch-and-bound framework, a tighter formulation is likely to require fewer nodes, while a smaller formulation is likely to require fewer iterations for each node. Therefore, the actual computational performance of a formulation is difficult to determine a priori because of the trade-off between tightness and size.

5.3. Piecewise Relaxation

All the relaxations discussed previously are "continuous" in nature. Because a continuous convex relaxation can often be very weak or loose and may be very slow in lifting the lower bounds in a global minimization algorithm. As a remedy, several recent works (Bergamini *et al*., 2005; Karuppiah and Grossmann, 2006; Meyer and Floudas, 2006) have explored the idea of piecewise MILP relaxation, embedded inside a global optimization framework (e.g. outer approximation, spatial branch-and-bound, RLT), on several specific problems with promising results. The idea involves defining *a priori* several known partitions of the search space and combining the continuous nonconvex-to-convex relaxations of individual partitions into an overall composite relaxation. Because this involves convex relaxations of nonconvex functions over smaller regions (partitions) of the feasible region, the tightness of the overall discrete relaxation is improved as compared to the continuous relaxation over the entire feasible region. Each partition has its own distinct continuous nonconvex-to-convex relaxation and only one partition is allowed to be active at any time. Combining these individual relaxations in a seamless manner requires switching between different partitions and thus discrete decisions. Clearly, such a relaxation is discrete rather than continuous in nature and thus can be formulated as a MILP problem. Because solving the resulting MILP problem normally requires discrete-to-continuous relaxation, the overall framework of piecewise MILP relaxation comprises relaxations at two levels as shown in Figure 5.2 (compared with LP relaxation, which only has one level as shown in Figure 5.1). The first one, or the *first* (*upper*) *level relaxation*, transforms the original problem with partitioned search domain into a MILP. The second one, or the *second* (*lower*) *level relaxation*, transforms the MILP into a LP (i.e. RMILP). A complex interplay of both relaxations determines the overall efficiency of the entire framework.

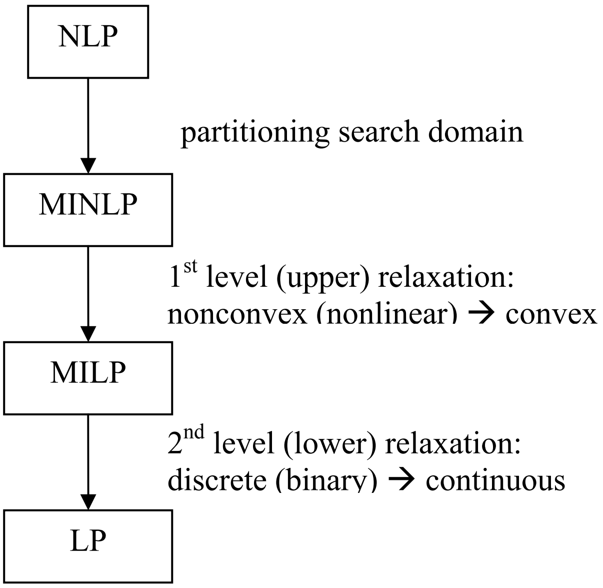
NLP
partitioning search domain
MINLP
1st level (upper) relaxation:
nonconvex (nonlinear) → convex
MILP
2nd level (lower) relaxation:
discrete (binary) → continuous
LP

Figure 5.2. Hierarchy of the piecewise MILP relaxation

(two-level-relaxation) for bilinear programs

5.4. Disjunctive Programming Models

The first step, as presented in the literature, in obtaining a piecewise MILP relaxation for a bilinear term is to define *N* partitions (Figure 5.3) of the search space in terms of N arbitrary but exhaustive segments of the range $[x^L, x^U]$. Let $\{[a(n), a(n+1)], n = 1, 2, \ldots, N\}$ denote these segments, where $a(1) = x^L$, $a(N+1) = x^U$, and $d(n) = a(n+1) - a(n) > 0$ for all *n*. Thus, the *N* search space partitions in the 2-D *xy* space are $\{[a(n), a(n+1)], [y^L, y^U]\}$ for $n = 1, 2, \ldots, N$. Clearly, each point in *S* must have its value of *x* in one of these *N* segments (or at the boundary of two adjacent segments). Then, using the convex envelope (Eqs. 5.R1 - R4) for each partition, an overall piecewise relaxation of *S* can be stated as the following special form (Bergamini *et al*., 2005) of a *disjunctive program* (Balas, 1979).

$$\bigvee_n \begin{bmatrix} W(n) \\ z \geq x \cdot y^L + a(n) \cdot y - a(n) \cdot y^L \\ z \geq x \cdot y^U + a(n+1) \cdot y - a(n+1) \cdot y^U \\ z \leq x \cdot y^L + a(n+1) \cdot y - a(n+1) \cdot y^L \\ z \leq x \cdot y^U + a(n) \cdot y - a(n) \cdot y^U \\ a(n) \leq x \leq a(n+1) \\ y^L \leq y \leq y^U \end{bmatrix} \tag{DP}$$

where *W*(*n*) is the boolean variable (“true” or “false”) indicating the status of disjunction n. The disjunctive logic OR implies that only one disjunction must hold (W(n) = “true” for exactly one n).

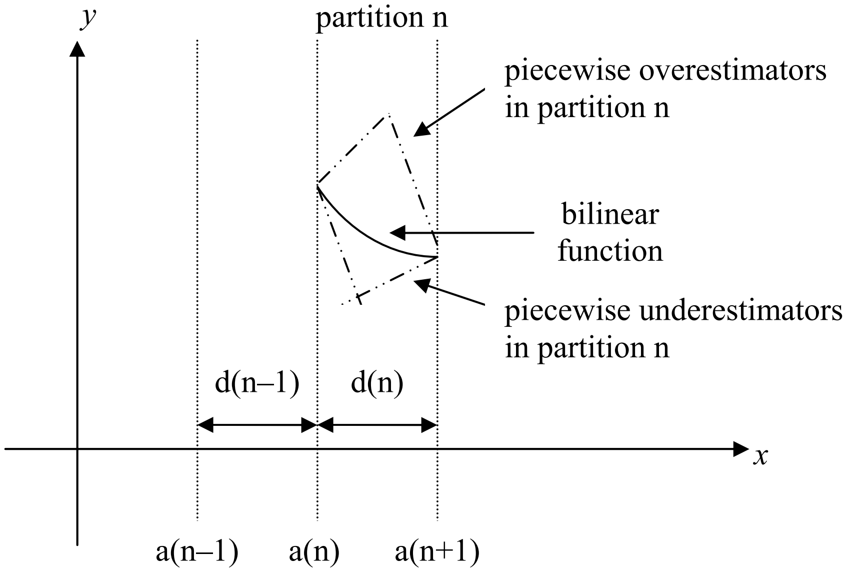

Figure 5.3. *Ab initio* partitioning of the search domain

One advantage of disjunctive programming is that it enables a systematic transformation of abstract disjunctive logic into a concrete mathematical programming model. Raman and Grossmann (1994) showed its usefulness in modeling chemical engineering problems. While several systematic methods exist for transforming a disjunctive program into a mixed-integer program, the two most common are big-M reformulation (Williams, 1985) and convex-hull reformulation (Balas, 1979; Balas, 1985; Balas, 1988). The pros and cons of these two reformulations are well known (Hooker, 2000; Vecchietti *et al.*, 2003). A big-M reformulation is generally smaller in size than a convex-hull reformulation, as it does not need additional disaggregated variables and constraints. However, its relaxation is typically poorer, as a convex-hull reformulation has proven tightness. In contrast, a convex-hull reformulation invariably needs additional disaggregated variables and constraints and is typically larger, but is at least as tight as big-M reformulation. A rigorous numerical comparison on several models is therefore required to gain the insight into the actual computational performance of competitive models.

5.4.1. Big-M Model

For the bilinear terms arising in a generalized pooling problem, Meyer and Floudas (2006) used a big-M reformulation for their piecewise MILP relaxation. Although their formulation was in the context of a specific problem, its main ideas can yield a complete big-M reformulation for DP. Such a complete formulation (BM) for an arbitrary *S* can be stated as follows.

$$\lambda(n) = \begin{cases} 1 & \text{if } a(n) \le x \le \ a(n+1) \\ 0 & \text{otherwise} \end{cases} \qquad \forall n \qquad \text{(BM-0)}$$

$$\sum_{n=1}^{N} \lambda(n) = 1 \qquad \text{(BM-1)}$$

$$x \ge a(n) \cdot \lambda(n) + x^L \cdot \left[1 - \lambda(n)\right] \qquad \forall n \qquad \text{(BM-2a)}$$

$$x \le a(n+1) \cdot \lambda(n) + x^U \cdot \left[1 - \lambda(n)\right] \qquad \forall n \qquad \text{(BM-2b)}$$

$$z \ge x \cdot y^L + a(n) \cdot (y - y^L) - M \cdot \left[1 - \lambda(n)\right] \qquad \forall n \qquad \text{(BM-3a)}$$

$$z \ge x \cdot y^U + a(n+1) \cdot (y - y^U) - M \cdot \left[1 - \lambda(n)\right] \qquad \forall n \qquad \text{(BM-3b)}$$

$$z \le x \cdot y^U + a(n) \cdot (y - y^U) + M \cdot \left[1 - \lambda(n)\right] \qquad \forall n \qquad \text{(BM-3c)}$$

$$z \le x \cdot y^L + a(n+1) \cdot (y - y^L) + M \cdot [1 - \lambda(i,n)] \qquad \forall n \qquad \text{(BM-3d)}$$

$$x^L \le x \le x^U, y^L \le y \le y^U \qquad \text{(BM-4)}$$

Note that Meyer and Floudas (2006) did not explicitly present the equivalents of Eq. BM-3b to BM-3d for their specific generalized pooling problem. Note that M is a common notation for a sufficiently large number required for Big-M reformulation.

5.4.2. Convex-Hull Model

For the bilinear terms arising in general and specific (integrated water network) process synthesis problems, Bergamini *et al.* (2005) and Karuppiah and Grossmann (2006) proposed a convex-hull reformulation. Their formulation is meant for arbitrary segment lengths [any possible arrangements of $d(n)$]; hence, it is suitable for both identical [the space between the bounds of the partitioned variables is divided into equal intervals i.e. $d(1) = ... = d(N)$] and non-identical segment lengths [i.e. the space between the bounds of the partitioned variable is divided into different intervals i.e. $d(1) \neq ... \neq d(N)$]. However, Karuppiah and Grossmann (2006) mentioned some issues with the use of non-identical segment lengths and used identical segment length exclusively in their reported examples. Although their formulation was intended for specific process synthesis problems, its main steps can be suitably modified for S in general. Then, for arbitrary segment lengths, a convex-hull formulation CH for S based on their main ideas can be stated as follows.

$$\lambda(n) = \begin{cases} 1 & \text{if } a(n) \leq x \leq a(n+1) \\ 0 & \text{otherwise} \end{cases} \tag{CH-0}$$

$$\sum_{n=1}^{N} \lambda(n) = 1 \tag{CH-1}$$

$$x = \sum_{n=1}^{N} u(n) \tag{CH-2a}$$

$$a(n) \cdot \lambda(n) \leq u(n) \leq a(n+1) \cdot \lambda(n) \qquad \forall n \tag{CH-2b}$$

$$y = \sum_{n=1}^{N} v(n) \tag{CH-3a}$$

$$y^L \cdot \lambda(n) \leq v(n) \leq y^U \cdot \lambda(n) \qquad \forall n \tag{CH-3b}$$

$$z \geq \sum_{n=1}^{N} \left[u(n) \cdot y^L + a(n) \cdot v(n) - a(n) \cdot y^L \cdot \lambda(n) \right] \tag{CH-4a}$$

$$z \geq \sum_{n=1}^{N} \left[u(n) \cdot y^U + a(n+1) \cdot v(n) - a(n+1) \cdot y^U \cdot \lambda(n) \right] \qquad \text{(CH-4b)}$$

$$z \leq \sum_{n=1}^{N} \left[u(n) \cdot y^L + a(n+1) \cdot v(n) - a(n+1) \cdot y^L \cdot \lambda(n) \right] \qquad \text{(CH-4c)}$$

$$z \leq \sum_{n=1}^{N} \left[u(n) \cdot y^U + a(n) \cdot v(n) - a(n) \cdot y^U \cdot \lambda(n) \right] \qquad \text{(CH-4d)}$$

$$x^L \leq x \leq x^U, y^L \leq y \leq y^U \qquad \text{(CH-5)}$$

5.5. Novel Models

The previous two formulations (BM and CH) for *S* will serve as the bases for evaluating several novel and superior formulations that we develop next. In contrast to the literature, we use a rather intuitive and algebraic approach for our novel formulations. The first step towards our several formulations is to model the partitioning of *x* and later, to derive the piecewise bilinear under- and overestimators (Figure 5.4).

