Nondominated Points of Biobjective Mixed-Integer Programming Problems

by

Ali Fattahi

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koç University

August 25, 2014

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Ali Fattahi

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Dr. Metin Turkay

Assoc. Prof. Dr. Emre Alper Yildirim

Asst. Prof. Dr. Emre Mengi

Date: ______________________________

ABSTRACT

The nondominated frontier in the objective space of biobjective mixed-integer linear/nonlinear programming problems consists of points that cannot be improved in value of one of the objectives without degrading the other objective value. This frontier is usually very involved consisting of many isolated points and open, closed, or half-open/half-closed line segments or curves. Some researchers considered specific classes of these problems to reduce the complexities in nondominated frontier. Some algorithms have been also proposed to find a subset of nondominated set. Several mathematical models for nonlinear process network problems have been developed and solved using ϵ-constraint, weighted sum, and minimum distance. This thesis outlines some possible complexities in nondominated frontier of BOMILPs and drawbacks of existing algorithms, and proposes an effective algorithm, EnpoBomip, to find the exact nondominated frontier of general BOMILPs, as well as all possible values of integer variables associated with each nondominated point. We also investigate biobjective mixed-integer nonlinear problems that are formulated using generalized disjunctive programming for nonlinear network synthesis problems and propose an effective algorithm, ϵ-OA, based on augmented ϵ-constraint and logic-based outer approximation (OA). We provide theoretical characterization of the proposed algorithm and show that the solutions generated are efficient.

An experimental study is conducted to present a comparative analysis between EnpoBomip and the existing algorithms on three well-known problems, and show that our novel algorithm significantly outperforms others with respect to solution

quality and computational performance. We also illustrate the effectiveness of ϵ-OA compared to the augmented ϵ-constraint with/without OA, and the traditional ϵ-constraint. Based on the results, ϵ-OA is very effective in solving the biobjective generalized disjunctive programming problems in the synthesis of nonlinear process networks.

ÖZETÇE

Ayrık noktalar, açık/kapalı ya da yarı -açık/yarı-kapalı doğru parçası ya da eğrilerden oluşan "Nondominated frontier", iki amaçlı karışık tamsayılı doğrusal ya da doğrusal olmayan programlama (BOMILP) problemlerinde amaç alanını belirlemede sıklıkla kullanılır. Bazı araştırmacılar nondominated frontier'de bulunan karmaşıklıkları azaltmak için bu problemlerin belli türlerini incelemiştir. Bazı algoritmalar da nondominated setin altkümesini bulmayı önermiştir. Doğrusal olmayan süreçlere ve sistemlere ait pek çok matematiksel model geliştirilmiş ve bu modeller epsilon kısıtı (epsilon-constraint), ağırlıklı toplam (weighted sum) ve minimum uzaklık (minimum distance) gibi yöntemler kullanılarak çözülmüştür. Bu tez BOMILP problemlerin nondominated frontier'lerindeki bazı olası karmaşıklıkları ve var olan algoritmaların handikaplarını özetler ve genel olarak BOMILP problemlerinin nondominated frontier'i ile birlikte her bir nondominated noktaya ait tamsayılı değişkenlerin değerlerini bulmaya yarayan verimli bir algoritma (ENPOBOMIP) sunar. Bununla birlikte biz bu tezde, doğrusal olmayan network sentez problemleri için genelleştirilmiş disjonktif programlama kullanılarak formüle edilmiş iki amaçlı karışık tamsayılı doğrusal olmayan problemleri inceledik ve augmented epsilon kısıt metodunu ve "logic-based outer approximation" metodunu (OA) baz alarak epsilon-OA adıyla bir algoritma geliştirdik. Önerilen algoritmanın teorik karakterizasyonu ile birlikte elde edilen çözümlerin etkin (efficient) olduğunu gösterdik.

ENPOBOMIP ve var olan algoritmalar arasında karşılaştırmalı bir analiz sunabilmek için tanınmış üç problem üzerine deneysel bir çalışma yürüttük ve algo-

ritmamızın çözüm kalitesi ve hesaplama performansı bazında diğerlerinden daha iyi sonuçlar verdiğini gördük. Ek olarak, epsilon-OA algoritmasının verimliliğini OA içeren ve içermeyen augmented epsilon kısıt metodlarını ve de epsilon kısıt metodunu kullanarak kıyasladık. Elde ettiğimiz sonuçlarda, epsilon-OA yönteminin doğrusal olmayan proses networklerini içeren iki amaçlı genelleştirilmiş disjonktif problemleri çözmede çok verimli olduğunu gördük.

ACKNOWLEDGMENTS

I would like to express my special appreciation and thanks to my advisor Prof. Dr. Metin Turkay, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless. I would also like to thank my committee members, Prof. Dr. Emre Alper Yildirim and Prof. Dr. Emre Mengi for serving as my committee members even at hardship. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I gratefully acknowledge the financial support for my thesis by TUBITAK under grant 104M322.

A special thanks to my family. Words cannot express how grateful I am to my mother and father for all of the sacrifices that you have made on my behalf. Your prayer for me was what sustained me thus far. I would also like to thank my parents-in-law, my brothers, and friends who incented me to strive towards my goal. At the end I would like express appreciation to my beloved wife Yasaman who spent sleepless nights with and was always my support in the moments when there was no one to answer my queries.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$bigM$	a big number
BOLP	biobjective linear programming
BOMILP	biobjective mixed-integer linear programming
BOMINLP	biobjective mixed-integer nonlinear programming
$c_j, f_j, \rho_j, \gamma_i, \eta_i$	objective coefficients
ESN	extreme supported nondominated
$f_j(x), g(x), h_i(x)$	convex functions
GDP	generalized disjunctive programming
$\mathrm{LP}^{\mathrm{NEP}}$	the problem that discovers if the next-lower extreme point exists
LX(.,.)	lexicographic optimization
MOOP	multiobjective optimization problem
OA	outer approximation
P	a general biobjective mixed-integer linear programming problem
$\mathrm{P}^{\mathrm{CAN}}$	The problem solved in each iteration of CAN
$\mathrm{P}^{\mathrm{GDP}}$	generalized disjunctive programming model for the synthesis of nonlinear process networks
P^{ϵ}	ϵ-constraint model
$\mathrm{P}^{\mathrm{AUG}}$	augmented ϵ-constraint model
$\mathrm{P}^{\mathrm{SAUG}}$	simple augmented ϵ-constraint model

$\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX1}}$,$\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX2}}$	lexicographic problems to find nondominated convex combinations
$\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX1}}$,$\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX2}}$	lexicographic problems to find the next point in EnpoBomip
$\mathrm{P}^{\mathrm{NW}}_{\mathrm{LX1}}$,$\mathrm{P}^{\mathrm{NW}}_{\mathrm{LX2}}$	lexicographic problems to determine North-West point
$\mathrm{P}^{\mathrm{SE}}_{\mathrm{LX1}}$,$\mathrm{P}^{\mathrm{SE}}_{\mathrm{LX2}}$	lexicographic problems to determine South-East point
Q	number of intervals in ϵ-constraint method
r	range of the second objective function
SP_j	jth subproblem of augmented ϵ-constraint method
s	slack variable in augmented ϵ-constraint method
S	feasible region of P in the space of decision variables
u	binary variables used in no-good constraints
w_i	binary variables to show with alternatives should be considered at the beginning of ϵ-OA
x	continuous variables
X	projection of S into the space of continuous variables
y	integer variables
Y	projection of S into the space of integer variables
Y^{EXC}	set of permanently excluded y's
$\tilde{Y}^{\mathrm{EXC}}$	set of temporarily excluded y's
$\mathcal{Y}$	Boolean variables in network synthesis model
z^{aug}	optimal value of SP_j
z_j	objective j
z^I	ideal point in objective space
z_2^L, z_2^U	lower and upper bounds on the second objective
z_{LB}, z_{UB}	lower and upper bounds in OA

z^N	nadir point in objective space
z^{NW}	North-West point in objective space
z^{SE}	South-East point in objective space
β	optimal value of $\mathrm{P}^{\textsc{Can}}$
δ	acceptable optimality gap in ϵ-OA
ϵ	upper bound of the second objective in each iteration of ϵ-constraint algorithm
Θ	feasible region of P in objective space
Θ^*	nondominated set of problem P
Θ_E	extreme supported nondominated points of problem P
ϑ_i	maximum values of integer variables
ϑ	maximum of ϑ_i's
μ	scalar in objective function of $\mathrm{P}^{\textsc{aug}}$
ξ	binary scalar to show if the current point in EnpoBomip is efficient
ϕ	optimal value of $\mathrm{LP}^{\textsc{nep}}$
Φ	the set of feasible solutions of problem P
$\tilde{\Phi}$	potential efficient solutions with alternative y-values in EnpoBomip
Φ^*	the set of efficient solutions of problem P with all alternative values of integer variables
φ	efficiency of algorithms
χ	feasible solution of problem P
$\chi^{\textsc{cp}}$	current point in EnpoBomip
$\chi^{\textsc{nep}}$	next-lower extreme point in EnpoBomip
$\chi^{\textsc{np}}$	next point in EnpoBomip

Chapter 1

INTRODUCTION

Most of the problems in science, business, and engineering involve multiple criteria and can be formulated as mathematical programs, referred as Multiple Objective Optimization Problems (MOOPs). As MOOPs do not generally possess a single solution to optimize all objective functions simultaneously, a set of efficient solutions found, solutions with no possibility of improvement in one objective value without worsening others. Projection of the set of efficient solutions into space of objective functions is called nondominated set or frontier. Several approaches have been proposed in literature to find the exact set of efficient solutions for some classes of MOOPs or to enumerate a subset of efficient solutions to provide an approximation of the efficient set.

MOOPs are categorized based on the number of objective functions, linearity of objective functions and constraints, and nature of decision variables. In my thesis, we consider biobjective mixed-integer linear/nonlinear programming problems where there are two objective functions to be simultaneously optimized, objective functions and constraints are either linear or nonlinear convex and include integer or real-valued decision variables.

The nondominated frontiers of biobjective mixed-integer linear programming problems (BOMILPs) usually have very complex form and consist of many isolated points, line segments and distinct regions. Some researchers considered restrictive assumptions to prevent the complexity of the nondominated frontier. Some papers, on the other hand, considered general BOMILPs but proposed algorithms to find only a subset of nondominated set. To the best of our knowledge, there is no algorithm in literature to find the exact nondominated set of general BOMILPs. We examine general BOMILPs and propose an effective algorithm to find the exact nondominated frontier of these problems. Moreover, existing algorithms, e.g. augmented ϵ-constraint [Mavrotas, 2009, Mavrotas and Florios, 2013], report only one possible integer variable value realization for each nondominated point; however, our algorithm finds all alternative integer variable values associated with each nondominated point. As the values of integer variables dictate system specifications in most of BOMILPs, it may be critically important for decision makers to be aware of all possible systems associated with each nondominated point. We introduce an illustrative example of objective space of a BOMILP to show potential complexities in nondominated frontiers of BOMILPs and highlight the possible difficulties in finding the exact nondominated frontier with all possible values of integer variables associated with each nondominated point. The ϵ-constraint and CAN are examined and their weaknesses are illustrated both theoretically and practically on some examples. The efficiencies of augmented ϵ-constraint and CAN are also compared. We present fundamental theories that are necessary to highlight important characteristics of our novel algorithm, ENPOBOMIP. The ENPOBOMIP is presented and its accuracy and performance are proven.

We also investigate biobjective mixed-integer nonlinear programming problems (BOMINLPs) for the synthesis of process networks that are formulated using generalized disjunctive programming (GDP) and develop an effective algorithm, ϵ–OA, to solve bi-objective GDPs in the synthesis of nonlinear process networks. The GDP uses Boolean variables to formulate nonlinear disjunctions for process units in the networks. If a process unit is present in the network, the Boolean variable becomes "True" and a nonlinear convex inequality between flows holds, otherwise, the unit is absent, the Boolean variable is "False" and the associated flows are zero. The GDP also incorporates propositional logics that are redundant and do not eliminate optimal solutions; however, they are very useful in reducing the search space so that the solution time decreases [Türkay and Grossmann, 1996b]. The (ϵ–OA) uses augmented ϵ-constraint and logic-based outer approximation, to find nondominated points in a very short CPU time. Logic-based OA involves repeatedly solving the NLP subproblem, generated by fixing the values of discrete variables, and the Master problem, constructed by replacing the nonlinear functions by their linear approximations, until the bounds from these two problems converge. The ϵ–OA uses the results of the previously solved subproblems to eliminate the unnecessary operations in subsequent iterations.

A comprehensive experimental work is conducted to perform a through comparative analysis of augmented ϵ-constraint, CAN, and ENPOBOMIP on three well-known problem sets: biobjective synthesis of complex production networks, biobjective facility location problems, and biobjective fixed charge network design problems. We analyze the solution quality and computational time of augmented ϵ-constraint, CAN, and ENPOBOMIP, and show that the ENPOBOMIP is a very effective algorithm to find the exact nondominated frontiers of BOMILPs with all alternative values of integer variables, and it significantly outperforms augmented ϵ-constraint and CAN. We also present a sensitivity analysis to show the effects of magnitude of objective functions coefficients on the shape and complexity of nondominated frontiers, and the solution quality and computational performance of the three algorithms.

We also conduct an experiment to compare the effectiveness of ϵ–OA with the straightforward use of OA to solve subproblems of the augmented ϵ-constraint method (ϵ-con+OA), the augmented ϵ-constraint with MINLP solvers (ϵ-MINLP), and traditional ϵ-constraints (T-ϵ-con). We illustrate these four algorithms on 3 and 8 process networks and show that ϵ–OA is more effective than others. Moreover, compared to T-ϵ-con, our novel algorithm ϵ–OA enhances the quality of the results as it guarantees the efficiency of the solutions.

Chapter 2

LITERATURE SURVEY

As the decision making in optimization of real world problems usually involves multiple criteria, e.g. cost, social issues, environmental impacts, and profit, MOOPs have gained increasing interest in various applications in recent decades. A comprehensive literature review on MOOPs is provided by [Van Veldhuizen and Lamont, 2000, Marler and Arora, 2004, Figueira et al., 2005]. A branch and bound algorithm was presented by [Stidsen et al., 2014] for a limited class of BOMILPs where the integer variables are binary and continuous variables are only allowed to be part of one of the two objective functions. The ϵ-constraint [Haimes et al., 1971, Hwang et al., 1979], augmented ϵ-constraint [Mavrotas, 2009, Mavrotas and Florios, 2013], and simple augmented ϵ-constraint [Zhang and Reimann, 2014] have been proposed to find a finite number of nondominated points to provide an approximation of nondominated frontiers of BOMILPs. The Can algorithm [Cohon, 2013, Aneja and Nair, 1979] is used by many researchers to find a subset of nondominated set of BOMILPs, called extreme supported nondominated points.

The Generalized Disjunctive Programming (GDP) framework for the representation of superstructures and its demonstration in linear systems was presented by [Raman and Grossmann, 1994]. The discrete-continuous nature of the system was modeled using disjunctions and the logical interrelationships among the discrete decisions were represented with propositional logic expressions. The extension of GDP to nonlinear systems was investigated by [Türkay and Grossmann, 1996b]. They presented efficient logic-based outer approximation (OA) and Benders decomposition algorithms to solve this challenging problem. The efficiency of the proposed algorithms was illustrated on nonlinear process networks with complex investment costs [Türkay and Grossmann, 1996a, Turkay and Grossmann, 1998]. The algebraic representation of nonlinear disjunctions and the propositional logics are investigated by [Raman and Grossmann, 1994, Türkay and Grossmann, 1996b, Turkay and Grossmann, 1998, Grossmann and Trespalacios, 2013, Sawaya and Grossmann, 2007, Lee and Grossmann, 2005, Grossmann and Lee, 2003].

Several researchers used multi-objective mixed-integer nonlinear programming to solve MOOP problems in different applications. [Martinez and Eliceche, 2008, Martínez and Eliceche, 2011] studied minimization of environmental impacts and cost in operation of steam and power plant and formulated the problem as MOMINLP. They solved the resulting MOOP using minimum distance to the utopia point, ϵ-constraint, weighted sum and global criterion. [Čuček et al., 2012b] developed a MI(N)LP model for regional biomass energy supply chains and used ϵ-constraint method to find the efficient solutions to this problem. [Čuček et al., 2012a] reduced the dimensionality of the criteria in MOMINLP of biomass energy supply chains based on similar behavior among criteria. [Chakraborty and Linninger, 2002] presented combinatorial process synthesis for developing plant-wide recovery and treatment policies with conflicting objectives cost and environmental impact. A common issue in all of the previous work is the straightforward use of existing methods, specially the traditional ϵ-constraint (T-ϵ-con), to solve the MOOP problems. In T-ϵ-con, one of the objectives optimized while the values of other criteria are bounded by a parameter, ϵ. Then, a set of efficient solutions are found by systematic variation of ϵ [Haimes et al., 1971, Hwang et al., 1979]. Several methods for selecting ϵ-values are studied [Goicoechea et al., 1976, Stadler, 1988]. The drawback of T-ϵ-con is, however, the possibility of generating dominated solutions. The augmented ϵ-constraint method overcomes this impediment by including the slacks of the constraints that are added due to the other criteria to the objective function, weighted by a parameter μ, to eliminate the possibility of generating weekly dominated solutions [Mavrotas, 2009, Mavrotas and Florios, 2013].

MINLPs are categorized as difficult problems as they include both discrete and continuous variables and involve nonlinearity in Objective function and constraints. Several methods have been proposed to solve MINLPs including NLP-based branch and bound [Nabar and Schrage, 1991], generalized Beneders decomposition [Geoffrion, 1972], and OA [Duran and Grossmann, 1986]. The OA is sutiable for convex MINLPs and its execution involves repeatedly solving the NLP subproblem, generated by fixing the values of discrete variables, and the Master problem, constructed by replacing the nonlinear functions by their linear approximations, until the bounds from these two problems converge.

Chapter 3

PRELIMINARIES AND TERMINOLOGY

In this chapter, we present the preliminaries and terminology for biobjective mixed-integer linear programming problems and biobjective nonlinear process networks that are used throughout the thesis. Most of the definitions for BOMILPs are extendable to BOMINLPs. Illustrative examples are used to clarify the definitions.

3.1 Preliminaries and Terminology for BOMILPs

A bi-objective mixed-integer linear programming problem is represented as follows.

$$\text{P: } \min z = (z_1, z_2) = (c_1^\top x + f_1^\top y, c_2^\top x + f_2^\top y) \text{ s.t. } (x, y) \in S \tag{3.1}$$

where $S = \{(x, y) : Ax + By = b, x \in \mathbb{R}^n, y \in \mathbb{Z}^p\}$, and x and y are n and p-dimensional continuous and discrete variables, respectively. The objective functions z_1 and z_2 are linear in x and y and should be simultaneously minimized, c_1 and c_2 are n-vectors and f_1 and f_2 are real p-vectors, and A and B are real $m \times n$ and $m \times p$ matrices, respectively. Problem P can be converted into a bi-objective linear programming problem (BOLP) by fixing the values of discrete variables, i.e. $y = \bar{y}$. The objective function of the problem P becomes $z = (c_1^\top x + \bar{F}_1, c_2^\top x + \bar{F}_2)$, where $\bar{F}_j = f_j^\top \bar{y}$, $j = 1, 2$, and the feasible region becomes $Ax = \bar{b}$, $x \in \mathbb{R}^n$ where $\bar{b} = b - B\bar{y}$.

$$\text{P}(\bar{y})\text{: } \min z = (z_1, z_2) = (c_1^\top x + \bar{F}_1, c_2^\top x + \bar{F}_2) \text{ s.t. } x \in S(\bar{y}) \tag{3.2}$$

where $S(\bar{y}) = \{x \in \mathbb{R}^n : Ax = \bar{b}\}$ is the feasible region of $\mathrm{P}(\bar{y})$. Note that $\bar{F}_1, \bar{F}_2$, and $\bar{b}$ are constant and $S(\bar{y})$ is a convex set. Let X and Y denote the projection of S into $\mathbb{R}^n$ and $\mathbb{Z}^p$ spaces, i.e. $X = \{x : (x, y) \in S\}$ and $Y = \{y : (x, y) \in S\}$. Note that Y is the set of possible fixes of discrete variables and X can be calculated as $X = \bigcup_{y \in Y} S(y)$. Each point in S has an image in the objective space. The set of all attainable objective vectors Θ can be defined as,

$$\Theta = \{z \in \mathbb{R}^2 : z_1 = c_1^\top x + f_1^\top y, z_2 = c_2^\top x + f_2^\top y, (x, y) \in S\} \tag{3.3}$$

The image of a particular feasible point $(\bar{x}, \bar{y}) \in S$ in the objective space, i.e. $\Theta(\bar{x}, \bar{y})$, and any $\bar{S} \subset S$, i.e. $\Theta(\bar{S})$, can be similarly defined. We assume $S \neq \emptyset$ and $\min z_j \neq -\infty, j = 1, 2$; hence, $\Theta \neq \emptyset$. For a given $\bar{y} \in Y$, $\Theta(\bar{y})$ is the image of $S(\bar{y})$ in objective space, i.e. $\Theta(\bar{y}) = \{\Theta(x, \bar{y}) : x \in S(\bar{y})\}$. Note that $\Theta(\bar{y})$, $\forall \bar{y} \in Y$, is closed and convex. Let $\partial\Theta(\bar{y})$ denote the boundary of $\Theta(\bar{y})$.

We define a feasible solution $\chi = (x, y, z)$ to problem P such that $z = \Theta(x, y)$ and $(x, y) \in S$. Let Φ denote the set of all feasible solutions, i.e. $\Phi = \{\chi = (x, y, z) : z = \Theta(x, y), (x, y) \in S\}$. For objective vectors z^1 and z^2:

- If $z^1 \leqq z^2$, i.e. $z^1_j \leq z^2_j, j = 1, 2$, then z^1 weakly dominates z^2.

- If $z^1 \leq z^2$, i.e. $z^1_j \leq z^2_j, j = 1, 2$ and $z^1 \neq z^2$, then z^1 dominates z^2.

- If $z^1 < z^2$, i.e. $z^1_j < z^2_j, j = 1, 2$, then z^1 strictly dominates z^2.

We define the set of nondominated points Θ^* as the set of attainable objective vectors z^* that are not dominated by points in Θ; hence, $\Theta^* = \{z^* \in \Theta : \nexists \hat{z} \in \Theta, \hat{z} \leq z^*\}$. For a nondominated point z^*, the associated (x^*, y^*) such that $z^* = \Theta(x^*, y^*)$ is called efficient point. For BOMILPs, we aim to find the exact nondominated points of problem P. For each $z^* \in \Theta^*$, we provide all associated y values. Since x values can be determined using an LP formulation after knowing z and y values, we do not provide all associated x values for each specific z and y values. Also note that the associated x values may be convex sets that include infinitely many number of alternative points; hence it is not always possible to report all of these points. We denote the set of efficient solutions that we aim to find by $\Phi^* = \{\chi^* = (x^*, y^*, z^*)\}$, where $\Phi^* \subset \Phi$, $z^* \in \Theta^*$, and (x^*, y^*, z^*)'s are distinct with respect to (y^*, z^*), i.e. if (x^{1*}, y^{1*}, z^{1*}), $(x^{2*}, y^{2*}, z^{2*}) \in \Phi^*$, then $(y^{1*}, z^{1*}) \neq (y^{2*}, z^{2*})$.

We define $\mathbb{R}^2_> = \{z \in \mathbb{R}^2 : z > 0\}$, $\mathbb{R}^2_\geq = \{z \in \mathbb{R}^2 : z \geq 0\}$, and $\mathbb{R}^2_\geqq = \{z \in \mathbb{R}^2 : z \geqq 0\}$. For a point $\bar{z} \in \mathbb{R}^2$ and a set $\bar{\Theta} \subset \mathbb{R}^2$, we define $\bar{z}^> = \bar{z} + \mathbb{R}^2_>$ and $\bar{\Theta}^> = \bar{\Theta} + \mathbb{R}^2_>$ where $+$ denotes algebraic sum. The sets $\bar{z}^\geq$, $\bar{z}^\geqq$, $\bar{\Theta}^\geq$ and $\bar{\Theta}^\geqq$ are defined similarly. We also define these sets for $<$, $\leq$ and $\leqq$. Let $\bar{z}_j^<$ and $\bar{z}_j^\leq$ respectively denote $\{z \in \mathbb{R}^2 : z_j < \bar{z}_j\}$ and $\{z \in \mathbb{R}^2 : z_j \leq \bar{z}_j\}$, $j = 1, 2$. We similarly define $\bar{z}_j^>$ and $\bar{z}_j^\geq$, $j = 1, 2$. Note that Θ^* can be also defined as $\{z^* \in \Theta : z^{*\leq} \cap \Theta = \emptyset\}$. In this paper, we mostly use this definition to prove that a point or a set is nondominated, i.e. point $\bar{z} \in \Theta$ is nondominated if $\bar{z}^\leq \cap \Theta = \emptyset$, and set $\bar{\Theta} \subset \Theta$ is nondominated if $\bar{\Theta}^\leq \cap \Theta = \emptyset$.

Let $[z^1, z^2]$ denote convex combination of z^1 and z^2, i.e. $[z^1, z^2] = \lambda z^1 + (1-\lambda)z^2$, $0 \leq \lambda \leq 1$. The set $[z^1, z^2]$ is the line segment in $\mathbb{R}^2$ from z^1 to z^2. We define $(z^1, z^2) = \lambda z^1 + (1-\lambda)z^2$, $0 < \lambda < 1$. The sets $(z^1, z^2]$ and $[z^1, z^2)$ are similarly defined. The convex combination of $\chi^1 = (x^1, \bar{y}, z^1)$ and $\chi^2 = (x^2, \bar{y}, z^2)$ is defined as $[\chi^1, \chi^2] = (\lambda x^1 + (1-\lambda)x^2, \bar{y}, \lambda z^1 + (1-\lambda)z^2)$, $0 \leq \lambda \leq 1$. Note that if $\chi^1 = (x^1, \bar{y}, z^1)$ and $\chi^2 = (x^2, \bar{y}, z^2)$ are feasible solutions of problem P, then $[\chi^1, \chi^2]$ is also feasible. The sets (χ^1, χ^2), $(\chi^1, \chi^2]$, and $[\chi^1, \chi^2)$ are similarly defined.

The ideal point $z^I = (z_1^I, z_2^I)$ in objective space is obtained by $z_j^I = \min z_j$ s.t. $(x, y) \in S$, $j = 1, 2$. The nadir point $z^N = (z_1^N, z_2^N)$ is found by $z_j^N = \max z_j$ s.t. $z \in \Theta^*$, $j = 1, 2$. Note that $z^I_\geqq \cap z^N_\leqq$ specifies a rectangle with other extreme points $z^{\text{NW}} = (z_1^I, z_2^N)$ and $z^{\text{SE}} = (z_1^N, z_2^I)$, North-West and South-East points, that is a bound for nondominated points of problem P.

The extreme nondominated points in objective space are the extreme points in nondominated frontier, and the remaining points in Θ^* are nonextreme nondominated points. Supported nondominated points are those that can be found when optimizing a weighted sum of objectives for some weights, otherwise, the nondominated point is called unsupported. The extreme supported nondominated (ESN) points are both extreme and supported. The set of ESN points is denoted as Θ_E.