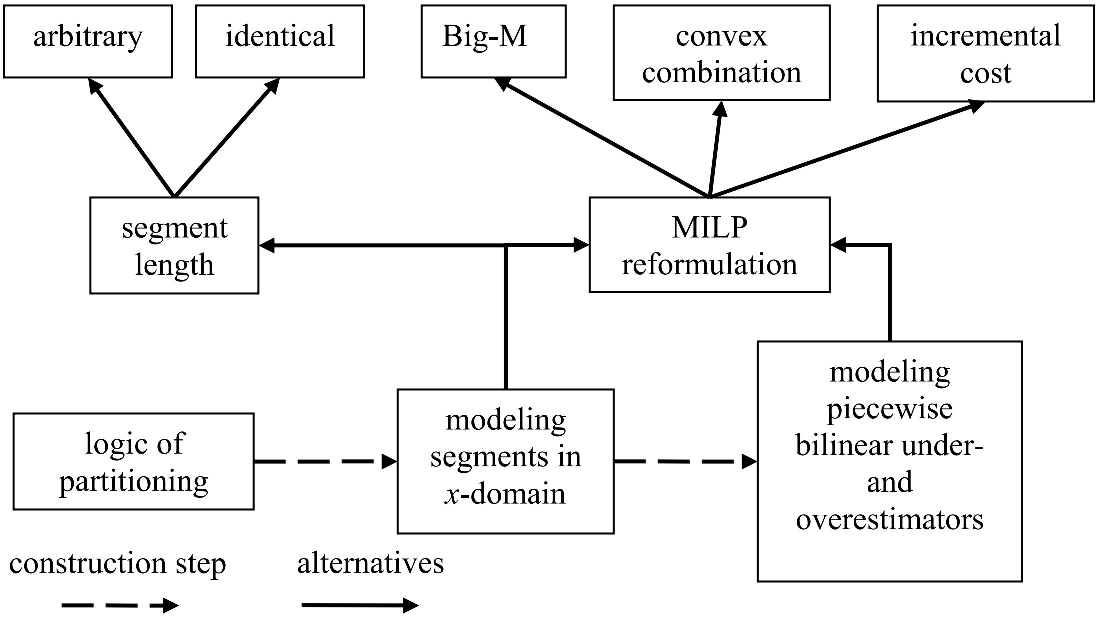

Figure 5.4.Alternatives in constructing piecewise MILP under- and overestimators for bilinear programs

Let $d(n) = a(n+1) - a(n)$ for $n = 1$ to $N-1$. It is clear that every value of x must fall in one of the N partitions. This fact has been modeled in the literature using Eqs. CH-0 and CH-1 (or BM-0 and BM-1) as discussed earlier. Using the same binary variable, we can express x in two different ways. One is to define a differential variable [$\Delta x(n)$] for each segment as follows:

$$x = \sum_{n=1}^{N} \left[a(n) \cdot \lambda(n) + \Delta x(n) \right] \qquad (1a)$$

$$0 \le \Delta x(n) \le d(n) \cdot \lambda(n) \qquad \forall n \qquad (1b)$$

The other is to aggregate the differential variables [$\Delta x(n)$] into a single differential variable [$\Delta x = \Delta x(1) + \Delta x(2) + \ldots + \Delta x(N)$] as follows.

$$x = \sum_{n=1}^{N} \left[a(n) \cdot \lambda(n) \right] + \Delta x \qquad (2a)$$

$$0 \le \Delta x \le \sum_{n=1}^{N} \left[d(n) \cdot \lambda(n) \right] \qquad (2b)$$

As far as their eventual performances in a global optimization algorithm are concerned, the differences in the above two approaches are significant. On the other hand, since Eqs. 2 can be easily derived from Eqs. 1, thus the latter cannot be tighter than the former. However, these two represent the same relaxation constructed in different variable spaces. The projections of both Eqs. 1 and 2 on the space of original variables are equivalent as can be shown easily via Fourier-Motzkin Elimination of differential variables. It is indeed critical to give utmost attention to and exploit the special structure of the piecewise under- and overestimators to develop a competitive formulation/s, because as mentioned earlier, the piecewise MILP relaxations will be solved repeatedly in a global optimization algorithm and they typically consume most of the time in each iteration. Even slight improvements will affect the overall efficiency of the global optimization algorithm, as any inefficiency in each step will propagate and eventually add up over iterations.

At this stage, it is useful to contrast our above modeling approaches (Eqs. 1 and Eqs. 2) with those (Eqs. BM-2 and CH-2) from the literature. In contrast to Eqs. CH-2, Eqs. 1 and 2 use differential variables [$\Delta x(n)$ and Δx]. While both Eqs. 1 and CH-2 disaggregate variables, Eqs. 1 disaggregate the differential variable Δx rather than x itself as done by Eqs. CH-2. This way, Eqs. 1 use N+1 constraints and Eqs. 2 use only 3 constraints as compared to 2N+1 for Eqs. CH-2 and 2N for Eqs. BM-2. Furthermore, Eqs. 1 use N+1 [x and Δx(n)] and Eqs. 2 use two variables [x and Δx] as compared to N+1 [x and $u(n)$] for Eqs. CH-2 and one (x) for Eqs. BM-2. Bilinear under- and overestimators constructed from Eqs. 1 and 2 tend to have fewer nonzeros as compared to those constructed from CH-2 and BM-2. This is because the lower bound for each differential variable is zero. These are differences in model sizes, which as we see later, do affect the quality of relaxation and overall performance significantly.

Interestingly, the following binary variable is an equivalent alternative to $\lambda(n)$ for modeling the partitioning of x.

$$\theta(n) = \begin{cases} 1 & \text{if } x \geq a(n+1) \\ 0 & \text{otherwise} \end{cases} \qquad 1 \leq n \leq (N\text{–}1) \qquad \text{(NF-0)}$$

$$\theta(n) \geq \theta(n+1) \qquad 1 \leq n \leq (N\text{–}2) \qquad \text{(NF-1)}$$

The above variable has been used in several works (Dantzig, 1963; Padberg, 2000; Oh and Karimi, 2001; Keha *et al.*, 2004) for approximating separable nonlinear functions. In particular, Padberg (2000) showed that a piecewise MILP formulation based on *θ(n)* for separable nonlinear functions has the property of *total unimodularity*, which means that the corresponding polytope has more of integral extreme points. This improves the quality of such a formulation rendering it *locally ideal* (Padberg, 2000).

Using $\theta(n)$, we can express x in two ways. The first is in terms of an incremental variable [$\Delta u(n)$] in each partition called as *local incremental variable*.

$$x = x^L + \sum_{n=1}^{N}[d(n) \cdot \Delta u(n)] \qquad 0 \le \Delta u \le 1 \tag{3a}$$

$$0 \le \Delta u(N) \le \theta(N-1) \le \Delta u(N-1) \le \theta(N-2) \le ... \le \Delta u(2) \le \theta(1) \le \Delta u(1) \le 1 \tag{3b}$$

Note that Eqs. 3b make Eq. NF-1 redundant.

The second is in terms of one incremental variable [Δx] that is common to all partitions called as *global incremental variable*.

$$x = x^L + \sum_{n=1}^{N-1}\left[d(n) \cdot \theta(n)\right] + \Delta x \tag{4a}$$

$$0 \le \Delta x \le d(1) + \sum_{n=1}^{N-1}\left[\{d(n+1) - d(n)\} \cdot \theta(n)\right] \tag{4b}$$

Similar to Eqs. 1 and 2, Eqs. 3 require more variables and constraints than Eqs. 4, thus models based on the former would be larger. On the other hand, since Eqs. 4 can be easily derived from Eqs. 3, the latter cannot be tighter than the former. However, both represent the same relaxation constructed in different variable spaces as can be trivially shown via Fourier-Motzkin Elimination of incremental variables.

Note that $\lambda(n)$, $\theta(n)$, $\Delta x(n)$, and $\Delta u(n)$ are related by,

$$\lambda(1) = 1 - \theta(1)$$

$$\lambda(n+1) = \theta(n) - \theta(n+1) \qquad\qquad n = 1 \text{ to } N\text{–}2$$

$$\lambda(N) = \theta(N-1)$$

$$\Delta x(n) = d(n) \cdot \left(\Delta u(n) + \sum_{n'=1}^{n}[\lambda(n')] - 1\right)$$

Note that we need (N–1) $\theta(n)$ variables for modeling the segments in each x-domain as compared to N $\lambda(n)$ (Figure 5.5). Furthermore, unlike $\lambda(n)$, $\theta(n)$ does not require the typical disjunctive constraint (Eq. CH-0 or BM-0), as none, one, or several

$\theta(n)$ can be one simultaneously. In this approach, the incremental variable in a given partition builds up on the variables in the preceding partitions to represent x as in Eqs. 3 and 4.

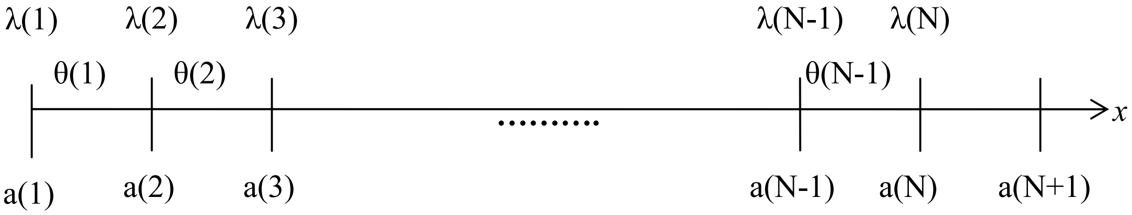

Figure 5.5. Comparison between convex combination (λ) formulation and incremental cost (θ) formulation in modeling segments in x-domain

Our approaches for modeling x defer from the existing literature in one significant manner. Instead of invoking the DP reformulation strategies behind CH and BM, Eqs. 1-4 employ rather intuitive and algebraic strategies of expressing x explicitly in terms of the basic binary variables of piecewise mixed-integer linear relaxation and new incremental variables. Using these and some other unique modeling ideas, we now develop several novel MILP formulations for the piecewise relaxation of S. We allow arbitrary partitions (arbitrary or non-identical segment lengths) first, then we assume identical segment lengths.

5.5.1. Big-M Models

The first group of our models relies on Big-M. First, we take Eqs. 1 and reformulate the continuous convex relaxation of S using the big-M constraints presented for BM. This gives us NF1, which comprises Eqs. BM-0, BM-1, 1, BM-3, and BM-4.