3.1.1 An illustrative example of BOMILPs

Fig. 3.1 presents a hypothetical example of objective space of a BOMILP that is used throughout this paper for the purpose of illustration. There are 12 possible y values, i.e. $|Y| = 12$. The convex sets in objective space $\Theta(y^j)$, $j = 1, \cdots, 12$ are shown that are images of $S(y^j)$'s, e.g. $\Theta(y^1)$is the image of $S(y^1)$ in objective space. These sets can be singleton, e.g. $\Theta(y^1)$ and $\Theta(y^7)$, a line segment, e.g. $\Theta(y^{10})$, or polygons. The set Θ is the union of $\Theta(y^j)$'s and is not necessarily convex. Each point in the objective space is the image of one or more $(x, y) \in S$. For example, z^1 is the image of $(\bar{x}, y^1)$ for some values of $\bar{x} \in S(y^1)$. Note that $S(y^1)$ is not necessarily singleton. For a $\bar{x} \in S(y^1)$, the vector $\bar{\chi} = (\bar{x}, y^1, z^1)$ is a feasible solution of problem P, i.e. $\bar{\chi} \in \Phi$. Some points in objective space are associated with more than one y value in the space of decision variables , e.g. z^{10} is associated with y^4 and y^5, and z^{14} is associated with y^9, y^{10}, and y^{12}.

The set $z^{9\leqq}$, for instance, is defined by equations $z_1 \leq 10$ and $z_2 \leq 8$. The set $z^{9\leq}$ is all points in $z^{9\leqq}$ except for z^9, i.e. $z^{9\leq} = z^{9\leqq} \backslash \{z^9\}$. The set $z^{9<}$, is defined by equations $z_1 < 10$ and $z_2 < 8$. For the line segment $[z^{14}, z^{18}]$, for instance, $[z^{14}, z^{18}]^{\leqq}$ is defined by $z_1 \leq 5$, $z_2 \leq 18$, and $z_1 + z_2 \leq 19$, and the set $[z^{14}, z^{18}]^{\leq}$ is defined by $z_1 \leq 5$, $z_2 \leq 18$, and $z_1 + z_2 < 19$. The set $[z^{14}, z^{18}]^{<}$ is defined by $z_1 < 5$, $z_2 < 18$, and $z_1 + z_2 < 19$. For the line segment $(z^{10}, z^{11}]$, we have $(z^{10}, z^{11}]^{\leqq} =$

$\{z_1 < 10, z_2 \leq 11, z_1 + z_2 \leq 19\}$, $(z^{10}, z^{11}]^{\leq} = \{z_1 < 10, z_2 \leq 11, z_1 + z_2 < 19\}$, and $(z^{10}, z^{11}]^{<} = \{z_1 < 10, z_2 < 11, z_1 + z_2 < 19\}$.

The set of nondominated points is shown in black filled circles and line segments; $\Theta^* = \{z^1, (z^5, z^6], [z^7, z^8), z^9, (z^{10}, z^{11}], z^{12}, [z^{14}, z^{18}]\}$. All nondominated points are associated with one y value except for z^{14} (with y^9, y^{10}, and y^{12}) and $[z^{15}, z^{16}]$ (with y^{11} and y^{12}). Note that for all $z^* \in \Theta^*$, the relationship $z^{*\leq} \cap \Theta = \emptyset$ holds. Examples of domination are shown in Fig. 3.1. For instance, z^1 dominates z^2, z^3, and z^5, but it does not dominate z^8 or z^{10}. The points z^8 and z^{10} are dominated by z^9. Point z^{18} strictly dominates z^{19}. Since z^{14} is a member of $\Theta(y^9)$, $\Theta(y^{10})$, and $\Theta(y^{12})$, then $z^{14} \in \Theta(y^9)$ weakly dominates $z^{14} \in \Theta(y^{10})$ and $z^{14} \in \Theta(y^{12})$, and vice versa.

The ideal, nadir, North-West and South-East points are shown in Fig. 3.1. Note that z^I and z^N are not feasible.

The set of extreme nondominated points is $\{z^1, z^6, z^7, z^9, z^{11}, z^{12}, z^{14}, z^{18}\}$. The remaining points in Θ^* are nonextreme nondominated points, e.g. (z^5, z^6) and (z^7, z^8). There are only two ESN points and $\Theta_E = \{z^1, z^{18}\}$; hence, $\{z^6, z^7, z^9, z^{11}, z^{12}, z^{14}\}$ is the set of extreme unsupported nondominated points.

3.2 Preliminaries and Terminology for biobjective nonlinear process networks

The generalized disjunctive programming model of bi-objective nonlinear process networks (P^{GDP}) is represented as follows.

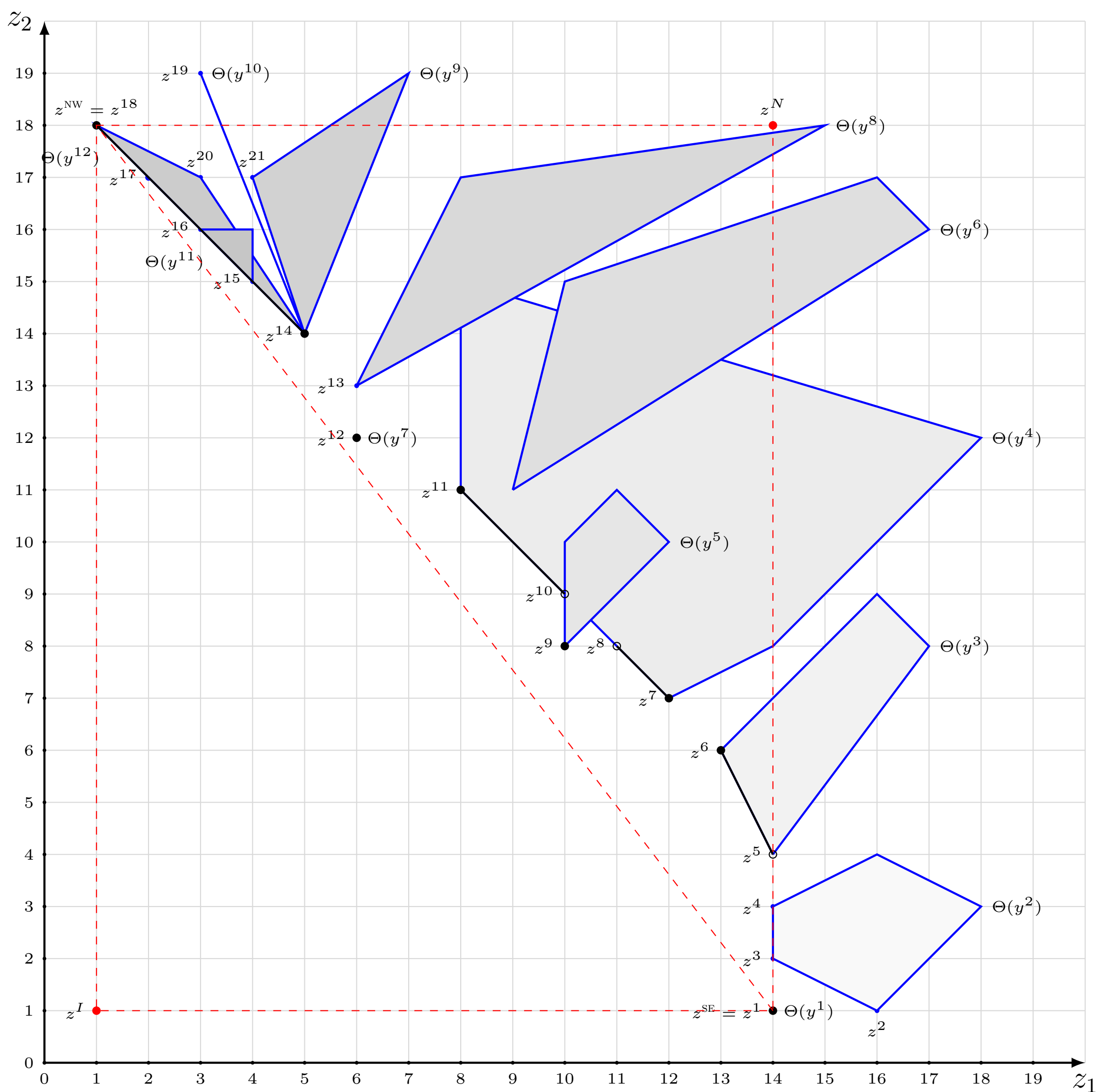


Figure 3.1: An illustrative example of objective space of a BOMILP

$$
\begin{aligned}
\min z \quad & = (z_1, z_2) \\
& = \left(\sum_i c_i + f_1(x), \sum_i \rho_i + f_2(x) \right) && (3.4) \\
\text{s.t.} \quad & g(x) \leq 0, && (3.5) \\
& \begin{bmatrix} \mathcal{Y}_i \\ h_i(x) \leq 0 \\ c_i = \gamma_i \\ \rho_i = \eta_i \end{bmatrix} \vee \begin{bmatrix} \neg \mathcal{Y}_i \\ B^i x = 0 \\ c_i = 0 \\ \rho_i = 0 \end{bmatrix} \qquad \forall i \in D, && (3.6) \\
& \Omega(\mathcal{Y}) = True, && (3.7) \\
& x \in \mathbb{R}^n, c \geq 0, \rho \geq 0, \mathcal{Y} \in \{True, False\}^m. && (3.8)
\end{aligned}
$$

The bi-objective optimization model involves continuous (x, c, and ρ) and Boolean ($\mathcal{Y}$) decision variables. The Boolean variables are defined for process units in the network and show the existence of the corresponding unit. The objective function simultaneously minimizes two functions, z_1 and z_2. Eq. (3.5) represent global inequalities that always hold independent of the values of Boolean variables. Disjunctions (Eq. 3.6) are written for all processing units; if the corresponding Boolean variable is "true", then a particular relation among x's hold and the relevant fixed costs (c and ρ) are paid, otherwise the associated c, ρ, and x's become zero. Here, $B^i = [b_k^\top]$, where $b_k^\top = e_k^\top$ if $x_k = 0$, and $b_k^\top = 0^\top$ if $x_k \neq 0$. Eq. (3.7) is the logic relations between Boolean variables based on the connections and interactions among units. In this model, $f_1(x)$, $f_2(x)$, $g(x)$, and $h_i(x)$ can be linear or nonlinear convex functions.

3.2.1 Example: 3-process nonlinear network

In this section, we present the well-known 3-process network synthesis example (Kocis and Grossmann, 1989) to illustrate the bi-objective model given by Eqs. (3.4)-(3.8).

This example consists of 3 processing units, as shown in Fig 3.2. The raw material A should be processed in either units 2 or 3 to produce B. The material B can be also purchased from outside. Unit 1 processes B and produces C. The variables $x_k, k = 1, \cdots, 8$, are the amount of material flows.

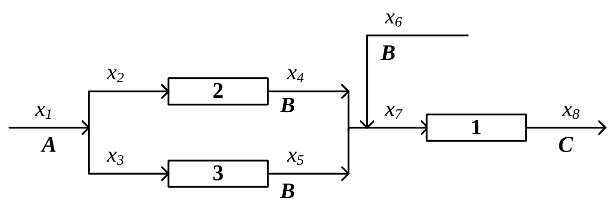


Figure 3.2: Superstructure for the 3-process network synthesis

The objective function of this problem is to simultaneously minimize z_1 and z_2. Note that we generate the second objective function, z_2, so that the solution corresponding to the best compromise between these objectives is found.

$$
\begin{aligned}
\min \quad & z = (z_1, z_2) \\
& z_1 = c_1 + c_2 + c_3 + x_4 + 1.8x_1 + 1.2x_5 + 7x_6 - 11x_8 \\
& z_2 = \rho_1 + \rho_2 + \rho_3 \\
& \qquad +1.8x_1 + 2x_2 + 3x_3 + x_4 + 1.2x_5 + 7x_6 - 5x_7 - 11x_8
\end{aligned} \tag{3.9}
$$

The material balance equations at mixing/splitting points, and specifications on the flows are as follows. These relations are general and hold independent of the values of Boolean variables.

$$
\begin{aligned}
&x_1 - x_2 - x_3 = 0, \\
&x_7 - x_4 - x_5 - x_6 = 0, \\
&x_5 \leq 5, \\
&x_8 \leq 1
\end{aligned} \tag{3.10}
$$

The disjunction for the processing units:

$$
\begin{gathered}
\begin{bmatrix} \mathcal{Y}_1 \\ x_8 = 0.9x_7 \\ c_1 = 3.5 \\ \rho_1 = 3.5 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_1 \\ x_7 = x_8 = 0 \\ c_1 = 0 \\ \rho_1 = 0 \end{bmatrix}, \\
\begin{bmatrix} \mathcal{Y}_2 \\ x_4 = ln(1 + x_2) \\ c_2 = 1 \\ \rho_2 = 1 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_2 \\ x_2 = x_4 = 0 \\ c_2 = 0 \\ \rho_2 = 0 \end{bmatrix}, \\
\begin{bmatrix} \mathcal{Y}_3 \\ x_5 = 1.2ln(1 + x_3) \\ c_3 = 1.5 \\ \rho_3 = 12 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_3 \\ x_3 = x_5 = 0 \\ c_3 = 0 \\ \rho_3 = 0 \end{bmatrix}
\end{gathered} \tag{3.11}
$$

The design specifications and logic propositions based on the connections and interactions between units:

$$
\begin{aligned}
&\mathcal{Y}_1 \Rightarrow \mathcal{Y}_2 \vee \mathcal{Y}_3 \vee (\neg\mathcal{Y}_2 \wedge \neg\mathcal{Y}_3),\\
&\mathcal{Y}_2 \Rightarrow \mathcal{Y}_1,\\
&\mathcal{Y}_3 \Rightarrow \mathcal{Y}_1,\\
&\neg\mathcal{Y}_2 \vee \neg\mathcal{Y}_3
\end{aligned} \tag{3.12}
$$

Finally, the definition of variables as positive continuous and Boolean:

$$
\begin{aligned}
&x_k \geq 0 \qquad \forall k = 1, \cdots, 8,\\
&c_i, \rho_i \geq 0 \qquad \forall i = 1, 2, 3,\\
&\mathcal{Y}_i \in \{True, False\} \qquad \forall i = 1, 2, 3
\end{aligned} \tag{3.13}
$$

3.2.2 Example: 8-process nonlinear network

The superstructure for the 8-process network synthesis (Fig. 3.3, Kocis and Grossmann, 1989) consists of 8 processing units and 25 continuous variables for the amount of material flows.

The generalized disjunctive programming model for 8-process nonlinear network is given as follows.

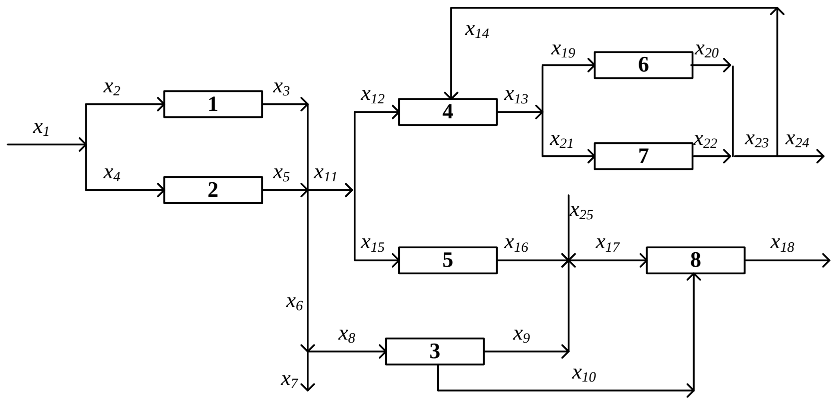


Figure 3.3: Superstructure for the 8-process network synthesis

$$
\begin{aligned}
\min \quad & z = (z_1, z_2), \\
& z_1 = c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8 \\
& \qquad + x_2 - 10x_3 + x_4 - 15x_5 - 40x_9 + 15x_{10} + 15x_{14} + 80x_{17} \\
& \qquad - 65x_{18} + 25x_{19} - 60x_{20} + 35x_{21} - 80x_{22} - 35x_{25} + 122, \\
& z_2 = \rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5 + \rho_6 + \rho_7 + \rho_8 \\
& \qquad + x_1 - 2x_6 + 3x_7 + 2x_8 - 2x_{11} + 4x_{12} \\
& \qquad + 3x_{13} + 2x_{15} - x_{16} + 5x_{23} + x_{24} + 110
\end{aligned} \tag{3.14}
$$

Material balance equations:

$$
\begin{aligned}
x_{13} &= x_{19} + x_{21}, \\
x_{17} &= x_9 + x_{16} + x_{25}, \\
x_{11} &= x_{12} + x_{15},
\end{aligned}
$$

$$
\begin{aligned}
&x_3 + x_5 = x_6 + x_{11}, \\
&x_6 = x_7 + x_8, \\
&x_{23} = x_{20} + x_{22}, \\
&x_{23} = x_{14} + x_{24}, \\
&x_1 = x_2 + x_4
\end{aligned}
\tag{3.15}
$$

Specifications and upper bounds on the amount of flows:

$$
\begin{aligned}
&x_{10} \leq 0.8x_{17}, \\
&x_{10} \geq 0.4x_{17}, \\
&x_{12} \leq 5x_{14}, \\
&x_{12} \geq 2x_{17}, \\
&x_3 \leq 2, \\
&x_5 \leq 2, \\
&x_9 \leq 2, \\
&x_{10} \leq 1, \\
&x_{14} \leq 1, \\
&x_{17} \leq 2, \\
&x_{19} \leq 2, \\
&x_{21} \leq 2, \\
&x_{25} \leq 3
\end{aligned}
\tag{3.16}
$$

Disjunctions for units:

$$\begin{bmatrix} \mathcal{Y}_1 \\ exp(x_3) - 1 \leq x_2 \\ c_1 = 5 \\ \rho_1 = 95 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_1 \\ x_2 = x_3 = 0 \\ c_1 = 0 \\ \rho_1 = 0 \end{bmatrix},$$

$$\begin{bmatrix} \mathcal{Y}_2 \\ exp(x_5/1.2) - 1 \leq x_4 \\ c_2 = 8 \\ \rho_2 = 70 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_2 \\ x_4 = x_5 = 0 \\ c_2 = 0 \\ \rho_2 = 0 \end{bmatrix},$$

$$\begin{bmatrix} \mathcal{Y}_3 \\ 1.5x_9 + x_{10} = x_8 \\ c_3 = 6 \\ \rho_3 = 10 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_3 \\ x_9 = 0 \\ c_3 = 0 \\ \rho_3 = 0 \end{bmatrix},$$

$$\begin{bmatrix} \mathcal{Y}_4 \\ 1.25(x_{12} + x_{14}) = x_{13} \\ c_4 = 10 \\ \rho_4 = 45 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_4 \\ x_{12} = x_{13} = x_{14} = 0 \\ c_4 = 0 \\ \rho_4 = 0 \end{bmatrix},$$

$$\begin{bmatrix} \mathcal{Y}_5 \\ x_{15} = 2x_{16} \\ c_5 = 6 \\ \rho_5 = 84 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_5 \\ x_{15} = x_{16} = 0 \\ c_5 = 0 \\ \rho_5 = 0 \end{bmatrix},$$

$$\begin{bmatrix} \mathcal{Y}_6 \\ exp(x_{20}/1.5) - 1 \leq x_{19} \\ c_6 = 7 \\ \rho_6 = 15 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_6 \\ x_{19} = x_{20} = 0 \\ c_6 = 0 \\ \rho_6 = 0 \end{bmatrix},$$
$$\begin{bmatrix} \mathcal{Y}_7 \\ exp(x_{22}) - 1 \leq x_{21} \\ c_7 = 4 \\ \rho_7 = 80 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_7 \\ x_{21} = x_{22} = 0 \\ c_7 = 0 \\ \rho_7 = 0 \end{bmatrix},$$
$$\begin{bmatrix} \mathcal{Y}_8 \\ exp(x_{18}) - 1 \leq x_{10} + x_{17} \\ c_8 = 5 \\ \rho_8 = 60 \end{bmatrix} \vee \begin{bmatrix} \neg\mathcal{Y}_8 \\ x_{10} = x_{17} = x_{18} = x_{25} = 0 \\ c_8 = 0 \\ \rho_8 = 0 \end{bmatrix} \tag{3.17}$$

Logic propositions and specifications:

$$\begin{aligned}
&\mathcal{Y}_1 \underline{\vee} \mathcal{Y}_2, \\
&\mathcal{Y}_4 \underline{\vee} \mathcal{Y}_5, \\
&\mathcal{Y}_6 \underline{\vee} \mathcal{Y}_7, \\
&\mathcal{Y}_1 \Rightarrow \mathcal{Y}_3 \vee \mathcal{Y}_4 \vee \mathcal{Y}_5, \\
&\mathcal{Y}_2 \Rightarrow \mathcal{Y}_3 \vee \mathcal{Y}_4 \vee \mathcal{Y}_5, \\
&\mathcal{Y}_3 \Rightarrow \mathcal{Y}_8, \\
&\mathcal{Y}_3 \Rightarrow \mathcal{Y}_1 \vee \mathcal{Y}_2, \\
&\mathcal{Y}_4 \Rightarrow \mathcal{Y}_1 \vee \mathcal{Y}_2,
\end{aligned}$$

$$
\begin{aligned}
&\mathcal{Y}_4 \Rightarrow \mathcal{Y}_6 \vee \mathcal{Y}_7, \\
&\mathcal{Y}_5 \Rightarrow \mathcal{Y}_1 \vee \mathcal{Y}_2, \\
&\mathcal{Y}_5 \Rightarrow \mathcal{Y}_8, \\
&\mathcal{Y}_6 \Rightarrow \mathcal{Y}_4, \\
&\mathcal{Y}_7 \Rightarrow \mathcal{Y}_4, \\
&\mathcal{Y}_8 \Rightarrow \mathcal{Y}_3 \vee \mathcal{Y}_5 \vee (\neg\mathcal{Y}_3 \wedge \mathcal{Y}_5)
\end{aligned}
\tag{3.18}
$$

Variable definitions as continuous and boolean:

$$
\begin{aligned}
&x_k \geq 0 \qquad \forall k = 1, \cdots, 25, \\
&c_i, \rho_i \geq 0 \qquad \forall i = 1, \cdots, 8, \\
&\mathcal{Y}_i \in \{True, False\} \qquad \forall i = 1, \cdots, 8
\end{aligned}
\tag{3.19}
$$

Chapter 4

ENPOBOMIP

This chapter presents our novel algorithm, EnpoBomip, to find the exact nondominated points with all alternative values of integer variables for biobjective mixed-integer linear programming problems. The existing algorithms are discussed first. Then we present the fundamental theories to establish our proposed algorithm.

4.1 Existing methodologies

In this section, we introduce two of the most widely used methodologies, ϵ-constraint and Can, employed by many researchers to solve BOMILPs. The Can is used to find the exact ESN points of BOMILPs.

4.1.1 ϵ-constraint

The ϵ-constraint method [Haimes et al., 1971, Hwang et al., 1979] optimizes one objective while including other objectives as constraints bounded by a value, ϵ. The formulation for BOMILPs can be represented as follows.

$$\mathrm{P}^{\epsilon}: \quad \min z_1 = c_1^\top x + f_1^\top y \tag{4.1}$$

$$\text{s.t. } c_2^\top x + f_2^\top y \leq \epsilon, (x, y) \in S \tag{4.2}$$

[Goicoechea et al., 1976, Stadler, 1988] presented some methods for selecting ϵ-values. Most commonly, the range of the second objective, $(z_2^U - z_2^L)$, is calculated and divided by Q resulting in $Q+1$ equidistant evaluation points, and problem P^{ϵ} is solved setting ϵ to each of these points. In the best case, $Q+1$ solutions are obtained that may be dominated points. For example, if ϵ-constraint is used to solve the problem given in Fig. 3.1, for $\epsilon = 4$, problem P^{ϵ} has multiple optimal solutions and can find any point in $\{z^1, [z^3, z^4], z^5\}$; while the only nondominated point among alternative optimal solutions is z^1.

Several approaches have been proposed to eliminate the possibility of generating dominated solutions. [Mavrotas, 2009] proposed the augmented ϵ-constraint, Augmecon, that includes the slack of the constraint that is added due to the second objective to the objective function, weighted by a parameter μ. [Mavrotas and Florios, 2013] presented modifications and speed-up techniques for Augmecon. The augmented formulation is as follows.

$$\mathrm{P}^{\text{AUG}}: \quad \min c_1^\top x + f_1^\top y - \mu \left(\frac{s}{z_2^U - z_2^L} \right) \tag{4.3}$$

$$\text{s.t. } c_2^\top x + f_2^\top y + s = \epsilon, (x, y) \in S, s \geq 0 \tag{4.4}$$

[Zhang and Reimann, 2014] presented an alternative augmented formulation, called Saugmecon:

$$\mathrm{P}^{\text{SAUG}}: \quad \min c_1^\top x + f_1^\top y + \mu \left(\frac{c_2^\top x + f_2^\top y}{z_2^U - z_2^L} \right) \tag{4.5}$$

$$\text{s.t. } c_2^\top x + f_2^\top y \leq \epsilon, (x, y) \in S \tag{4.6}$$

The Augmecon for solving BOMILPs is represented in algorithm 1. Each objective function is first minimized to calculate the range of the second objective function. In the *while* loop, the augmented problem P^{AUG} is solved for each evaluation point,

$j = 0, \cdots, Q$. The nondominated point (z_1^*, z_2^*) is reported based on the solution obtained. The second *while* loop is to eliminate the unnecessary runs throughout the algorithm [Mavrotas and Florios, 2013]; when a nondominated point (z_1^*, z_2^*) is found, feasibility of z_2^* is checked for the larger values of j, if so, the same nondominated solution is obtained from solving those iterations, hence they can be skipped.

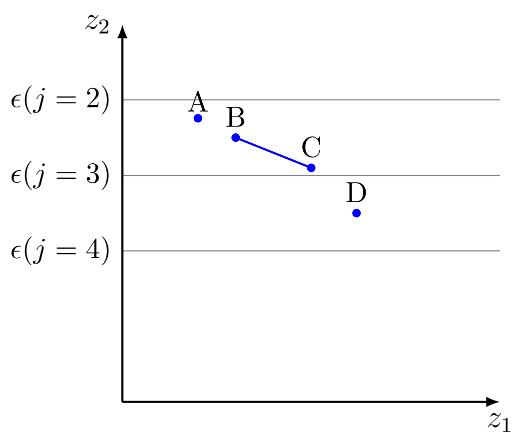
Algorithm 1 AUGMECON to find a subset of nondominated points of BOMILPs

Input: P, Q
Output: Nondominated points
$z_2^L = \min z_2$ s.t. $(x, y) \in S$
$(\bar{z}_1, \bar{z}_2) \Leftarrow \min z_1$ s.t. $(x, y) \in S$
if $z_2^L = \bar{z}_2$ **then**
    $\Theta^* = \{(\bar{z}_1, \bar{z}_2)\}$ is singleton, Stop!
end if
Set $j \leftarrow 0$, $z_2^U \leftarrow \bar{z}_2$, $\epsilon \leftarrow z_2^U$
while $j \leq Q$ **do**
    Solve $\mathrm{P}^{\text{AUG}} \Rightarrow$ nondominated point (z_1^*, z_2^*)
    while $z_2^* \leq \epsilon$ **do**
        Set $j \leftarrow j + 1$, $\epsilon \leftarrow z_2^U - \frac{(z_2^U - z_2^L)}{Q} j$
    end while
end while

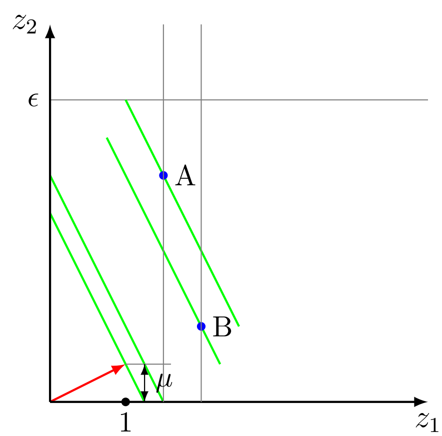
The AUGMECON produces a number of nondominated points that presents an approximation of Θ^*. The number of nondominated points increases when Q increases; however, it requires more computational effort since the first *while* loop may be executed $Q + 1$ times. The number of MILP problems resolution, #MILP, is at most $Q + 3$ and the number of nondominated points found, #NDP, is #MILP $- 2$, given that Θ^* is not singleton. We define the efficiency of the algorithm as $\varphi = \frac{\#\text{NDP}}{\#\text{MILP}}$ which is a representation of the number of nondominated points generated for each MILP resolution. Hence, $\varphi(\text{AUGMECON}) = 1 - \frac{2}{\#\text{MILP}}$. For large Q, $\varphi(\text{AUGMECON}) \approx 1$ meaning that AUGMECON almost generates a nondominated solution for each MILP solved.