A straightforward alternative formulation (NF2) can be obtained by replacing Eqs. 1 in NF1 by Eqs. 2. However, note that Δx can be eliminated from Eqs. 2 to obtain,

$$x \geq \sum_{n=1}^{N} \left[a(n) \cdot \lambda(n) \right] \tag{5a}$$

$$x \leq \sum_{n=1}^{N} \left[a(n+1) \cdot \lambda(n) \right] \tag{5b}$$

Then, using Eqs. 5 in place of Eqs. 2, we get NF2. NF2 comprises Eqs. BM-0, BM-1, 5, BM-3, and BM-4.

The differences (discussed earlier) in Eqs. 1, 5, and BM-2 make NF1 and NF2 significantly different from BM. NF1 and NF2 use fewer constraints (see Table 6.1) than BM. While NF2 and BM use the same variables, NF1 uses *N* more variables. Thus, NF1 and NF2 are smaller in size. Furthermore and more importantly, we show later that both NF1 and NF2 are as tight as or tighter than BM for the same value of *M*. As stated earlier, NF2 uses far fewer variables and constraints, and is smaller than NF1. Since smaller size is often an advantage in big-M formulations, NF2 may actually outperform NF1.

5.5.2. Convex Combination Models

While NF1, NF2, and BM used the BM reformulation approach for piecewise relaxation, and CH used the CH reformulation approach; we now build on our algebraic approach to develop several novel formulations. Our second set of formulations is constructed using the *convex combination approach* (CC), which is based on the use of λ (Eq. CH-0) as binary variables and is free of big-M constraints. In this sense, CH is also a convex combination formulation.

For our first convex combination formulation (NF3), we use the following differential variables.

$$x = \sum_{n=1}^{N} \left[a(n) \cdot \lambda(n) + \Delta x(n) \right] \tag{1a}$$

$$y = y^L + \Delta y \qquad \Delta y \le y^U - y^L$$

Substituting the above equations into $z = xy$, we obtain,

$$z = y^L \cdot x + \sum_{n=1}^{N} \left[a(n) \cdot \lambda(n) \cdot \Delta y \right] + \Delta y \cdot \sum_{n=1}^{N} \Delta x(n) \tag{6}$$

The second term in the above involves products of binary and continuous variables, which we linearize exactly by defining $\Delta y(n) = \lambda(n) \cdot \Delta y$ and using,

$$y = y_L + \sum_{n=1}^{N} \Delta y(n) \tag{7a}$$

$$0 \le \Delta y(n) \le (y^U - y^L) \cdot \lambda(n) \qquad \forall n \tag{7b}$$

Using the above and Eq. CH-1, we simplify Eq. 6 to obtain,

$$z = y^L \cdot x + \sum_{n=1}^{N} \left[a(n) \cdot \Delta y(n) \right] + \left(\sum_{n=1}^{N} \Delta x(n) \right) \cdot \left(\sum_{n=1}^{N} \Delta y(n) \right) \tag{8a}$$

$$z = y^L \cdot x + \sum_{n=1}^{N} \left[a(n) \cdot \Delta y(n) \right] + \sum_{n=1}^{N} \Delta x(n) \cdot \Delta y(n) \tag{8b}$$

Note that we have successfully converted the original BLP represented by *S* into a MIBLP represented by Eqs. CH-0, CH-1, CH-5, 1 or 2, 7, and 8. However, more importantly, we have expressed *S* in terms of one or more bilinear products of differential variables instead of one bilinear product ($x \cdot y$) of original variables. Now, to convert this MIBLP into a MILP, we relax the bilinear terms in Eq. 8 using Eqs. R1 to R4. However, we have several options in this regard. We can relax any one of $\Delta x(n) \cdot \Delta y(n)$, $\Delta x \cdot \Delta y$, $\Delta x(n) \cdot \Delta y$, and $\Delta x \cdot \Delta y(n)$. Furthermore, while we must use $\Delta y(n)$, we can use either $\Delta x(n)$ or Δx as variables. Thus, we have eight possible options as follows. Of these, the relaxations of $\Delta z(n) = \Delta x(n) \cdot \Delta y$ and $\Delta z(n) = \Delta x(n) \cdot \Delta y(n)$ using Δx are not possible and the following six remain.

1. Use $\Delta x(n)$ as the variable and relax $\Delta z(n) = \Delta x(n) \cdot \Delta y(n)$.
2. Use $\Delta x(n)$ as the variable and relax $\Delta z(n) = \Delta x(n) \cdot \Delta y$.
3. Use $\Delta x(n)$ as the variable and relax $\Delta z(n) = \Delta x \cdot \Delta y(n)$.
4. Use $\Delta x(n)$ as the variable and relax $\Delta z = \Delta x \cdot \Delta y$.
5. Use Δx as the variable and relax $\Delta z = \Delta x \cdot \Delta y$.
6. Use Δx as the variable and relax $\Delta z(n) = \Delta x \cdot \Delta y(n)$.

Note that $\Delta z(n) \geq 0$ and $\Delta z \geq 0$. Now, to use Eqs. R1 to R4 for the above options, we need the bounds of $\Delta x(n)$, $\Delta y(n)$, Δx, and Δy. Because the lower bounds for all are zero, Eq. R1 becomes redundant, and Eqs. R2 to R4 simplify as follows.

$$z \geq y^U x + x^U y - x^U y^U \tag{9a}$$

$$z \leq x^U y \tag{9b}$$

$$z \leq y^U x \tag{9c}$$

This is also one significant difference between our approach and those in the literature. By transforming the lower bounds of all variables involved in the construction of the under- and overestimators for the bilinear term to zero, we reduce the size of the piecewise MILP relaxation problem in terms of both constraints and nonzeros.

From Eqs. 1b, 2b, and 7b, we identify the upper bounds of $\Delta x(n)$, $\Delta y(n)$, Δx, and Δy as $\Delta a(n)\cdot\lambda(n)$, $(y^U - y^L)\cdot\lambda(n)$, $\sum_{n=1}^{N}\left[\Delta a(n)\cdot\lambda(n)\right]$, and $(y^U - y^L)$ respectively. Using them, we now relax $\Delta z(n) = \Delta x(n) \cdot \Delta y(n)$. Substituting $\Delta z(n)$ for z, $\Delta x(n)$ for x, $\Delta y(n)$ for y, $\Delta a(n)\cdot\lambda(n)$ for x^U, and $(y^U - y^L)\cdot\lambda(n)$ for y^U in Eq. 10 and simplifying, we obtain our next formulation (NF3). NF3 comprises Eqs. CH-0, CH-1, CH-5, 1, 7, NF3-1, and NF3-2.

$$z = y^L \cdot x + \sum_{n=1}^{N}\left[a(n)\cdot \Delta y(n)\right] + \sum_{n=1}^{N}\Delta z(n) \qquad \forall n \qquad \text{(NF3-1)}$$

$$\Delta z(n) \le (y^U - y^L)\cdot \Delta x(n) \qquad \forall n \qquad \text{(NF3-2a)}$$

$$\Delta z(n) \le d(n)\cdot \Delta y(n) \qquad \forall n \qquad \text{(NF3-2b)}$$

$$\Delta z(n) \ge (y^U - y^L)\cdot[\Delta x(n) - d(n)\cdot \lambda(n)] + d(n)\cdot \Delta y(n) \qquad \forall n \qquad \text{(NF3-2c)}$$

NF3 is a novel formulation. In contrast to CH, NF3 relaxes the bilinear product [Δ*x*(*n*)·Δ*y*(*n*)] of differential and disaggregated variables rather than (*x*·*y*) itself as in CH. This may make NF3 as tight as or tighter than CH.

Interestingly, the relaxations of $\Delta z(n) = \Delta x(n)\cdot \Delta y$ and $\Delta z(n) = \Delta x \cdot \Delta y(n)$ using Δ*x*(*n*) and Δ*y*(*n*) as variables also lead to NF3, making the first three options listed earlier for relaxation identical. For option 4, i.e. the relaxation of $\Delta z = \Delta x \cdot \Delta y$ using Δ*x*(*n*) as the variable, we get Eqs. CH-0, CH-1, CH-5, 1, 7, NF4-1, and NF4-2 as an alternate formulation (NF4).

$$z = y^L \cdot x + \sum_{n=1}^{N}\left[a(n)\cdot \Delta y(n)\right] + \Delta z \qquad \text{(NF4-1)}$$

$$\Delta z \le (y^U - y^L)\cdot \sum_{n=1}^{N}\Delta x(n) \qquad \text{(NF4-2a)}$$

$$\Delta z \le \sum_{n=1}^{N} d(n)\cdot \Delta y(n) \qquad \text{(NF4-2b)}$$

$$\Delta z \ge (y^U - y^L)\cdot\left[x - \sum_{n=1}^{N-1}[a(n+1)\cdot \lambda(n)]\right] + \sum_{n=1}^{N}[d(n)\cdot \Delta y(n)] \qquad \text{(NF4-2c)}$$

However, note that using Δ*x* as a variable instead of Δ*x*(*n*) can simplify the above considerably. Furthermore, this is exactly what option 5 gives us too. Thus, options 4 and 5 both give us NF4, which comprises Eqs. CH-0, CH-1, CH-5, 2, 7, NF4-1, NF4-2b, NF4-2c, and NF4-3.

$$\Delta z \le (y^U - y^L)\cdot \Delta x \qquad \text{(NF4-3)}$$

For the last option of relaxation, namely using Δx as the variable to relax $\Delta z(n) = \Delta x \cdot \Delta y(n)$, we find that the model is nonlinear, unless we use $\Delta x(n)$ as a variable. And, if we do use $\Delta x(n)$, then it just leads to an earlier model. Thus, we have exhausted all the options of relaxation.

Note that applying the Theorem of Balas (1985) to (DP), another formulation called as **TCH**, which cannot be looser than CH, can be constructed. **TCH** comprises of Eq. (CH-0) - (CH-3) and (TCH-1) - (TCH-2). Later, we discuss the connection between **CH** and **TCH**. Obviously, **TCH** belongs to the class of *convex combination* formulations.

$$z = \sum_{n=1}^{N} w(n) \tag{TCH-1}$$

$$w(n) \geq u(n) \cdot y^L + a(n) \cdot [v(n) - y^L \cdot \lambda(n)] \qquad \forall n \tag{TCH-2a}$$

$$w(n) \geq u(n) \cdot y^U + a(n+1) \cdot [v(n) - y^U \cdot \lambda(n)] \qquad \forall n \tag{TCH-2b}$$

$$w(n) \leq u(n) \cdot y^L + a(n+1) \cdot [v(n) - y^L \cdot \lambda(n)] \qquad \forall n \tag{TCH-2c}$$

$$w(n) \leq u(n) \cdot y^U + a(n) \cdot [v(n) - y^U \cdot \lambda(n)] \qquad \forall n \tag{TCH-2d}$$

In Appendix, we show that all fomulations that belong to the class of convex combination have equivalent discrete-to-continuous tightness. We also show that their 2nd level relaxations have a direct relationship with the bilinear envelope. However, in terms of model size, **NF4** is clearly more attractive than **NF3**, **CH**, and **TCH**.