Although it is proven that the reported solutions by AUGMECON are nondominated points, there is no guarantee that this method generates the entire nondominated frontier. Two types of errors can be encountered when executing AUGMECON:

1. ϵ-ERROR occurs when progressing from one iteration to the next. As shown in Fig. 4.1(a), suppose that the algorithm is at iteration $j = 2$. Point A is found and then algorithm moves to the next iteration, $j = 3$. The nondominated line segment $[B, C]$ is skipped and point D is found.



(a) ϵ-ERROR



(b) μ-ERROR

Figure 4.1: AUGMECON's errors

2. μ-ERROR due to the augmentation. Suppose that at a particular iteration, the ϵ and μ are as shown in Fig 4.1(b) and points A and B are the only two feasible points with $z_2 \leq \epsilon$. The objective vector $(1, \mu)$ and a number of perpendicular lines are drawn. All points on each line has similar objective value, and the line closer to the origin is better. Hence, point B is found from solving P^{AUG} and nondominated point A is skipped.

The possibility of ϵ-ERROR and μ-ERROR can be reduced by decreasing the values of ϵ and μ, respectively; however, they cannot be eliminated. Decreasing ϵ requires increasing Q, hence more computational effort is needed. Moreover, very small μ

makes the augmented problem difficult to solve, hence each iteration takes more time.

4.1.2 CAN

The CAN, presented by [Cohon, 2013, Aneja and Nair, 1979], is an exact algorithm to find all ESN points of BOMILPs, Θ_E. Algorithm 2 represents the main steps of CAN [Stidsen et al., 2014]. First, z^{NW} and z^{SE} are found using the following lexicographic optimization.

$$\begin{aligned}
&\mathrm{P}^{\text{NW}}_{\text{LX1}} : z^{\text{NW}}_1 = \min z_1 \text{ s.t. } (x, y) \in S \\
&\mathrm{P}^{\text{NW}}_{\text{LX2}} : z^{\text{NW}}_2 = \min z_2 \text{ s.t. } z_1 = z^{\text{NW}}_1, (x, y) \in S &\quad (4.7)\\
&\mathrm{P}^{\text{SE}}_{\text{LX1}} : z^{\text{SE}}_2 = \min z_2 \text{ s.t. } (x, y) \in S \\
&\mathrm{P}^{\text{SE}}_{\text{LX2}} : z^{\text{SE}}_1 = \min z_1 \text{ s.t. } z_2 = z^{\text{SE}}_2, (x, y) \in S &\quad (4.8)
\end{aligned}$$

If $z^{\text{NW}} = z^{\text{SE}}$, then Θ_E and Θ^* are singleton and the algorithm is terminated. In each iteration of the CAN, two ESN points, $\dot{z}$ and $\ddot{z}$, are known and the algorithm searches for another ESN point between them. This can be done using the following optimization problem.

$$\mathrm{P}^{\text{CAN}}(\dot{z}, \ddot{z}) : \beta = \min z_2 - m z_1 \text{ s.t. } (x, y) \in S \quad (4.9)$$

where $m = \frac{\dot{z}_2 - \ddot{z}_2}{\dot{z}_1 - \ddot{z}_1}$. If $\beta = \dot{z}_2 - m\dot{z}_1$, there is no ESN point between $\dot{z}$ and $\ddot{z}$. Otherwise, a new ESN point is found, hence the algorithm continues to search between the new point and $\dot{z}$, and $\ddot{z}$.

If Algorithm 2 is applied to the problem given in Fig. 3.1, it finds z^1 and z^{18}

Algorithm 2 Can to find the exact ESN points of BOMILPs

1: **Input:** P
2: **Output:** sorted Θ_E
3: $\mathrm{LX}(\mathrm{P}^{\mathrm{SE}}_{\mathrm{LX1}},\mathrm{P}^{\mathrm{SE}}_{\mathrm{LX2}})$, $\mathrm{LX}(\mathrm{P}^{\mathrm{NW}}_{\mathrm{LX1}},\mathrm{P}^{\mathrm{NW}}_{\mathrm{LX2}}) \Rightarrow z^{\mathrm{SE}}$, z^{NW}
4: **if** $z^{\mathrm{SE}} = z^{\mathrm{NW}}$ **then**
5:     $\Theta^* = \Theta_E = \{z^{\mathrm{SE}}\}$ is singleton, Stop!
6: **end if**
7: Set $\Theta_E \leftarrow \{z^{\mathrm{NW}}, z^{\mathrm{SE}}\}$, $\dot{z} \leftarrow z^{\mathrm{SE}}$, $\ddot{z} \leftarrow z^{\mathrm{NW}}$
8: **while** $\ddot{z} \neq z^{\mathrm{SE}}$ **do**
9:     Solve $\mathrm{P}^{\mathrm{CAN}}(\dot{z}, \ddot{z}) \Rightarrow \beta, z^*$
10:     **if** $\beta = \dot{z}_2 - m\dot{z}_1$ **then**
11:         Set $\ddot{z} \leftarrow \dot{z}$
12:     **else** Insert z^* between $\ddot{z}$ and $\dot{z}$ in Θ_E
13:     **end if**
14:     **if** $\ddot{z} \neq z^{\mathrm{SE}}$ **then**
15:         $\dot{z} \Leftarrow$ the entry next to $\ddot{z}$ in Θ_E
16:     **end if**
17: **end while**

as z^{SE} and z^{NW} in line 3. Since there are no ESN points between z^1 and z^{18}, the algorithm terminates after the first execution of the *while*. Based on this example, Algorithm 2 provides a very ineffective approximation of Θ^*; the set Θ^* consists of infinitely many number of nondominated points comprising four line segments and three isolated points while Algorithm 2 finds only two of these nondominated points, i.e. z^1 and z^{18}.

In BOLPs, sorted Θ_E is representative of Θ^* since Θ^* is the set of all line segments that are generated by any two consecutive entries of Θ_E. This is not, however, extendable to BOMILPs since the line segments may be infeasible. Hence, Θ_E can be an approximation of Θ^* in BOMILPs. Algorithm 2 finds Θ_E by $2|\Theta_E| - 3$ iterations given $|\Theta_E| > 2$ [Aneja and Nair, 1979]. Thus, if $|\Theta_E| > 2$, the Can algorithm requires resolution of $2|\Theta_E| + 1$ MILP problems. Therefore, the efficiency of the algorithm is $\varphi(\text{Can}) = \frac{|\Theta_E|}{2|\Theta_E|+1}$. For large $|\Theta_E|$, $\varphi(\text{Can}) \approx \frac{1}{2}$, meaning that Can almost generates a nondominated point by two MILP resolutions. Based on φ, AUGMECON performs better than CAN; however, CAN is particularly useful in certain applications since it generates all ESN points.

The generalization of CAN for more than two objectives involves difficulties and is not guaranteed to find the entire Θ_E [Solanki et al., 1993, Przybylski et al., 2010]. Effective exact algorithms to find Θ_E for more than two objectives presented by [Przybylski et al., 2010, Özpeynirci and Köksalan, 2010]. In the two-phase method [Ulungu, 1995], the CAN algorithm is used in the first phase to find Θ_E that is used as an input for the second phase (for example [Pedersen et al., 2008, Przybylski et al., 2008, Przybylski et al., 2011, Raith and Ehrgott, 2009]).

4.2 EnpoBomip

In this section, we establish necessary fundamentals of our proposed exact algorithm. The ENPOBOMIP starts from z^{SE} and explores the nondominated frontier moving towards z^{NW}. The South-East point is found using lexicographic optimization LX($\text{P}^{\text{SE}}_{\text{LX1}}$,$\text{P}^{\text{SE}}_{\text{LX2}}$) and is a nondominated solution.

THEOREM 1. *Suppose that $\chi^{\text{SE}} = (x^{\text{SE}}, y^{\text{SE}}, z^{\text{SE}})$ is found by lexicographic optimization LX($P^{\text{SE}}_{\text{LX1}}$,$P^{\text{SE}}_{\text{LX2}}$). Then, $\chi^{\text{SE}} \in \Phi^*$ and $\nexists \bar{\chi} \in \Phi^*$ s.t. $\bar{z}_1 > z_1^{\text{SE}}$ or $\bar{z}_2 < z_2^{\text{SE}}$. Moreover, z^{SE} is an extreme point of $\Theta(y^{\text{SE}})$.*

PROOF OF THEOREM 1.

The first part is referred to [Cohon, 2013, Aneja and Nair, 1979]. We only prove that z^{SE} is an extreme point of $\Theta(y^{\text{SE}})$. Suppose to the contrary that z^{SE} is not an extreme point of $\Theta(y^{\text{SE}})$, hence $\exists \dot{z}, \ddot{z} \in \Theta(y^{\text{SE}})$ s.t. $z^{\text{SE}} = \lambda\dot{z} + (1-\lambda)\ddot{z}$, $0 < \lambda < 1$. Also suppose w.o.l.g. $\dot{z}_1 \leq \ddot{z}_1$. Two cases are possible:

1. $\dot{z}_2 \leq \ddot{z}_2$**.** In this case $\dot{z}_1 \leq z_1^{\text{SE}}$, $\dot{z}_2 \leq z_2^{\text{SE}}$, and $\dot{z} \neq z^{\text{SE}}$ hence $\dot{z} \leq z^{\text{SE}}$ i.e. $\dot{z}$ dominates z^{SE} (contradiction).

2. $\dot{z}_2 > \ddot{z}_2$**.** In this case $\ddot{z}_1 > z_1^{\text{SE}}$ and $\ddot{z}_2 < z_2^{\text{SE}}$. $\ddot{z}$ is either a nondominated point or dominated by a nondominated point $\bar{z}$ s.t. $\bar{z}_1 \leq \ddot{z}_1$ and $\bar{z}_2 \leq \ddot{z}_2$. Hence, there exists a nondominated point $\bar{z}$ s.t. $\bar{z}_2 < z_2^{\text{SE}}$ (contradiction).

□

EnpoBomip begins exploring nondominated frontier from an extreme point and it always moves on the boundaries of $\Theta(\bar{y})$ for some $\bar{y} \in Y$. The importance of boundaries is that they are the only candidates for nondominated points. Note that there cannot be any nondominated point in the interior of $\Theta(\bar{y})$. Consider $\chi^{\text{SE}} = (\bar{x}, y^1, z^1)$ in Fig. 3.1 that is obtained by solving LX($\text{P}^{\text{SE}}_{\text{LX1}}$,$\text{P}^{\text{SE}}_{\text{LX2}}$), where $\bar{x} \in S(y^1)$. Based on Theorem 1 $z^1 \in \Theta^*$ hence $\chi^{\text{SE}} \in \Phi^*$. The point z^1 is the right-most lower-most nondominated point in the objective space and is an extreme point of $\Theta(y^1)$.

EnpoBomip aims to distinguish nondominated line segments between two boundary points. For instance in Fig. 3.1, if the boundary points z^{14} and z^{18} are known, it is desired to figure out that $[z^{14}, z^{18}]$ is nondominated. Note that although z^{12} and z^{14} are nondominated points, $[z^{12}, z^{14}]$ is not even feasible. The points z^7 and z^{11} are nondominated and $[z^7, z^{11}]$ is feasible, but $[z^7, z^{11}]$ is not nondominated. For feasibility, it is enough for points to be associated with same y-values. For example, since z^{12} and z^{14} have different y-values, $[z^{12}, z^{14}]$ is not necessarily feasible.

Suppose that $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ and $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$ are feasible solutions. Note that $\dot{x}, \ddot{x} \in S(\bar{y})$, $\dot{z}, \ddot{z} \in \Theta(\bar{y})$, and $[\dot{\chi}, \ddot{\chi}]$ is feasible for problem P. A sufficient condition for efficiency of $[\dot{\chi}, \ddot{\chi}]$ is $[\dot{z}, \ddot{z}]^{\leq} \cap \Theta = \emptyset$. For example in Fig. 3.1, $[z^{14}, z^{18}]^{\leq} \cap \Theta = \emptyset$, hence $[z^{14}, z^{18}]$ is nondominated; however, $[z^7, z^{11}]^{\leq} \cap \Theta \neq \emptyset$ hence $[z^7, z^{11}]$ is not nondominated. The set $[\dot{z}, \ddot{z}]^{\leq}$ cannot be linearly formulated. Theorem 2 presents two sufficient conditions to conclude that $[\dot{z}, \ddot{z}]^{\leq} \cap \Theta = \emptyset$. The conditions of Theorem 2 can be handled without using nonlinear formulations.

THEOREM 2. *Let $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ and $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$ be feasible solutions of problem P. If $[\dot{z}, \ddot{z}]^{\leq}$ has empty intersection with $\Theta(\bar{y})$ and if $[\dot{z}, \ddot{z}]^{\leqq}$ has empty intersection* *with $\Theta(Y\backslash\{\bar{y}\})$, then $[\dot{\chi}, \ddot{\chi}]$ is an efficient solution of problem P.*

PROOF OF THEOREM 2. Let $\chi^* = (x^*, \bar{y}, z^*)$ be a convex combination of $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ and $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$. Note that $x^* \in S(\bar{y})$; hence, $(x^*, \bar{y}) \in S$. The requirement for efficiency of χ^* is $z^{*\leq} \cap \Theta = \emptyset$. Note that $\Theta = \Theta(Y\backslash\{\bar{y}\}) \cup \Theta(\bar{y})$. Hence,

$$\begin{aligned} z^{*\leq} \cap \Theta \quad &= z^{*\leq} \cap \left(\Theta(Y\backslash\{\bar{y}\}) \cup \Theta(\bar{y})\right) \\ &= \left(z^{*\leq} \cap \Theta(Y\backslash\{\bar{y}\})\right) \cup \left(z^{*\leq} \cap \Theta(\bar{y})\right) = \emptyset \qquad (4.10) \\ \Rightarrow \quad & z^{*\leq} \cap \Theta(Y\backslash\{\bar{y}\}) = \emptyset \text{ and } z^{*\leq} \cap \Theta(\bar{y}) = \emptyset \qquad (4.11) \end{aligned}$$

Since, $z^{*\leq} \subset z^{*\leqq}$,

$$z^{*\leqq} \cap \Theta(Y\backslash\{\bar{y}\}) = \emptyset \Rightarrow z^{*\leq} \cap \Theta(Y\backslash\{\bar{y}\}) = \emptyset \qquad (4.12)$$

Hence,

$$\begin{bmatrix} z^{*\leqq} \cap \Theta(Y\backslash\{\bar{y}\}) = \emptyset \\ \text{and } z^{*\leq} \cap \Theta(\bar{y}) = \emptyset \end{bmatrix} \Rightarrow z^{*\leq} \cap \Theta = \emptyset \qquad (4.13)$$

Therefore, if $z^{*\leq}$ has empty intersection with $\Theta(\bar{y})$ and if $z^{*\leqq}$ has empty intersection with $\Theta(Y\backslash\{\bar{y}\})$, then z^* is a nondominated point and χ^* is efficient solution of problem P. $\square$

Hence, for two feasible solutions with same y-values, the two conditions of Theorem 2 must hold to conclude the efficiency of their convex combination. For instance in Fig. 3.1, $[z^{17}, z^{18}]^{\leq}$ has empty intersection with $\Theta(y^{12})$ and $[z^{17}, z^{18}]^{\leqq}$ has empty intersection with $\Theta(Y\backslash\{y^{12}\})$; hence, $[z^{17}, z^{18}]$ is nondominated. For points z^7 and z^{11}, $[z^7, z^{11}]^{\leq}$ has empty intersection with $\Theta(y^4)$, but $[z^7, z^{11}]^{\leqq}$ has nonempty intersection with $\Theta(Y\backslash\{y^4\})$; hence, nothing can be concluded about $[z^7, z^{11}]$. Note that if conditions of Theorem 2 does not hold, this does not mean that the line segment contains dominated points. For example in Fig. 3.1, $[z^{14}, z^{18}]^{\leq}$ has empty intersection with $\Theta(y^{12})$ but $[z^{14}, z^{18}]^{\leqq}$ has nonempty intersection with $\Theta(Y\backslash\{y^{12}\})$, however, $[z^{14}, z^{18}]$ is a nondominated line segment.

Now, we establish a procedure to check the conditions of Theorem 2. Let $\chi^{\text{CP}} = (x^{\text{CP}}, y^{\text{CP}}, z^{\text{CP}})$ denote the current point that is being explored by the algorithm. Two conditions hold for χ^{CP} throughout the algorithm: $z^{\text{CP}} \in \partial\Theta(y^{\text{CP}})$ and $z^{\text{CP}\leq} \cap \Theta = \emptyset$ (see section 4.3). We say z^{NEP} is the next-lower extreme point with respect to z^{CP}, if z^{NEP} satisfies the following: z^{NEP} is an extreme point of $\Theta(y^{\text{CP}})$, $z_1^{\text{NEP}} < z_1^{\text{CP}}$, and $[z^{\text{CP}}, z^{\text{NEP}}]^{\leq}$ has empty intersection with $\Theta(y^{\text{CP}})$. We also say $\chi^{\text{NEP}} = (x^{\text{NEP}}, y^{\text{CP}}, z^{\text{NEP}})$ is the next-lower extreme point with respect to χ^{CP}. For instance in Fig. 3.1, z^{11} is the next-lower extreme point with respect to z^7, z^8, z^{10} or any point in $[z^7, z^{11})$. Observe that some points do not have next-lower extreme points, e.g. z^{12} and z^{20}. Note that if the next-lower extreme point, z^{NEP}, with respect to z^{CP} exists and is accurately determined, the first condition of theorem 2 is satisfied. For the second condition, we can solve problem $\text{P}_{\text{LX1}}^{\text{NCC}}$. (NCC: nondominated convex combination)

$$
\begin{aligned}
\text{P}_{\text{LX1}}^{\text{NCC}}(y^{\text{CP}}, z^{\text{CP}}, z^{\text{NEP}}) : \qquad & \check{z}_2 = \min z_2 && (4.14)\\
\text{s.t.} \qquad & z_2 \leq m z_1 + b, z_1 \leq z_1^{\text{CP}}, z_2 \leq z_2^{\text{NEP}} && (4.15)\\
& y \neq y^{\text{CP}}, (x, y) \in S && (4.16)
\end{aligned}
$$

where $m = \frac{z_2^{\text{NEP}} - z_2^{\text{CP}}}{z_1^{\text{NEP}} - z_1^{\text{CP}}}$ and $b = z_2^{\text{CP}} - m z_1^{\text{CP}}$. conditions on z in Eq. (4.15) is equivalent to $z \in [z^{\text{CP}}, z^{\text{NEP}}]^{\leqq}$. Hence, the feasible region of $\text{P}_{\text{LX1}}^{\text{NCC}}$ in the objective space is precisely the formulation of $[z^{\text{CP}}, z^{\text{NEP}}]^{\leqq} \cap \Theta(Y\backslash\{y^{\text{CP}}\})$. Therefore, if $\text{P}_{\text{LX1}}^{\text{NCC}}$ is infeasible, then $[z^{\text{CP}}, z^{\text{NEP}}]^{\leqq} \cap \Theta(Y\backslash\{y^{\text{CP}}\}) = \emptyset$; hence, based on Theorem 2, $[z^{\text{CP}}, z^{\text{NEP}}]$ is a nondominated line segment and $[\chi^{\text{CP}}, \chi^{\text{NEP}}]$ is an efficient solution of problem P.

Observe that $\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX1}}$ is not linear due to Eq. (4.16). The relationship, $y \neq \bar{y}$, can be formulated using no-good constraints [Hooker, 1994]. If $y_1, \cdots, y_p$ are binary variables, a no-good constraint to exclude $\bar{y}$ from Y is as follows.

$$\sum_{i:\bar{y}_i=1} y_i - \sum_{i:\bar{y}_i=0} y_i \leq (\sum_{i:\bar{y}_i=1} 1) - 1 \tag{4.17}$$

Note that $\bar{y}$ is infeasible for this inequality while any other solution is feasible. If $y_1, \cdots, y_p$ are general integer variables, no-good constraints require adding a new binary variable to the problem. Suppose without loss of generality that y_i's can take value between 0 and ϑ_i, i.e. $y_i \in \mathbb{Z}$, $0 \leq y_i \leq \vartheta_i$, $i = 1, \cdots, p$. We define $\vartheta = \max_i\{\vartheta_i\} + 1$. Then, no-good constraints to exclude $\bar{y}$ can be written as follows.

$$\begin{aligned}
&\sum_i \vartheta^{i-1} y_i - \sum_i \vartheta^{i-1} \bar{y}_i \leq -1 + u \left(1 + \sum_i \vartheta^{i-1} \vartheta_i\right) \\
&\sum_i \vartheta^{i-1} y_i - \sum_i \vartheta^{i-1} \bar{y}_i \geq 1 - (1-u) \left(1 + \sum_i \vartheta^{i-1} \vartheta_i\right) \\
&u \in \{0, 1\}
\end{aligned} \tag{4.18}$$

In the remaining, we use the notation $y \neq \bar{y}$ to indicate $Y \backslash \{\bar{y}\}$; however, no-good constraints can be used in implementations.

The second possibility when solving $\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX1}}$ is the feasibility of this problem. Note that $\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX1}}$ cannot be unbounded due to our assumption, i.e. $\{\min z_j \text{ s.t. } (x, y) \in S\} \neq -\infty, j = 1, 2$; hence, in the case of feasibility, $\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX1}}$ has optimal solution that implies a nondominated point in $[z^{\mathrm{NEP}}, z^{\mathrm{CP}}]^{\leqq}$. The problem $\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX2}}$ finds the right-most nondominated point in $[z^{\mathrm{NEP}}, z^{\mathrm{CP}}]^{\leqq} \cap \Theta(Y \backslash \{y^{\mathrm{CP}}\})$.

$$\mathrm{P}^{\text{NCC}}_{\text{LX2}}(y^{\text{CP}}, \check{z}_2): \qquad \check{z}_1 = \min z_1 \tag{4.19}$$

$$\text{s.t.} \qquad z_2 = \check{z}_2 \tag{4.20}$$

$$y \neq y^{\text{CP}}, (x, y) \in S \tag{4.21}$$

Note that the feasible region of $\mathrm{P}^{\text{NCC}}_{\text{LX2}}$, i.e. $\{z_2 = \check{z}_2\} \bigcap \Theta(Y \backslash \{y^{\text{CP}}\})$, is not empty due to the result of $\mathrm{P}^{\text{NCC}}_{\text{LX1}}$. Also note that we do not need to include Eq. (4.15) in $\mathrm{P}^{\text{NCC}}_{\text{LX2}}$, since it is guaranteed.

THEOREM 3. *Let $\chi^{\text{CP}} = (x^{\text{CP}}, y^{\text{CP}}, z^{\text{CP}})$ be a feasible solution of problem P s.t. $z^{\text{CP}} \in \partial\Theta(y^{\text{CP}})$ and $z^{\text{CP}\leq} \cap \Theta = \emptyset$. Suppose that χ^{NEP}, the next-lower extreme point with respect to χ^{CP}, exists. If $P^{\text{NCC}}_{\text{LX1}}$ is infeasible, then $[z^{\text{CP}}, z^{\text{NEP}}]^{\leqq} \cap \Theta(Y \backslash \{y^{\text{CP}}\}) = \emptyset$. If $P^{\text{NCC}}_{\text{LX1}}$ is feasible, solve $P^{\text{NCC}}_{\text{LX2}}$ and denote the solution by $\check{\chi} = (\check{x}, \check{y}, \check{z})$. Suppose that $\check{z} \neq z^{\text{CP}}$. Then,*

(a) *$\check{\chi}$ is an efficient solution of problem P,*

(b) *$(\tilde{\chi}, \chi^{\text{CP}}]$ is efficient, where $\tilde{z} = (\frac{\check{z}_2 - b}{m}, \check{z}_2)$, $\tilde{x} = \left(\frac{\tilde{z}_2 - z_2^{\text{CP}}}{z_2^{\text{NEP}} - z_2^{\text{CP}}}\right)(x^{\text{NEP}} - x^{\text{CP}}) + x^{\text{CP}}$, $\tilde{y} = y^{\text{CP}}$,*

(c) *If $\check{z} \notin [z^{\text{CP}}, z^{\text{NEP}}]$, there is no nondominated point z^* s.t. $\check{z}_1 < z_1^* \leq \tilde{z}_1$.*

PROOF OF THEOREM 3. The feasible region of $\mathrm{P}^{\text{NCC}}_{\text{LX1}}$ in objective space is precisely the formulation of $[z^{\text{NEP}}, z^{\text{CP}}]^{\leqq} \cap \Theta(Y \backslash \{y^{\text{CP}}\})$; hence, if $\mathrm{P}^{\text{NCC}}_{\text{LX1}}$ is infeasible, then $[z^{\text{NEP}}, \dot{z}]^{\leqq} \cap \Theta(Y \backslash \{\bar{y}\}) = \emptyset$.

Proof of (a). If $\mathrm{P}^{\text{NCC}}_{\text{LX1}}$ is feasible, then $\check{z}_2$ exists and $\check{z}_2^{<}$ has empty intersection with the feasible region of $\mathrm{P}^{\text{NCC}}_{\text{LX1}}$. Then,

$$
\begin{aligned}
&\check{z}_2^{<} \cap \left([z^{\text{NEP}}, z^{\text{CP}}]^{\leqq} \cap \Theta(Y \backslash \{y^{\text{CP}}\})\right) = \emptyset \\
&\Rightarrow \left(\check{z}_2^{<} \cap [z^{\text{NEP}}, z^{\text{CP}}]^{\leqq}\right) \cap \Theta(Y \backslash \{y^{\text{CP}}\}) = \emptyset \qquad (4.22) \\
&\text{since } \check{z}^{\leq} \subset [z^{\text{NEP}}, z^{\text{CP}}]^{\leqq} \quad \Rightarrow \left(\check{z}_2^{<} \cap \check{z}^{\leq}\right) \cap \Theta(Y \backslash \{y^{\text{CP}}\}) = \emptyset \qquad (4.23)
\end{aligned}
$$

Note that feasibility of $\text{P}_{\text{LX1}}^{\text{NCC}}$ implies the feasibility of $\text{P}_{\text{LX2}}^{\text{NCC}}$; hence, $\check{z}_1$ exists and $\check{z}_1^{<}$ has empty intersection with the feasible region of $\text{P}_{\text{LX2}}^{\text{NCC}}$. Then,

$$
\begin{aligned}
&\check{z}_1^{<} \cap (\{z_2 = \check{z}_2\} \cap \Theta(Y \backslash \{y^{\text{CP}}\})) = \emptyset \\
&\Rightarrow (\check{z}_1^{<} \cap \{z_2 = \check{z}_2\}) \cap \Theta(Y \backslash \{y^{\text{CP}}\}) = \emptyset \qquad (4.24)
\end{aligned}
$$

Eqs. (4.23) and (4.24) implies that,

$$
\left(\left(\check{z}_2^{<} \cap \check{z}^{\leq}\right) \cup (\check{z}_1^{<} \cap \{z_2 = \check{z}_2\})\right) \cap \Theta(Y \backslash \{y^{\text{CP}}\}) = \emptyset \qquad (4.25)
$$

And we have,

$$
\begin{aligned}
&\begin{bmatrix} \left(\check{z}_2^{<} \cap \check{z}^{\leq}\right) = \{z_2 < \check{z}_2, z_1 \leq \check{z}_1\} \\ (\check{z}_1^{<} \cap \{z_2 = \check{z}_2\}) = \{z_2 = \check{z}_2, z_1 < \check{z}_1\} \end{bmatrix} \\
&\Rightarrow \left(\left(\check{z}_2^{<} \cap \check{z}^{\leq}\right) \cup (\check{z}_1^{<} \cap \{z_2 = \check{z}_2\})\right) = \check{z}^{\leq} \qquad (4.26)
\end{aligned}
$$

Based on Eqs. (4.25) and (4.26),

$$\check{z}^{\leq} \cap \Theta(Y \backslash \{y^{\text{CP}}\}) = \emptyset \tag{4.27}$$

Moreover, since χ^{NEP} is the next-lower extreme point with respect to χ^{CP} and $\check{z} \in [z^{\text{CP}}, z^{\text{NEP}}]^{\leqq}$, then

$$\check{z}^{\leq} \cap \Theta(y^{\text{CP}}) = \emptyset \tag{4.28}$$

Eqs. (4.27) and (4.28) implies that,

$$\check{z}^{\leq} \cap (\Theta(Y \backslash \{y^{\text{CP}}\}) \cup \Theta(y^{\text{CP}})) = \check{z}^{\leq} \cap \Theta = \emptyset \tag{4.29}$$

Therefore, $\check{z}$ is nondominated point and $\check{\chi}$ is efficient solution of problem P.