5.5.3. Incremental Cost Models

Our third approach employs the use of θ (Eq. NF-0) as binary variables and is called as *incremental cost approach* (IC) due to its incremental nature as described previously. First, we use the differential variable in Eq. 3a.

$$x = x^L + \sum_{n=1}^{N} [d(n) \cdot \Delta u(n)] \qquad 0 \leq \Delta u(n) \leq 1 \tag{3a}$$

Multiplying by y and defining $\Delta w(n) = \Delta u(n)\cdot\Delta y$ give us,

$$z = x^L \cdot y + y^L \cdot (x - x^L) + \sum_{n=1}^{N} d(n)\cdot \Delta w(n) \qquad \text{(NF5-1)}$$

From Eq. 3b, we identify the bounds of [$\theta(1)$, 1] for $\Delta u(1)$, [$\theta(n)$, $\theta(n\text{–}1)$] for $\Delta u(n)$ from n=2 to n=N–1, and [0, $\theta(N\text{–}1)$] for $\Delta u(N)$. Using these and the bounds of [0, y^U–y^L] for Δy in Eqs. R1-R4, and defining $\Delta v(n) = \theta(n)\cdot\Delta y$ for n<N, we obtain,

$$\Delta w(n) \ge \Delta v(n) \qquad \forall n < N \qquad \text{(NF5-2a)}$$

$$\Delta w(1) \ge (y^U - y^L)\cdot \Delta u(1) + y - y^U \qquad \text{(NF5-2b)}$$

$$\Delta w(n) \ge (y^U - y^L)\cdot [\Delta u(n) - \theta(n-1)] + \Delta v(n-1) \qquad \forall n > 1 \qquad \text{(NF5-2c)}$$

$$\Delta w(1) \le y - y^L \qquad \text{(NF5-2d)}$$

$$\Delta w(n) \le \Delta v(n-1) \qquad \forall n > 1 \qquad \text{(NF5-2e)}$$

$$\Delta w(N) \le (y^U - y^L)\cdot \Delta u(N) \qquad \text{(NF5-2f)}$$

$$\Delta w(n) \le (y^U - y^L)\cdot [\Delta u(n) - \theta(n)] + \Delta v(n) \qquad \forall n < N \qquad \text{(NF5-2g)}$$

To linearize the bilinear product $\Delta v(n) = \theta(n)\cdot\Delta y$ for n<N, we use the bounds of [Δu(n+1), Δu(n)] for $\theta(n)$ from Eq. 3b and [0, y^U–y^L] for Δy in Eqs. R1-R4 to obtain Eqs. NF5-2a, NF5-2c, NF5-2e, and NF5-2g. Thus, no additional constraints are required for linearizing the bilinear product $\Delta v(n)$. Now, multiplying Eq. 3b by Δy gives us,

$$0 \le \Delta w(N) \le \Delta v(N-1) \le \Delta w(N-1) \le \Delta v(N-2) \le ... \le \Delta w(2) \le \Delta v(1) \le \Delta w(1) \le y - y^L \qquad \text{(3c)}$$

Interestingly, Eqs. 3c are identical to Eqs. NF5-2a, NF5-2d, and NF5-2e. Thus, our new formulation (NF5) comprises Eqs. NF-0, CH-5, 3a-b, NF5-1, and NF5-2.

Note that the need for $\Delta v(n) = \theta(n)\cdot\Delta y$ for n<N in NF5 arose, because we used the tightest possible bounds of $\Delta u(n)$ in terms of $\theta(n)$ from Eq. 3b. If we use the looser bounds of [0, 1] for $\Delta u(n)$, then we get the following in place of eq. NF5-2.

$$\Delta w(n) \le y - y^L \qquad \forall n \qquad \text{(NF6-1a)}$$

$$\Delta w(n) \le (y^U - y^L) \cdot \Delta u(n) \qquad \forall n \qquad \text{(NF6-1b)}$$

$$\Delta w(n) \ge (y^U - y^L) \cdot \Delta u(n) + y - y^U \qquad \forall n \qquad \text{(NF6-1c)}$$

Thus, our next formulation (NF6) comprises Eqs. NF-0, CH-5, 3a-b, NF5-1, and NF6-1. In addition, we can use the following from Eq. 3c.

$$0 \le \Delta w(N) \le \Delta w(N-1) \le ... \le \Delta w(2) \le \Delta w(1) \le y - y^L \qquad \text{(NF6-2)}$$

Note that Eq. NF6-2 makes Eq. NF6-1a redundant, and NF6-2 is already included inside NF5 through Eq. 3c.

In contrast to all previous formulations (CH, BM, and NF1 - NF4), NF5 and NF6 use one fewer binary variable for each bilinear term *z* and share the advantageous property mentioned earlier. Moreover, unlike the convex combination formulations presented previously, these incremental cost formulations do not require the disaggregation of *y*. It is clear that NF5 is as tight as or tighter than NF6, but NF6 uses fewer variables and constraints. Nevertheless, further study (see Appendix) shows that all the projected feasible regions of these formulations in the space of variables {*x*, *y*, *z*} are equivalent.

For our next formulation, we use the following global incremental variable from Eq. 4a, which is common for all partitions. The use of such variable makes NF7 contains less continuous variables than NF5 and NF6.

$$x = x^L + \sum_{n=1}^{N-1} \left[d(n) \cdot \theta(n) \right] + \Delta x \qquad \text{(4a)}$$

Using the above and defining $\Delta v(n) = \theta(n) \cdot \Delta y$ and $\Delta w = \Delta x \cdot \Delta y$, we obtain,

$$z = y^L \cdot x + x^L \cdot y - x^L \cdot y^L + \sum_{n=1}^{N-1} [d(n) \cdot \Delta v(n)] + \Delta w \qquad \text{(NF7-1)}$$

For linearizing $\Delta v(n) = \theta(n)\cdot\Delta y$, we use the bounds [$\theta(2)$, 1] for $\theta(1)$, [$\theta(n+1)$, $\theta(n–1)$] for $\Delta\theta(n)$ with n from 2 to $N–1$, [0, $\theta(N–1)$] for $\theta(N)$, and [0, $yU–yL$] for Δy in Eqs. R1-R4 to obtain,

$$0 \le \Delta v(N-1) \le \Delta v(N-2) \le \ldots \le \Delta v(2) \le \Delta v(1) \le y - y^L \qquad \text{(NF7-2a)}$$

$$\Delta v(1) \ge (y^U - y^L)\cdot\theta(1) + y - y^U \qquad \text{(NF7-2b)}$$

$$\Delta v(n) \ge (y^U - y^L)\cdot[\theta(n) - \theta(n-1)] + \Delta v(n-1) \qquad 2 \le n \le N-2 \qquad \text{(NF7-2c)}$$

$$\Delta v(N-1) \le \theta(N-1)\cdot\left[y^U - y^L\right] \qquad \text{(NF7-2d)}$$

Finally, to linearize $\Delta w = \Delta x\cdot\Delta y$, we use the bounds

$[0, d(1) + \sum_{n=1}^{N-1}\left[\{d(n+1) - d(n)\}\cdot\theta(n)\right]]$ for Δx and $[0, y^U–y^L]$ for Δy in Eqs. 9. This gives us,

$$\Delta w \le d(1)\cdot(y - y^L) + \sum_{n=1}^{N-1}\left[\left\{d(n+1) - d(n)\right\}\cdot\Delta v(n)\right] \qquad \text{(NF7-3a)}$$

$$\Delta w \le \left(y^U - y^L\right)\cdot\Delta x \qquad \text{(NF7-3b)}$$

$$\Delta w \ge \Delta x\cdot(y^U - y^L) + d(1)\cdot(y - y^U) + \sum_{n=1}^{N-1}\left[\left\{d(n+1) - d(n)\right\}\cdot\left\{\Delta v(n) - (y^U - y^L)\cdot\theta(n)\right\}\right] \qquad \text{(NF7-3c)}$$

NF1-NF7 all allowed arbitrary partitions of *x*. As mentioned earlier, the previous works (Karuppiah and Grossmann, 2006; Meyer and Floudas, 2006) found it easier to use identical rather than arbitrary segments. However, they did not fine-tune their formulations for the special and simplified case of identical segments. We now develop three such formulations and show that such tailoring does indeed lead to a more compact formulation, which can have significant effect on computational performance.

5.5.4. Models with Identical Segment Length

We use one single segment length (d) for all partitions. Thus,

$d(n) = d = (x^U - x^L) / N$

$$a(n) = x^L + (n-1)\cdot d \qquad \forall n < N \qquad (11)$$

One major consequence of using identical segments is that we need not use N differential variables [$\Delta x(n)$] any more. One single Δx and Eqs. 2 are sufficient without compromising tightness. With this, Eqs. 2 reduce to,

$$x = x^L + d\cdot\sum_{n=1}^{N}\left[(n-1)\cdot\lambda(n)\right] + \Delta x \qquad (12a)$$

$$0 \le \Delta x \le d \qquad (12b)$$

We right away reduce one constraint, as Eq. 12b, in contrast to Eq. 2b, is just a bound. Substituting from Eqs. 11 and 12a into Eq. 9b, and relaxing $\Delta z(n) = \Delta x\cdot\Delta y(n)$ as done previously forces us to use $\Delta x(n)$ again to avoid nonlinearity. In other words, our first formulation (NF8) for identical segments is nothing but NF3 simplified for identical segments. It comprises Eqs. CH-0, CH-1, CH-5, 12b, 7, NF3-2a, NF8-1, NF8-2, and NF8-3.

$$x = x^L + \sum_{n=1}^{N}\left[d\cdot(n-1)\cdot\lambda(n) + \Delta x(n)\right] \qquad (\text{NF8-1})$$

$$z = y^L\cdot x + x^L\cdot y - x^L\cdot y^L + d\cdot\sum_{n=1}^{N}\left[(n-1)\cdot\Delta y(n)\right] + \sum_{n=1}^{N}\Delta z(n) \qquad (\text{NF8-2})$$

$$\Delta z(n) \le d\cdot\Delta y(n) \qquad \forall n \qquad (\text{NF8-3a})$$

$$\Delta z(n) \ge (y^U - y^L)\cdot\Delta x(n) + d\cdot\Delta y(n) - d\cdot(y^U - y^L)\cdot\lambda(n) \qquad \forall n \qquad (\text{NF8-3b})$$

Then, by relaxing $\Delta x\cdot\Delta y$ rather than $\Delta x\cdot\Delta y(n)$, we obtain our second formulation (NF9) for identical segments. It comprises Eqs. CH-0, CH-1, CH-5, 7, NF4-3, NF9-1, and NF9-2.

$$z = y^L \cdot x + x^L \cdot y - x^L \cdot y^L + d \cdot \sum_{n=1}^{N} \left[(n-1) \cdot \Delta y(n) \right] + \Delta z \tag{NF9-1}$$

$$\Delta z \le d \cdot (y - y^L) \tag{NF9-2a}$$

$$\Delta z \ge (y^U - y^L) \cdot \Delta x + d \cdot (y - y^U) \tag{NF9-2b}$$

Note that NF9 uses fewer constraints and variables than NF8, but NF8 should be as tight as or tighter than NF9.

Our third formulation (NF10) for identical segments is obtained from NF7, where we can set $d(n+1) - d(n) = 0$. In this formulation, Eq. 4b reduces to Eq. 12b, Eqs. NF7-3 reduce to Eqs. NF4-3 and NF9-2, and Eq. 4a becomes,

$$x = x^L + d \cdot \sum_{n=1}^{N-1} \theta(n) + \Delta x \tag{NF10-1}$$

Therefore, NF10 comprises Eqs. 12b, NF-1, NF4-3, NF7-1, NF7-2, NF9-2, and NF10-1.

While the relative sizes and tightness of BM, CH, TCH, and NF1-NF10 are clear (see Table 6.1), it is difficult to predict the best of them in overall computational efficiency. Therefore, it is necessary for us to evaluate them numerically on a variety of problems. However, they all possess stronger discrete-to-continuous relaxations as compared to BM.

Chapter 6

COMPUTATIONAL AND THEORETICAL STUDIES ON PIECEWISE UNDER- AND OVERESTIMATORS FOR BILINEAR PROGRAMS

6.1. Case Studies

We use two case studies from the literature to derive three test problems (Examples 1, 2a, and 2b) for a comprehensive numerical comparison of the effectiveness of various models (BM, CH, TCH, and NF1-NF10). The first case study, from which we derive Example 1, is from Karuppiah and Grossmann (2006). It involves integrated water network synthesis. The second, from which we derive Examples 2a and 2b, is from page 44-46 of Floudas *et al.* (1999). It involves the sequencing of distillation columns for non-sharp separation of a 3-component mixture (propane, isobutene, and n-butane) into two products. Since a complete global optimization algorithm is not the focus of this paper, we restrict ourselves to the lower-bounding problem at the root node only. Thus, our approach in this work is to embed the various models (BM, CH, TCH, and NF1-NF10) within the respective mathematical programming formulations used by the two case studies in the literature and solve for lower bounds at the root nodes. Since the reader can refer the original references for full details on the two test case studies, we mention only those details that are different and/or essential for an adequate understanding of this work.

A fair, well-planned, extensive, and comprehensive procedure is essential (Karimi *et al*., 2004) for a reliable assessment of MILP models based on a numerical study. To achieve a solid comparison, we solve the three test problems (Examples 1, 2a, and 2b) for several numbers of partitions and several sets of grid-point positions. For generating the latter in a convenient manner, we use the following.

$$a(n) = x^L + \left(\frac{n-1}{N}\right)^{\gamma} \cdot (x^U - x^L) \qquad \gamma \geq 0,\ \forall n < N$$

$$a(N) = x^U$$

$$d(n) = \left[\left(\frac{n}{N}\right)^{\gamma} - \left(\frac{n-1}{N}\right)^{\gamma}\right] \cdot \left(x^U - x^L\right) \qquad \forall n$$

As $\gamma \rightarrow 1$, the interior grid points (those except the first $a(1) = x^L$ and last points $a(N) = x^U$) become equally distributed, and as $\gamma \rightarrow 0$ (∞), they move towards x^U (x^L). Thus, $\gamma = 1$ corresponds to the case of identical segment lengths.

For all runs, we used a Dell Precision PW690 workstation with 3 GHz Intel Xeon single CPU, 64 GB RAM, and Windows XP Professional x64 as the operating system. GAMS 22.2 / CPLEX 10 was used for all the LP and MILP problems, and GAMS 22.2 / CONOPT 3 for all the NLP problems. The relative gap tolerance was set to zero in all cases to ensure solution optimality. We used $M = (x^U - x^L) \cdot (y^U - y^L)$ in all big-M constraints for a bilinear product xy.

6.1.1. Integrated Water Sytems Design Problem

We use Example 4 from Karuppiah and Grossmann (2006) as the basis for our Example 1. For referencing the details in the original case study, we use KG to denote Karuppiah and Grossmann (2006). Thus, we call the original case study as Example KG-4, and use the same notation for equations, figures, tables, and sections in Karuppiah and Grossmann (2006). Example KG-4 is the largest problem in the study of Karuppiah and Grossmann (2006), and was represented as having industrial scale by them. It involves five process units using water, three water-treatment units, and three contaminants. The problem was represented (Fig. KG-18) as a superstructure similar to that in Takama *et al.* (1980). The nodes in the superstructure are mixers, splitters, water-using processes, and water-treatment plants, and the arcs are the streams connecting the units. The mathematical programming formulation proposed by Karuppiah and Grossmann (2006) employs flow and composition variables for each stream (arc), and total (Eq. KG-2, KG-4, KG-6, and KG-8) and component mass balances (Eq. KG-5, KG-7, and KG-9) for all unit (node). All balances are linear except the component mass balances (Eq. KG-3) for mixers, which are bilinear. Tables KG-7 and Table KG-8 in section KG-7.4 list all the numerical data for this Example.

The problem is a BLP, where nonconvexities are due to the mixing of water streams with different compositions. It contains 348 variables, 312 constraints, and 234 bilinear terms (Table KG-9a). While Example KG-4 included concave univariate terms describing the size effect of plant design in the objective function (Eq. KG-1b), we use a linear objective function in our study, as such nonconvexities are not the primary focus of this study. Thus, our objective in Example 1 is to minimize the total amount of fresh water usage and wastewater treated (Eq. KG-1a). This is the only difference between Examples KG-4 and Example 1.

As for the solution algorithm, we use the non-redundant bound-strengthening cuts (Eq. KG-15), the logical cuts (Eq. KG-16), the akin bound-contraction pre-processing procedure (Step 1 of Section KG-6), and the partitioning of the total flow rate variables as done by Karuppiah and Grossmann (2006).

For this example, we used $N = 2$, $N = 3$, and $N = 4$ for each of BM, CH, TCH, and NF1-NF10 and ten different sets of grid-point positions ($\gamma = 0.25$, 0.50, 0.75, 1.00, 1.50, 2.00, 2.50, 3.00, 3.50, and 4.00) for each N.

6.1.2. Non-sharp Distillation Column Sequencing Problem

The case study from page 44-46 of Floudas *et al.* (1999) is also a BLP with nonconvexities due to the products of flow rates and compositions. It comprises 24 variables, 18 constraints, and 12 bilinear terms. In this case, we partition along all flow rate variables. We use two sets of variable bounds in this case study. In Example 2a, we use the variable bounds (upper and lower) reported by Floudas *et al.* (1999). In Example 2b, we contract the upper bounds on all flow rate variables to 180 kgmol/h and use a new lower bound of 10 kgmol/h for the flow rate of stream 18. The bounds on other variables were kept unchanged. Note that these new bounds still contain the global optimum of the original problem as reported.

For Examples 2a and 2b, we used $N = 10$, $N = 12$, and $N = 15$ for each of BM, CH, TCH, and NF1-NF10 and the same ten sets of grid-point positions ($\gamma = 0.25$, 0.50, 0.75, 1.00, 1.50, 2.00, 2.50, 3.00, 3.50, and 4.00) for each N. We used larger numbers of partitions for these examples, because we wanted to examine the effects of large numbers of segments. This particular case study made it possible for us to do this, because it is much smaller than Example 1. Having larger numbers of segments also magnifies the differences in the computational performances of the various models, which makes model ranking easier and more reliable.

6.2. Computational Performance Analysis

For evaluating the computational performance of various models, we use relative rather than absolute solution times to eliminate the effect of problem-to-problem variation. This enables us to compare several different formulations across a variety of test problems with different numbers of segments, grid-point positions, variable bounds, and problem structures. As suggested by several researchers (Bixby, 2004; Liu and Karimi, 2007), *geometric mean relative rank* (*GMRR*) is a useful measure in this regard. As suggested by Liu and Karimi (2007), *GMRR* can also be used to obtain relative model ranks for not just solution times, but also other criteria such as numbers of binary variables, continuous variables, constraints, nodes, as well as optimal MILP and RMILP values. *GMRR*(*m*,*c*) for a formulation or model m and criterion *c* is defined as:

$$GMRR(m,c) = \sqrt[p]{\prod_{p=1}^{P} \frac{C(m,p)}{C^{*}(p)}}$$

Where, *p* refers to a test problem, *P* is the total number of problems, *C*(*m*,*p*) is the value of criterion *c* for problem *p*, and $C^{*}(p)$ is the best value of criterion *c* across all models used to solve problem *p*. We use the minimum as the best for criteria such as solution times, and numbers of binary variables, continuous variables, and constraints, while maximum as the best $C^{*}(p)$ for MILP and RMILP values. We set 4000 CPU s as the maximum solution time for each run. If a model fails to attain the global optimal solution within 4000 CPU s, then we take its solution time as 8000 CPU s.

Table 6.1 clearly shows that formulations differ significantly in computational performance. Since the lower-bounding problem is typically the most time-consuming step in each iteration and is solved repeatedly, even a slight improvement in the computational efficiency of a lower-bounding procedure can significantly affect the overall efficiency of a global minimization algorithm (e.g. outer-approximation, spatial branch-and-bound). Based on Table 6.1, NF4, NF6 and NF7 seem to be the best among the models with arbitrary segment lengths. These three formulations have CPU time *GMRR*s of 1.33, 1.73, and 1.23, respectively, which are significantly better than those of the existing formulations from the literature [*GMRR*(BM, CPU time) = 15.05, *GMRR*(CH, CPU time) = 2.35]. However, NF9 and NF10 offer even better performance although restricted to cases involving only identical segment length [*GMRR*(NF9, CPU time) = 1.21, *GMRR*(NF10, CPU time) = 1.21]. On the other hand, the big-M based formulations (BM, NF1-NF2) are not at all competitive in terms of solution efficiency, having CPU time *GMRR*s of 9.26 to 15.05. TCH [*GMRR*(TCH, CPU time) = 4.34] which theoretically represents the true convex hull 2nd level relaxation generally performs better than BM formulations although less competitive compared to other formulations.

Table 6.1. Characteristics and *GMRR*s of various model/performance criteria of various models

Feature or criterion	BM	NF1	NF2	CH	TCH	NF3	NF4	NF5	NF6	NF7	NF8	NF9	NF10
model type	BM			CC				IC			CC		IC
type of segments	Arbitrary										Identical		
*GMRR*s of models for various model / performance criteria													
CPU time (s)	15.05	10.32	9.26	2.35	4.34	2.93	1.33	2.21	1.73	1.23	1.54	1.21	1.21
Nonzeros	2.11	2.14	1.96	2.15	2.65	2.11	1.41	1.97	1.53	1.62	1.91	1.00	1.27
constraints	2.49	2.20	1.88	2.20	3.56	2.38	1.05	2.49	2.18	1.65	2.00	1.00	1.60
binary variables	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.00	1.00	1.00	1.25	1.25	1.00
continuous variables	1.00	3.22	1.00	5.53	7.81	7.81	4.14	7.28	5.53	3.70	7.81	4.14	3.70
Nodes	37.28	16.25	22.56	2.40	3.11	2.77	2.25	1.66	1.67	1.23	1.65	1.83	1.49
MILP objective	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
RMILP objective	0.89	0.90	0.90	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Note: BM = Big-M; CC = Convex Combination; IC = Incremental Cost

All convex combination and incremental cost formulations provide the same lower bound for the 2nd level relaxation; they all have a *GMRR*(*m*, RMILP) = 1. In other words, their RMILP values are identical to the value for LP relaxation. Further study (see Appendix) shows that the 2nd level relaxation of the convex combination and incremental cost formulations recovers the convex envelope of the bilinear terms in the continuous space of original variables $\{x, y, z\}$. This issue is discussed in the following section (see Remark and Appendix). Since all these convex combination and incremental cost models exhibit similar tightness in the 2nd level relaxation, their size plays an important role in the ease of solving them. TCH, NF3, and NF5, do not perform better due to their considerably larger sizes, while having similar 2nd level relaxation quality as those of smaller models, i.e. NF4, NF6, and NF7 (see Table 1). This fact also serves as the incentive towards the use of identical segment length, since identical segment length formulations offer smaller size. Conversely, with their worse values of *GMRR*(*m*, RMILP), the big-M based formulations are significantly looser compared to all other formulations, which causes their poor performance. This is because the partitioned feasible regions in the piecewise relaxation are disjoint.

All incremental cost models use the fewest binary variables [*GMRR*(m,binary) = 1 vs. GMRR(m,binary) = 1.25 for others]. The models employing identical segment lengths generally use fewer nonzeros compared to their more general counterparts. For instance, *GMRR*(NF9,nonzeros) = 1 as compared to *GMRR*(NF4,nonzeros) = 1.41. In fact, NF9 has the fewest nonzeros of all models.

Nevertheless, no single model outperforms all others in all cases. For instance, in Example 1 with N = 4 and $\gamma = 4$, NF4 takes 151.8 CPU s, while NF7 takes 502.1 CPU s, although GMRR analysis ranks NF7 better than NF4 in the CPU time. Several external factors, other than model size and relaxation quality, such as the algorithm implemented inside CPLEX are equally important. This makes it even harder to predict accurately the performance of a certain formulation. However, as suggested by Liu and Karimi[49] for batch process scheduling problems, the idea of using competitive models or formulations in parallel in multi-CPU machines, may be worth exploring. In this *parallel lower bounding scheme*, several competitive formulations run together in each iteration to compute the lower bound for the global minimization problem and once one of the formulations finds the optimal 1st level relaxation solution, all other formulations are stopped and the global optimization algorithm proceeds to the next step. In that sense, it is crucial to develop several competitive models as we have done in this work. Specifically for this work, NF4, NF6, and NF7 in tandem seem the best choice for arbitrary segment lengths. Moreover, NF9 and NF10 seem the best for identical segment lengths.