Proof of (b). Note that $\tilde{z} \in [z^{\text{NEP}}, z^{\text{CP}}]$ and $\tilde{z}_2 = \check{z}_2$; hence, $\tilde{z}_1$ can be calculated as $\frac{\check{z}_2 - b}{m}$ and $\tilde{x}$ is calculated using the following equation.

$$\frac{\tilde{x} - x^{\text{CP}}}{x^{\text{NEP}} - x^{\text{CP}}} = \frac{\tilde{z}_2 - z_2^{\text{CP}}}{z_2^{\text{NEP}} - z_2^{\text{CP}}} \Rightarrow \tilde{x} = \left(\frac{\tilde{z}_2 - z_2^{\text{CP}}}{z_2^{\text{NEP}} - z_2^{\text{CP}}}\right)(x^{\text{NEP}} - x^{\text{CP}}) + x^{\text{CP}}$$

Since $(\tilde{z}, z^{\text{CP}}] \subset [z^{\text{NEP}}, z^{\text{CP}}]$ and since χ^{NEP} is the next-lower extreme point with respect to χ^{CP}, then

$$(\tilde{z}, z^{\text{CP}}]^{\leq} \cap \Theta(y^{\text{CP}}) = \emptyset \tag{4.30}$$

Using Eq. (4.22), and since $(\tilde{z}, z^{\text{CP}}]^{\leq} \subset (\tilde{z}, z^{\text{CP}}]^{\leqq}$,

$$\left(\check{z}_2^{<} \cap (\tilde{z}, z^{\text{CP}}]^{\leq}\right) \cap \Theta(Y \backslash \{y^{\text{CP}}\}) = \emptyset \tag{4.31}$$

We have,

$$\left.\begin{array}{l}\left(\check{z}_2^{<} \cap (\tilde{z}, z^{\text{CP}}]^{\leq}\right) = (\tilde{z}, z^{\text{CP}}]^{\leq} \\ \text{Eqs. (4.30) and (4.31)}\end{array}\right. \Rightarrow (\tilde{z}, z^{\text{CP}}]^{\leq} \cap \Theta = \emptyset \tag{4.32}$$

Therefore, $(\tilde{\chi}, \chi^{\text{CP}}]$ is efficient and $(\tilde{z}, z^{\text{CP}}]$ is a nondominated line segment.

Proof of (c). Suppose to the contrary that there is a nondominated point z^* s.t. $\check{z}_1 < z_1^* \leq \tilde{z}_1$. Two cases are possible:

1. $z_2^* < \check{z}_2$**.** This contradicts Eq. (4.32).

2. $z_2^* \geq \check{z}_2$**.** In this case z^* is dominated by $\check{z}$ (contradiction).

□

For example, in Fig. 3.1, consider the current point z^7 with the next-lower extreme point z^{11}. If problems $\text{P}_{\text{LX1}}^{\text{NCC}}$ and $\text{P}_{\text{LX2}}^{\text{NCC}}$ are respectively solved, the algorithm actually searches for a point in $\Theta(Y \backslash \{y^4\})$ such that this point is located in $[z^7, z^{11}]^{\leqq}$ and is the lowermost leftmost point; hence, z^9 is found. Note that $z^9 \neq z^7$, then z^9 is a nondominated point, the line segment $(z^8, z^7]$ is nondominated and there is no nondominated point z^* s.t. $z_1^9 < z_1^* \leq z_1^8$. Moreover, z^8 can be calculated as $(\frac{z_2^9 - b}{m}, z_2^9)$ where m and b are the slope and z_2-intercept of the line passing z^7 and z^{11}. The associated x-value can be calculated based on ratios between z-values.

Note that the assumption $\check{z} \neq z^{\text{CP}}$ is necessary for parts (a)-(c) to be meaningful. For example, if we solve $\text{P}_{\text{LX1}}^{\text{NCC}}$ and $\text{P}_{\text{LX2}}^{\text{NCC}}$ for z^{14} and z^{18} excluding y^{12}, we will find z^{14}

since z^{14} is also in $\Theta(y^9)$ and $\Theta(y^{10})$. Note that the assumption does not hold. We call this case a tie and break it by determining y^9 and y^{10} and excluding them from Y (see section 4.3). If y^9 and y^{10} are excluded from Y, $\mathrm{P}^{\mathrm{NCC}}_{\mathrm{LX1}}$ becomes infeasible and based on Theorem 2, $[z^{14}, z^{18}]$ is a nondominated line segment.

Consider the current point $\chi^{\mathrm{CP}} = (x^{\mathrm{CP}}, y^{\mathrm{CP}}, z^{\mathrm{CP}})$ s.t. $z^{\mathrm{CP}} \in \partial\Theta(y^{\mathrm{CP}})$ and $z^{\mathrm{CP}\leq} \cap \Theta = \emptyset$. Also, suppose that the left-hand-side of z^{CP} in objective space ($z_1^{\mathrm{CP}<}$) should be explored. If $\Theta(y^{\mathrm{CP}})$ has nonempty intersection with this area in the objective space, we can find the next-lower extreme point with respect to χ^{CP}. The results of Theorems 2 and 3 can be used afterwards. The following LP problem shows if $z_1^{\mathrm{CP}<} \cap \Theta(y^{\mathrm{CP}}) = \emptyset$.

$$\mathrm{LP}^{\mathrm{NEP}}(y^{\mathrm{CP}}) : \phi = \min z_1 \text{ s.t. } x \in S(y^{\mathrm{CP}}) \tag{4.33}$$

If ϕ equals z_1^{CP}, then $\Theta(y^{\mathrm{CP}})$ has empty intersection with $z_1^{\mathrm{CP}<}$.

THEOREM 4. *Let $\chi^{\mathrm{CP}} = (x^{\mathrm{CP}}, y^{\mathrm{CP}}, z^{\mathrm{CP}})$ be a feasible solution of problem P s.t. $z^{\mathrm{CP}} \in \partial\Theta(y^{\mathrm{CP}})$ and $z^{\mathrm{CP}\leq} \cap \Theta = \emptyset$. Exactly one of the following cases holds:*

1. *$\phi = z_1^{\mathrm{CP}}$ and $z_1^{\mathrm{CP}<} \cap \Theta(y^{\mathrm{CP}}) = \emptyset$, or*

2. *$\phi < z_1^{\mathrm{CP}}$ and the next-lower extreme point with respect to χ^{CP} exists.*

PROOF OF THEOREM 4. Note that ϕ cannot be greater than z_1^{CP} since $x^{\mathrm{CP}} \in S(y^{\mathrm{CP}})$. Case 1 is obvious and we focus on proving case 2, where $\phi < z_1^{\mathrm{CP}}$. Recall that we assume in this paper $\min z_j \neq -\infty, j = 1, 2$; then, $\phi \neq -\infty$ and $\mathrm{LP}^{\mathrm{NEP}}$ has optimal solution. Thus, there exists an optimal solution $\bar{z}$ that is an extreme point of $\Theta(y^{\mathrm{CP}})$. Therefore, either $\bar{\chi}$ is the next-lower extreme point with respect to χ^{CP} or a next-lower extreme point can be found. Note that the number of extreme points of $\Theta(y^{\mathrm{CP}})$ is finite. $\square$

For example, Consider the current point z^3 in Fig. 3.1. Note that $z^{3\leq} \cap \Theta = \emptyset$ holds since y^1 has been permanently excluded from Y in previous iterations (see section 4.3).

Solving $\mathrm{LP}^{\mathrm{NEP}}(y^2)$ results in $\phi = 14$ that equals z_1^3. Therefore, $z_1^{3<} \cap \Theta(y^2) = \emptyset$. However, if the algorithm is at the current point z^5, solving $\mathrm{LP}^{\mathrm{NEP}}(y^3)$ results in $\phi = 13 < z_1^5$. This means that the next-lower extreme point with respect to z^5 exists.

If $z_1^{\mathrm{CP}<}$ has empty intersection with $\Theta(y^{\mathrm{CP}})$, then $\Theta(y^{\mathrm{CP}})$ contains no nondominated point in $z_1^{\mathrm{CP}<}$; hence, this area can be explored excluding y^{CP} from Y. Thus, if the first case of theorem 2 happens, the following lexicographic optimization is used to find the next current point in $z_1^{\mathrm{CP}\leq}$. (NP: next point)

$$\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX1}}(y^{\mathrm{CP}}, z_1^{\mathrm{CP}}) : z_2^{\mathrm{NP}} = \min z_2 \text{ s.t. } z_1 \leq z_1^{\mathrm{CP}}, y \neq y^{\mathrm{CP}}, (x, y) \in S, \tag{4.34}$$

$$\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX2}}(z_2^{\mathrm{NP}}) : z_1^{\mathrm{NP}} = \min z_1 \text{ s.t. } z_2 = z_2^{\mathrm{NP}}, y \neq y^{\mathrm{CP}}, (x, y) \in S \tag{4.35}$$

Infeasibility of $\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX1}}$ shows that there is no nondominated point in $z_1^{\mathrm{CP}<}$. In the case of feasibility, however, $\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX2}}$ is solved. If $z_1^{\mathrm{NP}} < z_1^{\mathrm{CP}}$ or $z^{\mathrm{NP}} = z^{\mathrm{CP}}$, then z^{NP} is nondominated point and χ^{NP} that is found from solving $\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX2}}$ is an efficient solution.

THEOREM 5. *Let $\chi^{\mathrm{CP}} = (x^{\mathrm{CP}}, y^{\mathrm{CP}}, z^{\mathrm{CP}})$ be a feasible solution of problem P s.t. $z^{\mathrm{CP}} \in \partial\Theta(y^{\mathrm{CP}})$, $z^{\mathrm{CP}\leq} \cap \Theta = \emptyset$, and $z_1^{\mathrm{CP}<} \cap \Theta(y^{\mathrm{CP}}) = \emptyset$. If $P^{\mathrm{NP}}_{\mathrm{LX1}}$ is infeasible, then there is no nondominated point in $z_1^{\mathrm{CP}<}$.*

PROOF OF THEOREM 5. The feasible region of $\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX1}}$ is precisely the formulation of $\left(z_1^{\mathrm{CP}\leq} \cap \Theta(Y \backslash \{y^{\mathrm{CP}}\})\right)$. Infeasibility of $\mathrm{P}^{\mathrm{NP}}_{\mathrm{LX1}}$ is equivalent to $\left(z_1^{\mathrm{CP}\leq} \cap \Theta(Y \backslash \{y^{\mathrm{CP}}\})\right) = \emptyset$. Note that $z_1^{\mathrm{CP}<} \subset z_1^{\mathrm{CP}\leq}$, therefore $(z_1^{\mathrm{CP}<} \cap \Theta(Y \backslash \{y^{\mathrm{CP}}\})) = \emptyset$. And since we suppose $z_1^{\mathrm{CP}<} \cap \Theta(y^{\mathrm{CP}}) = \emptyset$, then

$$z_1^{\text{CP}<} \cap (\Theta(Y \backslash \{y^{\text{CP}}\}) \cup \Theta(y^{\text{CP}})) = z_1^{\text{CP}<} \cap \Theta = \emptyset \tag{4.36}$$

Therefore, there is no nondominated point in $z_1^{\text{CP}<}$. $\square$

Theorem 5 outlines the termination of our algorithm. For example, consider the current point z^{18} in Fig. 3.1. Note that $z_1^{18<} \cap \Theta(y^{12}) = \emptyset$. The problem $\text{P}_{\text{LX1}}^{\text{NP}}$, i.e. $\min z_2$ s.t. $z_1 \leq 1, y \neq y^{12}, (x,y) \in S$, is infeasible meaning that no nondominated point exists in $z_1^{18<}$ and the algorithm stops.