6.3. Theoretical and Observed Properties

Although our main aim in this study is to develop and evaluate different formulations or models for piecewise MILP relaxation, it is useful to analyze our results further to gain some valuable insights into the properties of such relaxations. These insights may prove useful in improving existing algorithms or developing novel algorithms, which we hope to report in near future.

The main goal of piecewise relaxation is to improve the quality of overall piecewise MILP relaxation compared to the conventional LP relaxation which uses one segment. Therefore, it would be good to measure the improvement in relaxation quality by means of some metrics. Recall that the piecewise relaxation involves two levels. The 1st level involves the relaxation of a NLP into a MILP, and the 2nd level involves the relaxation of the MILP into a RMILP. To measure the extents of improvements in the qualities of these two relaxations solely due to piecewise partitioning, we define two gains, namely *piecewise gain* (*PG*) and *relaxed piecewise gain* (*RPG*) for a model *m* in a global minimization problem as follows.

$$PG(m,p,N,g)=\begin{cases}\dfrac{MILP(m,p,N,g)-LP(p)}{\left|LP(p)\right|},\text{if} \quad LP(p)\neq 0\\ MILP(m,p,N,g),\text{otherwise}\end{cases}$$

$$RPG(m,p,N,g)=\begin{cases}\dfrac{RMILP(m,p,N,g)-LP(p)}{\left|LP(p)\right|},\text{if} \quad LP(p)\neq 0\\ RMILP(m,p,N,g),\text{otherwise}\end{cases}$$

where, *p* is a NLP problem solved by model *m* that uses *N* segments and set *g* of grid-point positions, *LP*(*p*) is the optimal objective value of the LP (1-segment) relaxation of *p*, *MILP*(*m*,*p*,*N*,*g*) is the optimal objective value of the 1st level relaxation (piecewise MILP model *m*), and *RMILP*(*m*,*p*,*N*,*g*) is the optimal objective value of the 2nd level relaxation (the RMILP of *m*). For a global maximization problem, the numerators in the definitions of *PG* and *RPG* are multiplied by -1.

Table 6.2. Piecewise gains (*PG*) for various *N* (number of segments) and γ (grid positioning)

Example	*N*	γ = 0.25	γ = 0.50	γ = 0.75	γ = 1.00	γ = 1.50	γ = 2.00	γ = 2.50	γ = 3.00	γ = 3.50	γ = 4.00
1	2	0	0	0	0	0	0.021	0.045	0.063	0.075	0.084
	3	0	0	0	0	0.05	0.074	0.088	0.097	0.102	0.104
	4	0	0	0	0.028	0.074	0.093	0.103	0.107	0.108	0.108
2a	10	0	0	0	0	0.219	0.348	0.373	0.307	0.347	0.291
	12	0	0	0	0	0.304	0.403	0.406	0.408	0.355	0.314
	15	0	0	0	0	0.454	0.492	0.508	0.472	0.485	0.438
2b	10	0	0	0	0	0.188	0.189	0.214	0.18	0.185	0.136
	12	0	0	0	0.089	0.246	0.229	0.225	0.207	0.184	0.246
	15	0	0	0	0.136	0.262	0.277	0.25	0.246	0.243	0.214

Table 6.3. Relaxed piecewise gains (*RPG*) for various *N* (number of segments) and γ (grid positioning)

Example		1				2b		
γ	N	BM	NF1	NF2	N	BM	NF1	NF2
0.25	2	-0.055	-0.055	-0.055	10	-0.210	-0.177	-0.177
0.50		-0.060	-0.060	-0.060		-0.212	-0.191	-0.191
0.75		-0.064	-0.064	-0.064		-0.214	-0.202	-0.202
1.00		-0.068	-0.068	-0.068		-0.216	-0.209	-0.209
1.50		-0.075	-0.075	-0.075		-0.218	-0.216	-0.216
2.00		-0.082	-0.082	-0.082		-0.219	-0.218	-0.218
2.50		-0.087	-0.087	-0.087		-0.219	-0.219	-0.219
3.00		-0.091	-0.091	-0.091		-0.220	-0.220	-0.220
3.50		-0.094	-0.094	-0.094		-0.220	-0.220	-0.220
4.00		-0.097	-0.097	-0.097		-0.220	-0.220	-0.220
0.25	3	-0.072	-0.058	-0.058	12	-0.212	-0.178	-0.178
0.50		-0.076	-0.065	-0.065		-0.214	-0.193	-0.193
0.75		-0.080	-0.071	-0.071		-0.215	-0.204	-0.204
1.00		-0.083	-0.076	-0.076		-0.216	-0.211	-0.211
1.50		-0.089	-0.085	-0.085		-0.218	-0.217	-0.217
2.00		-0.094	-0.092	-0.092		-0.219	-0.219	-0.219
2.50		-0.097	-0.096	-0.096		-0.220	-0.219	-0.219
3.00		-0.099	-0.098	-0.098		-0.220	-0.220	-0.220
3.50		-0.100	-0.100	-0.100		-0.220	-0.220	-0.220
4.00		-0.101	-0.101	-0.101		-0.220	-0.220	-0.220
0.25	4	-0.080	-0.060	-0.060	15	-0.213	-0.180	-0.180
0.50		-0.084	-0.068	-0.068		-0.215	-0.195	-0.195
0.75		-0.087	-0.075	-0.075		-0.216	-0.206	-0.206
1.00		-0.090	-0.082	-0.082		-0.217	-0.212	-0.212
1.50		-0.094	-0.091	-0.091		-0.219	-0.218	-0.218
2.00		-0.097	-0.096	-0.096		-0.219	-0.219	-0.219
2.50		-0.099	-0.099	-0.099		-0.220	-0.220	-0.220
3.00		-0.101	-0.100	-0.100		-0.220	-0.220	-0.220
3.50		-0.101	-0.101	-0.101		-0.220	-0.220	-0.220
4.00		-0.102	-0.101	-0.101		-0.220	-0.220	-0.220

Note: Convex Combination and Incremental Cost Formulations give RPG = 0 for all cases tested.

PG = 0 and *RPG* = 0 mean no gains from the piecewise relaxation, with higher values being more desirable. Positive value of *PG* (*RPG*) indicates that the 1st (2nd) level relaxation is stronger than the LP relaxation while negative value of *PG* (*RPG*) points the other way. Table 6.2 and 6.3 show the values of *PG* and *RPG*, from which we draw the following observations.

Remark 1: $RPG(m,p,N,g) \leq 0$ irrespective of *m*, *N*, and *g* for every *p*, as $RMILP(m,p,N,g) \leq LP(p)$ $[RMILP(m,p,N,g) \geq LP(p)]$ in a global minimization (maximization) problem as proven in Theorem 1 of Appendix.

Remark 2: $RPG(m,p,N,g) = 0$ for all convex combination and incremental cost type models (TCH, CH, NF3-NF10) for all test runs in this work.

Interestingly, the bounds provided by all convex combination and incremental cost formulations in the 2nd level relaxation are the same and they are equivalent with the bounds_provided by the convex continuous envelope. Even in all the case studies performed in this work, the optimal values of all variables obtained by the 2nd level relaxation of the convex combination and incremental cost formulations are the same with the ones obtained by the LP relaxation. Further study shows that the projections of all convex combination and incremental cost formulations into the space of original variables $\{x, y, z\}$ yield the same feasible region as the one represented by the continuous convex envelope (see Appendix). This fact explains the result stated in Remark 2.

Remark 3: $RPG(m,p,N,g) < 0$ for several test runs using the big-M based models (BM, NF1, and NF2).

This is not surprising, as the big-M based models are known to give looser 2nd level relaxations than convex-hull reformulations in many problems, and indeed do so prominently in this work. Interestingly, big-M formulations can give *RPG* = 0, even when the variable bounds are very loose as seen in Example 2a. Moreover, *RPG* for BM is identical to those of NF-1 and NF-2 in Example 1 for $N = 2$ irrespective of grid-point positioning. This is possible because these three formulations become more similar for $N = 2$ compared to higher values of N.

Remark 4: *RPG*(m,p,N,g) varies with N, g, big-M values, and variable bounds for the big-M based models (BM, NF1, and NF2).

This is in contrast to the convex combination and incremental cost models (CH, NF3-NF10) that have $RPG = 1$ irrespective of m, N, g, and variable bounds for the problems tested in this work. While looser 2nd level relaxations may be expected from big-M models, it is interesting to note that their relaxations could be worsened by poor choices of N, g, big-M, and variable bounds.

Remark 5: *RPG*(NF1,p,N,g) ≥ *RPG*(BM,p,N,g) and *RPG*(NF2,p,N,g) ≥ *RPG*(BM,p,N,g).

Although all three models (BM, NF1, and NF2) use big-M constraints, the difference between BM and the other two is that the latter do not use big-M constraints for modeling the partitioning of x. As discussed previously, a MILP piecewise relaxation model for S involves two disjunctive modeling, one for partitioning x and the other for the bilinear under- and overestimators. While all three models (BM, NF1, and NF2) use the same constraints for the bilinear under- and overestimators, BM uses big-M type constraints for modeling x partitions with $a(n) - x^L$ and $x^U - a(n+1)$ as big-M values. By modeling x partitions without using the big-M constraints, NF1 and NF2 achieve better RPG values.

Remark 6: *PG*(m,p,N,g) ≥ 0 irrespective of m, N, and g for every p, as *MILP*(m,p,N,g) ≥ *LP*(p) [*MILP*(m,p,N,g) ≤ *LP*(m,p)] in a global minimization [maximization] problem as proven in Theorem 2 of Appendix.

Remark 7: For given p, N, g, and variable bounds, $PG(m,p,N,g)$ is the same for all m, but computational efficiency varies.

This is because all models have the same optimal MILP objective for any given p irrespective of N and g. It is important to note that the optimal here refers to a MILP solution with zero relative gap. However, big-M based models (BM, NF1, NF2) are far slower than the remaining models as discussed before.

Remark 8: For a given p, $PG(m,p,N,g)$ depends on N, g, and variable bounds.

Normally $PG(m,p,N,g)$ increases with larger N for given m, p, g, and variable bounds as the tightness of the 1st level relaxation improves. However, when the variable bounds are loose, as for instance in Example 2a where identical segment length was used ($\gamma = 1$), increasing N from 10 to 15 has no effect on PG (PG remains at the value of 0). Therefore, tight variable bounds are crucial. For instance, consider Example 2b, where identical segment length was used with tighter bounds, increasing N from 10 ($PG = 0$) to 15 ($PG = 0.136$) has relatively significant effect on PG. Nevertheless, PG may not improve through tighter bounds, despite the fact that the 1st level relaxation becomes tighter. This is because the variable bounds change the $LP(p)$ value as well. For instance, where $N = 10$ and $\gamma = 1.50$, $PG = 0.219$ in Example 2a (looser variable bounds), while $PG = 0.188$ in Example 2b (tighter variable bounds).

It is expected, and clear from our results, that grid-point positioning affects PG. However, more significantly, the use of identical segment length (via special formulations such as NF9-NF10) is not necessarily the best for attaining a higher PG, even though it seems more efficient for each lower bounding computation. This is evident from the results on Examples 1 and 2 in Table 6.2. For instance, examine Example 2a for $N = 15$ and grid points positioned via $\gamma = 2.50$ gives a considerably higher PG (0.508) as compared to that for identical segment length positioning ($PG =$

0). Thus, note that using identical segments need not be the best overall for a global optimization algorithm, because of the tradeoff between PG and the computational efficiency for each lower bounding problem.

Chapter 7

CONCLUSIONS

7.1. Optimization of Fuel Gas Network in a Natural Gas Liquefaction Plant

In this work, we demonstrated that our superstructure and MINLP model for fuel gas network are efficient and practically useful. Our approach is able to globally optimize the fuel gas network synthesis problem for both operational and retrofit purposes. We showed using industrial case study that our methodology can optimize the fuel gas network to significantly reduce the FFF consumption. Further reduction of FFF consumption was achieved by integrating the available jetty BOG.

7.2. Modeling Piecewise Under- and Overestimators for Bilinear Programs via Mixed-integer Linear Programming

In this paper, we presented eleven novel formulations (i.e. TCH and NF1 - NF10) for piecewise MILP under- and overestimators for global optimization of bilinear programs. These were derived using three systematic approaches: big-M, convex combination, and incremental cost, and two segmentation schemes: arbitrary and identical segment lengths. These systematic approaches and segmentation schemes can also derive the existing formulations. We compared their performance with the existing formulations in the literature (i.e. BM and CH) using two case studies arising in process network synthesis: integrated water systems synthesis and non-sharp distillation column sequencing. Based on several runs with various numbers of segments, grid-points positioning, and variable bounds, we demonstrated that our novel formulations give superior relaxation tightness as compared to existing BM formulations, especially those constructed via convex combination and incremental cost approaches. Some of them are much more compact than those existing in the literature, with no loss of relaxation quality. Note that the incremental cost formulations require one less binary variable for modeling segments in each *x*-domain as compared to other formulations. Further reduction in formulation size is achieved through the use of identical segment length. In general, our novel formulations are more efficient than all existing formulations. NF4, NF6, and NF7 in tandem seem the most competitive choice for arbitrary segment lengths, while NF9 and NF10 seem the best for identical segment length. Parallel use of convex combination and incremental cost formulations with compact size seem favorable for practical purpose. On the other hand, big-M formulations, with their considerably inferior relaxation quality, seem not competitive for piecewise relaxation in this study.

7.3. Computational and Theoretical Studies on Piecewise Under- and Overestimators for Bilinear Programs

Piecewise relaxation involves two levels. The 1st level transforms the original nonconvex NLP into a MILP. The 2nd level transforms the MILP into a LP. In this work, we introduced *PG* (piecewise gain) and *RPG* (relaxed piecewise gain) as metrics to measure the effectiveness of piecewise relaxation by comparing the tightness of the LP relaxation with those of the 1st and 2nd level relaxations, respectively. Number of segments, grid-points positioning, and variable bounds are among the factors that affect *PG* and *RPG*. The use of identical segment length is not necessarily the best for attaining a higher *PG* while *PG* does not depend on formulation. All formulations are shown and proved to have $PG \geq 0$ and $RPG \leq 0$ with $RPG = 0$ is a necessary condition for those representing convex hull type relaxation of the 2nd level relaxation. All convex combination and incremental cost formulations have $RPG = 0$.

BIBLIOGRPAHY

Adhya N, Tawarmalani M, Sahinidis NV. A Lagrangian Approach to the Pooling Problems. *Ind Eng Chem Res*. 1999;**38**:1956.

Al-Khayyal FA, Falk JE. Jointly Constrained Biconvex Programming. *Math Oper Res*. 1983;**8**:273.

Androulakis IP, Maranas CD, Floudas CA. αBB: A Global Optimization Method for General Constrained Nonconvex Problems. *Journal of Global Optimization*. 1995;**7**: 4.

Balas, E. Disjunctive Programming. *Annals of Discrete Mathematics*. 1979;**5**:3.

Balas E. Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. *SIAM Journal on Algebraic and Discrete Methods*. 1985;**6**:466.

Balas E. On The Convex Hull of The Union of Certain Polyhedra. *Operations Research Letters*. 1988;**7**:279.

Ben-Tal A, Eiger G, Gershovitz V. Global Minimization by Reducing Duality Gap. *Mathematical Programming*. 1994;**63**:193.

Bergamini ML, Aguirre P, Grossmann IE. Logic-based Outer Approximation for Globally Optimal Synthesis of Process Networks. *Comput Chem Eng*. 2005;**29**:1914.

Biegler LT, Grossmann, IE, Westerberg AW. Systematic Methods of Chemical Process Design. New Jersey, USA: Prentice Hall, 1997.

Biegler LT, Grossmann IE. Retrospective on Optimization. *Comput Chem Eng*. 2004;**28**: 1169.

Bixby R. Mixed-integer Programming – It Works Out of the Box. In: Proceedings of the Sixth International Conference on Optimization. Ballarat, Australia, December 9 – 11, 2004.

Brooke A, Kendrick D, Meeraus A, Raman R. GAMS - A User's Guide. GAMS Development Corporation, 2005.

Dantzig GB. Linear Programming and Extensions. Princeton, USA: Princeton University Press, 1963.

Edgar TF, Himmelblau DM, Lason LS. Optimization of Chemical Processes, 2001.

Floudas, CA, Pardalos PM, Adjiman CS, Esposito WR, Gumus ZH, Harding ST, Klepeis JL, Meyer CA, Schweiger CA. Handbook of Test Problems in Local and Global Optimization. Dordrecht, The Netherlands: Kluwer Academic, 1999.

Floudas CA. Deterministic Global Optimization: Theory, Methods, and Applications. Dordrecht, The Netherlands: Kluwer Academic, 2000.

Floudas CA, Pardalos PM (ed.). Frontiers in Global Optimization. Dordrecht, The Netherlands: Kluwer Academic, 2004.

Floudas CA, Akrotirianakis IG, Caratzoulas S, Meyer CA, Kallrath J. Global Optimization in the 21st Century: Advances and Challenges. *Comput Chem Eng*. 2005;**29**:1185.

Flynn, TM. Cryogenic Engineering (2nd Edition). New York, USA: Marcel Dekker, 2005.

Foulds LR, Haughland D, Jornsten K. A Bilinear Approach to the Pooling Problem. *Optimization*. 1992;**24**:165.

Greenwald GB. Liquefied Natural Gas: Developing and Financing International Energy Projects. The Hague, The Netherlands: Kluwer Law International, 1998.

Grossmann IE (ed.). Global Optimization in Engineering Design. Dordrecht, The Netherlands: Kluwer Academic, 1996.

Haverly CA. Studies of the Behaviour of Recursion for the Pooling Problem. *ACM Sigmap Bul.* 1978;**25:**19.

Hooker JN. Logic-based Methods for Optimization: Combining Optimization and Constraint Satisfaction. John Wiley and Sons, 2000.

Horst R, Tuy H. Global Optimization: Deterministic Approaches (2nd edition). Germany: Springer-Verlag, 1993.

Karimi IA, Tan ZYL, Bhushan S. An Improved Formulation for Scheduling An Automated Wet-etch Station. *Comput Chem Eng.* 2004;**29**:217.

Karuppiah R, Grossmann IE. Global Optimization for the Synthesis of Integrated Water Systems in Chemical Processes. *Comput Chem Eng*. 2006;**30**:650.

Kearfott B. Decomposition of Arithmetic Expressions to Improve the Behavior of Interval Iteration for Nonlinear Systems. *Computing*. 1991;**47**:169.

Keha AB, de Farias IR, Nemhauser GL. Models for Representing Piecewise Linear Cost Functions. *Operations Research Letters*. 2004;**32**:44.

Liu Y, Karimi IA. Scheduling Multistage, Multiproduct Batch Plants with Nonidentical Parallel Units and Unlimited Intermediate Storage. *Chemical Engineering Science*. 2007;**62**:1549.

McCormick GP. Computability of Global Solutions to Factorable Nonconvex Programs: Part I-Convex Underestimating Problems. *Mathematical Programming*. 1976;**10:**146.

Meyer CA, Floudas CA. Global Optimization of a Combinatorially Complex Generalized Pooling Problem. *AIChE Journal.* 2006;**52**:1027.

Meyer CA, Floudas CA. Convex Underestimation of Twice Continuously Differentiable Functions by Piecewise Quadratic Perturbation: Spline αBB Underestimators. *Journal of Global Optimization*. 2005;**32**:221.

Nemhauser G, Wolsey L. Integer and Combinatorial Optimization. New York, USA: John Wiley and Sons, 1988.

Neumaier A. Complete Search in Continuous Global Optimization and Constraint Satisfaction. Cambridge University Press, 2004.

Neumaier A, Shcherbina O, Huyer W, Vinko T. A Comparison of Complete Global Optimization Solvers. *Mathematical Programming*. 2005;**103**:335.

Oh HC, Karimi IA. Planning Production on a Single Processor with Sequence-dependent Setups: Part 1-Determination of Champaigns. *Comput Chem Eng*. 2001;**25**:1021.

Padberg M. Approximating Separable Nonlinear Functions via Mixed Zero-One Programs. *Operations Research Letters*. 2000;**27**:1.

Qatargas Plant Operating Manual. Personal Communication. 2006.

Quesada I, Grossmann IE. Global Optimization of Bilinear Process Networks with Multicomponent flows. *Comput Chem Eng.* 1995;**19**:1219.

Raman R, Grossmann IE. Modeling and Computational Techniques for Logic Based Integer Programming. *Comput Chem Eng*. 1994;**18**:563.

Reddy PCP, Karimi IA, Srinivasan R. Novel solution approach for optimizing crude oil operations. *AIChE Journal*. 2004;**50**:1177.

Reddy PCP, Karimi IA, Srinivasan R. A new continuous-time formulation for scheduling crude oil operations. *Chemical Engineering Science*. 2004;**59**:325.

Ryoo HS, Sahinidis NV. A Branch-and-Reduce Approach to Global Optimization. *Journal of Global Optimization*. 1996;**8**:107.

Sahinidis NV. BARON: A General Purpose Global Optimization Software Package. Journal of Global Optimization. 1996:**8**:201.

Sherali H, Alameddine A. A New Reformulation-Linearization Technique for Bilinear Programming Problems. *Journal of Global Optimization*. 1992;**2**:379.

Smith, R. Chemical Process Design. New York, USA: McGraw-Hill, 1995.

Smith EMB, Pantelides CC. A Symbolic Reformulation/Spatial Branch-and-bound Algorithm for The Global Optimisation of Nonconvex MINLPs. *Comput Chem Eng.* 1999;**23**: 457.

Takama N, Kuriyama Y, Shiroko K, Umeda T. Optimal Water Allocation in a Petrochemical Refinery. *Comput Chem Eng.* 1980;**4**:251.

Tawarmalani M, Ahmed S, Sahinidis NV. Product Disaggregation in Global Optimization and Relaxations of Rational Programs. *Journal of Global Optimization*. 2002;**3**:281.

Tawarmalani M, Sahinidis NV. Convexification and Global Optimization in Continuous and Mixed-integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. Dordrecht, The Netherlands: Kluwer Academic, 2002.

Timmerhaus KD, Reed RP (ed.). Cryogenic Engineering - Fifty Years of Progress. New York, USA: Springer, 2007.

Tuy H. Convex Analysis and Global Optimization. The Netherlands: Kluwer Academic, 1998.

Vecchietti A, Lee S, Grossmann IE. Modeling of Discrete/Continuous Optimization Problems: Characterization and Formulation of Disjunctions and Their Relaxations. *Comput Chem Eng*. 2003;**27**:433.

Wicaksono DS, Karimi IA, Alfadala H, Al-Hatou, OI. Optimization of Fuel Gas Network in an LNG Plant. In: Proceedings of AIChE Annual Meeting. San Francisco, USA, November 12 – 17, 2006.

Wicaksono DS, Karimi IA, Alfadala H, Al-Hatou OI. Integrating Recovered Jetty Boil-off Gas as a Fuel in an LNG Plant. In: Proceedings of the 17th European Symposium of Computer Aided Process Engineering. Bucharest, Romania, May 29 – 31, 2007.

Williams HP. Model Building in Mathematical Programming. John Wiley and Sons, 1985.

Zamora JM, Grossmann IE. A Branch and Contract Algorithm for Problems with Concave Univariate, Bilinear, and Linear Fractional Terms. *Journal of Global Optimization*. 1999;**14**:217.

http://www.eia.doe.gov/pub/oil_gas/natural_gas/analysis_publications/natural_gas_1998_issues_trends/pdf/chapter2.pdf

Appendix

Theoretical Results on Piecewise Under- and Overestimators for Bilinear Programs

Appendix: Theoretical Results

Let *m* denotes a piecewise MILP under- and overestimators model for bilinear term xy over rectangle $x^L \le x \le x^U, y^L \le y \le y^U$. m is based on partitioning the bilinear envelope of the original domain with arbitrary number of segments on *x* (e.g. BM, CH, TCH, and NF1-NF10 presented in this paper). Let *p* denotes a BLP problem. Let *LR*(*p*) denotes the feasible region representing the 1-segment LP relaxation [convex envelope for bilinear term represented by Eqs. (R-1) - (R-4)] of *p* in the space of variables $\{x, y, z\}$. Furthermore, let *RMR*(*m*,*p*) and *MR*(*m*,*p*) respectively denote the projected feasible regions of the 2nd and 1st level relaxations of *m* for *p* in the same space of variables $\{x, y, z\}$.

Theorem 1. *LP*(*p*) $\subseteq$ *RMR*(*m*,*p*)

Proof:

Let $u(x, y) \le z$ and $o(x, y) \ge z$ denote the continuous linear under- and overestimators for the bilinear term *xy* over rectangle $x^L \le x \le x^U, y^L \le y \le y^U$ representing the facets of *RMR*(*m*,*p*), respectively. Comparing them to the bilinear envelope (Eqs. (R-1) – (R-4)), it is clear[25] that

$$u(x, y) \le \max(x^L \cdot y + x \cdot y^L - x^L \cdot y^L, x^U \cdot y + x \cdot y^U - x^U \cdot y^U)$$

and

$$o(x,y) \geq \min\left(x^U \cdot y + x \cdot y^L - x^U \cdot y^L, x^L \cdot y + x \cdot y^U - x^L \cdot y^U\right).$$

over $x^L \leq x \leq x^U, y^L \leq y \leq y^U$. Thus, *LP*(*p*) is either contained within *RMR*(*m*,*p*) or equivalent to *RMR*(*m*,*p*).

Proposition 1. *RMR*(CC,*p*) = *LP*(*p*) [Projection of a convex combination model (CC) in the space of variables {*x*, *y*, *z*} produces the facet constraints described by Eqs. (R-1) - (R-4).]

Proof:

This can be done via Fourier-Motzkin elimination.

Consider model (CH). All *u*(*n*), *v*(1), and λ(1) from Eqs. (CH-4a) and (CH-4c) as well as all *u*(*n*), *v*(*N*) and λ(*N*) from Eqs. (CH-4b) and (CH-4d) can be eliminated by utilizing Eqs. (CH-1), (CH-2a), and (CH-3a).

$$z \geq x \cdot y^L + x^L \cdot y - x^L \cdot y^L + \sum_{n=2}^{N}\left[\left(a(n) - a(1)\right) \cdot \Delta v(n)\right] - \sum_{n=2}^{N}\left[\left(a(n) - a(1)\right) \cdot y^L \cdot \lambda(n)\right] \quad \text{(A-1)}$$

$$z \geq x^U \cdot y + x \cdot y^U - x^U \cdot y^U + \sum_{n=1}^{N-1}\left[\left(a(n+1) - a(N+1)\right) \cdot \Delta v(n)\right] - \sum_{n=1}^{N-1}\left[\left(a(n+1) - a(N+1)\right) \cdot y^U \cdot \lambda(n)\right]$$

(A-2)

$$z \le x \cdot y^L + x^U \cdot y - x^U \cdot y^L + \sum_{n=1}^{N-1} \left[a(n+1) - a(N+1) \right] \cdot \Delta v(n) - \sum_{n=1}^{N-1} \left[\left(a(n+1) - a(N+1) \right) \cdot y^L \cdot \lambda(n) \right]$$

(A-3)

$$z \le x \cdot y^U + x^L \cdot y - x^L \cdot y^U + \sum_{n=2}^{N} \left[\left(a(n) - a(1) \right) \cdot \Delta v(n) \right] - \sum_{n=2}^{N} \left[\left(a(n) - a(1) \right) \cdot y^U \cdot \lambda(n) \right] \quad \text{(A-4)}$$

Multiply both sides of $y^L \cdot \lambda(n) \le v(n)$ from Eq. (CH-2b) with $a(n) - a(1)$ and $a(n+1) - a(N+1)$.

$$\left[a(n) - a(1) \right] \cdot \Delta y(n) - \left[a(n) - a(1) \right] \cdot y^L \cdot \lambda(n) \ge 0 \qquad \forall n > 2 \qquad \text{(A-5)}$$

$$\left[a(n+1) - a(N+1) \right] \cdot \Delta v(n) - \left[a(n+1) - a(N+1) \right] \cdot y^L \cdot \lambda(n) \le 0 \qquad \forall n < N+1 \qquad \text{(A-6)}$$

Multiply both sides of $v(n) \le y^U \cdot \lambda(n)$ from Eq. (CH-2b) with $a(n+1) - a(N+1)$ and $a(n) - a(1)$.

$$\left[a(n+1) - a(N+1) \right] \cdot \Delta y(n) - \left[a(n+1) - a(N+1) \right] \cdot y^U \cdot \lambda(n) \ge 0 \qquad \forall n < N+1 \qquad \text{(A-7)}$$

$$\left[a(n) - a(1) \right] \cdot \Delta y(n) - \left[a(n) - a(1) \right] \cdot y^U \cdot \lambda(n) \le 0 \qquad \forall n > 2 \qquad \text{(A-8)}$$

The last two terms of Eqs. (A-1) – (A-4) can be eliminated by using Eqs. (A-5), (A-7), (A-6), and (A-8), respectively. The final result is equivalent with Eqs. (R-1) – (R-4). Since the latter represents the bilinear envelope, it is clear that the remaining facets generated via Fourier-Motzkin Elimination are redundant.

Consider model (TCH). Consider the following one facet of the piecewise MILP under- and overestimators of model (TCH). $\Delta z(N)$ from Eq. (TCH-2a) can be eliminated by utilizing $\Delta z(N) = z - \sum_{n=1}^{N-1} \Delta z(n)$ from Eq. (TCH-1).

$$\Delta z(n) \geq \Delta x(N) \cdot y^L + a(N) \cdot \Delta y(N) - a(N) \cdot y^L \cdot \lambda(N) \qquad \forall n < N \qquad \text{(A-9a)}$$

$$z - \sum_{n=1}^{N-1} \Delta z(n) \geq \Delta x(N) \cdot y^L + a(N) \cdot \Delta y(N) - a(N) \cdot y^L \cdot \lambda(N) \qquad \text{(A-9b)}$$

It is clear that $\Delta z(N-1)$ from Eq. (A-9a) can be eliminated by utilizing Eq. (A-9b) resulting in

$$\Delta z(n) \geq \Delta x(N) \cdot y^L + a(N) \cdot \Delta y(N) - a(N) \cdot y^L \cdot \lambda(N) \qquad \forall n < N-1 \qquad \text{(A-10a)}$$

$$z - \sum_{n=1}^{N-2} \Delta z(n) \geq \sum_{n=N-1}^{N} \Delta x(n) \cdot y^L + a(n) \cdot \Delta y(n) - a(n) \cdot y^L \cdot \lambda(n) \qquad \text{(A-10b)}$$

Repeating the same steps until the entire remaining $\Delta z(n)$ are eliminated produces Eq. (CH-4a). Using the same steps for the other three facets of (TCH) produces Eqs. (CH-4b) – (CH-4d). From this point, the next steps follow directly from those of (CH) described previously.

Consider (NF3). (NF3) is related to (TCH) via

$$u(n) = a(n) \cdot \lambda(n) + \Delta x(n) \qquad \forall n$$

$$v(n) = y^L \cdot \lambda(n) + \Delta y(n) \qquad \forall n$$

$$w(n) = y^L \cdot \Delta x(n) + a(n) \cdot \Delta y(n) + y^L \cdot a(n) \cdot \lambda(n) + \Delta z(n) \qquad \forall n$$

Thus, it is clear that the same result applies for (NF3). It can be easily shown as well that the same result also applies for the other Convex Combination models.

Corrolary. *LP*(*p*) $\subseteq$ *RMR*(BM,*p*)

Proof:

It is clear[42] that for a disjunctive program that the discrete-to-continuous relaxation quality of a model obtained from convex hull reformulation is tighter than or as tight as those of big-M. We already showed that *RMR*(CC,*p*) = *LP*(*p*) in Proposition 1. Since one model of CC can be obtained from convex hull reformulation, it is clear that *LP*(*p*) $\subseteq$ *RMR*(BM,*p*).

Proposition 2. *RMR*(IC,*p*) = *LP*(*p*) [Projection of an incremental cost model IC in the space of variables {*x*, *y*, *z*} produces the facet constraints described by Eqs. (R-1) - (R-4).]

Proof:

This can be done using Fourier – Motzkin Elimination via similar arguments used in the proof of Proposition 1.

Remark: Theorem 1 and Proposition 1 - 3 implies that *RPG*(*m*,*p*,*N*,*g*) $\leq$ 0, *RPG*(*BM*,*p*,*N*,*g*) $\leq$ 0 and *RPG*(CC,*p*,*N*,*g*) = *RPG*(IC,*p*,*N*,*g*) = 0.

<u>Theorem 2.</u> *MR*(*m*,*p*) $\subseteq$ *LP*(*p*)

Proof:

As depicted by (DP), *MR*(*m*,*p*) is defined as the bilinear envelope over partition *n** which $\lambda(n^*) = 1$ for Big-M and Convex Combination models or equivalently $\theta(1) = ... = \theta(n^*) = 1, \theta(n^*+1) = ... = \theta(N-1) = 0$ for Incremental Cost models. Hence, *MR*(*m*,*p*) is defined as the following.

$$z \geq x \cdot y^L + a(n^*) \cdot y - a(n^*) \cdot y^L$$

$$z \geq x \cdot y^U + a(n^*+1) \cdot y - a(n^*+1) \cdot y^U$$

$$z \leq x \cdot y^L + a(n^*+1) \cdot y - a(n^*+1) \cdot y^L$$

$$z \leq x \cdot y^U + a(n^*) \cdot y - a(n^*) \cdot y^U$$

$$a(n^*) \leq x \leq a(n^*+1), y^L \leq y \leq y^U$$

Since $x^L \leq a(n) \leq x^U$ for any *n* and thus $x^L \leq a(n^*) \leq x^U$ and $x^L \leq a(n^*+1) \leq x^U$, it is clear that

$$x \cdot y^L + a(n^*) \cdot y - a(n^*) \cdot y^L \geq x \cdot y^L + x^L \cdot y - x^L \cdot y^L$$

$$x \cdot y^U + a(n^*+1) \cdot y - a(n^*+1) \cdot y^U \geq x \cdot y^U + x^U \cdot y - x^U \cdot y^U$$

and

$$x \cdot y^L + a(n^*+1) \cdot y - a(n^*+1) \cdot y^L \le x \cdot y^L + x^U \cdot y - x^U \cdot y^L$$

$$x \cdot y^U + a(n^*) \cdot y - a(n^*) \cdot y^U \le x \cdot y^U + x^L \cdot y - x^L \cdot y^U$$

over $a(n^*) \le x \le a(n^*+1), y^L \le y \le y^U$. This completes the proof.

Remark: Theorem 2 implies that $PG(m,p,N,g) \geq 0$, which supports the use of piecewise MILP under- and overestimators.