THEOREM 6. *Let $\chi^{\text{CP}} = (x^{\text{CP}}, y^{\text{CP}}, z^{\text{CP}})$ be a feasible solution of problem P s.t. $z^{\text{CP}} \in \partial\Theta(y^{\text{CP}})$, $z^{\text{CP}\leq} \cap \Theta = \emptyset$, and $z_1^{\text{CP}<} \cap \Theta(y^{\text{CP}}) = \emptyset$. Suppose that $P_{\text{LX1}}^{\text{NP}}$ is feasible, solve $P_{\text{LX2}}^{\text{NP}}$ and let $\chi^{\text{NP}} = (x^{\text{NP}}, y^{\text{NP}}, z^{\text{NP}})$ be the optimal solution of $P_{\text{LX2}}^{\text{NP}}$. Then,*

(a) *If $z_1^{\text{NP}} = z_1^{\text{CP}}$, z^{NP} is (weakly) dominated.*

(b) *If $z_1^{\text{NP}} < z_1^{\text{CP}}$, then χ^{NP} is an efficient solution, and there is no efficient solution χ^* s.t. $z_1^{\text{NP}} < z_1^* < z_1^{\text{CP}}$.*

PROOF OF THEOREM 6. ***Proof of (a).*** Since $z_2^{\text{NP}} \geq z_2^{\text{CP}}$ and $z_1^{\text{NP}} = z_1^{\text{CP}}$, then z^{CP} (weakly) dominates z^{NP}.

Proof of (b). Since $z_1^{\text{NP}} < z_1^{\text{CP}}$, then $z_1^{\text{NP}\leq} \subset z_1^{\text{CP}<}$, and since $z^{\text{NP}\leq} \subset z_1^{\text{NP}\leq}$, then $z^{\text{NP}\leq} \subset z_1^{\text{CP}<}$. As we suppose that $z_1^{\text{CP}<} \cap \Theta(y^{\text{CP}}) = \emptyset$, then

$$z^{\text{NP}\leq} \cap \Theta(y^{\text{CP}}) = \emptyset \tag{4.37}$$

If $\text{P}_{\text{LX1}}^{\text{NP}}$ is feasible, then z_2^{NP} exists and $z_2^{\text{NP}<}$ has empty intersection with the feasible region of $\text{P}_{\text{LX1}}^{\text{NP}}$. Then,

$$\begin{aligned}
& z_2^{\text{NP}<} \cap \left(z_1^{\text{CP}\leq} \cap \Theta(Y\backslash\{y^{\text{CP}}\})\right) = \emptyset \\
& \Rightarrow \left(z_2^{\text{NP}<} \cap z_1^{\text{CP}\leq}\right) \cap \Theta(Y\backslash\{y^{\text{CP}}\}) = \emptyset \qquad (4.38) \\
& \text{since } z_1^{\text{NP}\leq} \subset z_1^{\text{CP}\leq} \quad \Rightarrow \left(z_2^{\text{NP}<} \cap z_1^{\text{NP}\leq}\right) \cap \Theta(Y\backslash\{y^{\text{CP}}\}) = \emptyset \qquad (4.39)
\end{aligned}$$

Note that feasibility of $\text{P}_{\text{LX1}}^{\text{NP}}$ implies the feasibility of $\text{P}_{\text{LX2}}^{\text{NP}}$; hence, z_1^{NP} exists and $z_1^{\text{NP}<}$ has empty intersection with the feasible region of $\text{P}_{\text{LX2}}^{\text{NP}}$. Then,

$$\begin{aligned}
& z_1^{\text{NP}<} \cap (\{z_2 = z_2^{\text{NP}}\} \cap \Theta(Y\backslash\{y^{\text{CP}}\})) = \emptyset \\
& \Rightarrow (z_1^{\text{NP}<} \cap \{z_2 = z_2^{\text{NP}}\}) \cap \Theta(Y\backslash\{y^{\text{CP}}\}) = \emptyset \qquad (4.40)
\end{aligned}$$

Eqs. (4.39) and (4.40) implies that,

$$\begin{aligned}
& \left(\left(z_2^{\text{NP}<} \cap z_1^{\text{NP}\leq}\right) \cup (z_1^{\text{NP}<} \cap \{z_2 = z_2^{\text{NP}}\})\right) \cap \Theta(Y\backslash\{y^{\text{CP}}\}) = \\
& (\{z_1 \leq z_1^{\text{NP}}, z_2 < z_2^{\text{NP}}\} \cup \{z_2 = z_2^{\text{NP}}, z_1 < z_1^{\text{NP}}\}) \cap \Theta(Y\backslash\{y^{\text{CP}}\}) = \\
& z^{\text{NP}\leq} \cap \Theta(Y\backslash\{y^{\text{CP}}\}) = \emptyset \\
& \text{using Eq. (4.37)} \Rightarrow z^{\text{NP}\leq} \cap (\Theta(Y\backslash\{y^{\text{CP}}\}) \cup \Theta(y^{\text{CP}})) = z^{\text{NP}\leq} \cap \Theta = \emptyset \quad (4.41)
\end{aligned}$$

Therefore, z^{NP} is nondominated point and χ^{NP} is efficient solution of problem P.

Now, we should prove that there is no efficient solution χ^* s.t. $z_1^{\text{NP}} < z_1^* < z_1^{\text{CP}}$. Suppose to the contrary that there is such an efficient solution χ^*. Two cases are possible:

1. $z_2^* < z_2^{NP}$**.** In this case, $z^* \in (z_2^{\text{NP}<} \cap z_1^{\text{CP}<})$ and it follows that,

$$
\begin{aligned}
&(z_2^{\text{NP}<} \cap z_1^{\text{CP}<}) \cap \Theta \neq \emptyset \\
&\Rightarrow (z_2^{\text{NP}<} \cap z_1^{\text{CP}<}) \cap (\Theta(Y\backslash\{y^{\text{CP}}\}) \cup \Theta(y^{\text{CP}})) \neq \emptyset \\
&\Rightarrow (z_2^{\text{NP}<} \cap z_1^{\text{CP}<} \cap \Theta(Y\backslash\{y^{\text{CP}}\})) \cup (z_2^{\text{NP}<} \cap z_1^{\text{CP}<} \cap \Theta(y^{\text{CP}})) \neq \emptyset \\
&\text{Eq. (4.38) and since } z_1^{\text{CP}<} \subset z_1^{\text{CP}\leq} \Rightarrow \emptyset \cup (z_2^{\text{NP}<} \cap z_1^{\text{CP}<} \cap \Theta(y^{\text{CP}})) \neq \emptyset \\
&\Rightarrow z_2^{\text{NP}<} \cap (z_1^{\text{CP}<} \cap \Theta(y^{\text{CP}})) \neq \emptyset \\
&\text{since } z_1^{\text{CP}<} \cap \Theta(y^{\text{CP}}) = \emptyset \Rightarrow z_2^{\text{NP}<} \cap \emptyset \neq \emptyset \\
&\Rightarrow \emptyset \neq \emptyset \text{ (contradiction)}
\end{aligned}
$$

2. $z_2^* \geq z_2^{NP}$**.** In this case z^* is dominated by z^{NP} (contradiction).

□

Consider the current point z^{12} in Fig. 3.1 and note that $z_1^{12<} \cap \Theta(y^7) = \emptyset$. Solving $\text{P}_{\text{LX1}}^{\text{NP}}$ and $\text{P}_{\text{LX2}}^{\text{NP}}$, we find z^{13}. Since $z_1^{13} = z_1^{12}$, then part (a) of Theorem 6 applies and z^{13} is dominated by z^{12}. The algorithm proceeds to the current point z^{13} permanently excluding y^7 from Y. Note that $z_1^{13<} \cap \Theta(y^8) = \emptyset$. Since y^7 was permanently excluded from Y, then $z^{13\leq} \cap \Theta = \emptyset$ holds. Problem $\text{P}_{\text{LX1}}^{\text{NP}}$ and $\text{P}_{\text{LX2}}^{\text{NP}}$ are solved and z^{14} is obtained. Since $z_1^{14} < z_1^{13}$, part (b) applies; z^{14} is nondominated and there is no nondominated point z^* s.t. $z_1^{14} < z_1^* < z_1^{13}$.

4.3 Algorithm of EnpoBomip

In this section, we present our novel algorithm, EnpoBomip, to find the exact nondominated points, with all alternative y-values, of general BOMILPs. We first propose a procedure to find the next-lower extreme point. Suppose that $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ is a feasible solution of problem P and the next-lower extreme point, $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$, with respect

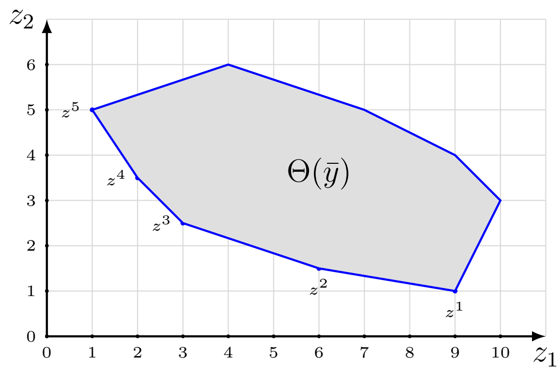


Figure 4.2: An illustrative example for procedure FindNep

to $\dot{\chi}$ exists, the procedure FindNep, given in Algorithm 3, is used to find it. The first candidate for $\ddot{\chi}$ is found in Line (2). In Lines (4)-(5), $\ddot{\chi}$ is checked to know if it is the appropriate next-lower extreme point. The LP in Line 5 finds the smallest possible z_2-intercept of the line in objective space that contains feasible points and has the same slope with the line passing $\dot{z}$ and $\ddot{z}$. If this is equivalent to the z_2-intercept of the line passing $\dot{z}$ and $\ddot{z}$, then $\ddot{\chi}$ is correct and returned by the procedure; otherwise, $\ddot{\chi}$ is updated to the optimal solution of the problem given in Line (5). For example, consider finding the next-lower extreme point with respect to z^1 in Fig. 4.2. Point z^5 is found in Line 2. The α is set to the slope of the line passing z^1 and z^5, hence $\alpha = -\frac{1}{2}$. The LP in Line 5 finds the lowest line that contains feasible point. Then, z^3 is found and $\tilde{\beta} = 4$. Since $\tilde{\beta} < z_2^5 + .5z_1^5 = 5.5$, the *repeat* is executed again while eliminating z^5 and using z^3. In the next iteration, z^2 is found. In the 3^{rd} iteration of *repeat*, the LP problem in line 5 has multiple optimal solution in $[z^1, z^2]$. For any optimal solution, $\tilde{\beta} = 2.5$ that equals $z_2^2 - \frac{1}{6}z_1^2$, hence the loop is broken and z^2 is returned as the next-lower extreme point with respect to z^1.

THEOREM 7. *Suppose that $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ is a feasible solution of problem P and the next-lower extreme point, $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$, with respect to $\dot{\chi}$ exists. The procedure*

Algorithm 3 Find the next-lower extreme point, $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$, with respect to $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$

1: **procedure** FindNep($\dot{\chi}$)
2: $\quad \tilde{\chi} \Leftarrow \min z_1$ s.t. $z \in \Theta(\bar{y})$
3: $\quad$ **repeat**
4: $\quad\quad$ Set $\ddot{\chi} \leftarrow \tilde{\chi}$, $\alpha \leftarrow \frac{\dot{z}_2 - \ddot{z}_2}{\dot{z}_1 - \ddot{z}_1}$
5: $\quad\quad \tilde{\chi}, \tilde{\beta} \Leftarrow \min \beta$ s.t. $z_2 = \alpha z_1 + \beta, z \in \Theta(\bar{y})$
6: $\quad$ **until** $\tilde{\beta} < \ddot{z}_2 - \alpha \ddot{z}_1$
7: $\quad$ **return** $\ddot{\chi}$
8: **end procedure**

FindNep *finds* $\ddot{\chi}$ *in at most* $\#EP\Theta(\bar{y})$ *iterations of the* repeat*. (*$\#EP\Theta(\bar{y})$*: number of extreme points of* $\Theta(\bar{y})$*).)*

PROOF OF THEOREM 7. We first propose three Lemmas that are used to prove Theorem 7.

LEMMA 1. *Suppose that* $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ *is a feasible solution of problem P and the next-lower extreme point,* $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$*, with respect to* $\dot{\chi}$ *exists. If Algorithm 3 is used to find* $\ddot{\chi}$*, when Line (5) is reached,* $\ddot{z} \in \partial\Theta(\bar{y})$ *and* $\ddot{z}_1 < \dot{z}_1$*.*

PROOF OF LEMMA 1. We use mathematical induction. Note that the existence of $\ddot{\chi}$ is examined before invoking FindNep; hence, based on theorem 4, $\tilde{\chi}$ that is found by the LP in Line (2) is such that $\tilde{z} \in \partial\Theta(\bar{y})$ and $\tilde{z}_1 < \dot{z}_1$; then, in the first iteration, the assumptions are satisfied.

Suppose that FindNep is in a later iteration of the *repeat* loop, $\ddot{\chi}$ is s.t. $\ddot{z} \in \partial\Theta(\bar{y})$ and $\ddot{z}_1 < \dot{z}_1$, and $\tilde{\chi}$ and $\tilde{\beta}$ are s.t. $\tilde{\beta} < \ddot{z}_2 - \alpha\ddot{z}_1$. Note that if $\tilde{\beta} = \ddot{z}_2 - \alpha\ddot{z}_1$, then the loop breaks. Hence, $\tilde{z}$ is on a line that is parallel to the line that contains $\dot{z}$ and $\ddot{z}$ in objective space. Suppose to the contrary that $\tilde{z}_1 \geq \dot{z}_1$. In this case, since $\tilde{z}, \ddot{z} \in \Theta(\bar{y})$ and $\Theta(\bar{y})$ is convex, then $[\tilde{z}, \ddot{z}] \subset \Theta(\bar{y})$; however, $[\tilde{z}, \ddot{z}] \cap \dot{z}^{\leqq} \neq \emptyset$ and $\dot{z}^{\leqq} \cap \Theta(\bar{y}) = \emptyset$ (contradiction). Hence, $\tilde{z}_1 < \dot{z}_1$. $\square$

LEMMA 2. *Let* $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ *and* $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$ *be two feasible solutions of* *problem P s.t. $\dot{z}, \ddot{z} \in \partial\Theta(\bar{y})$ and $\ddot{z}_1 < \dot{z}_1$. Solve $\min \beta$ s.t. $z_2 = \alpha z_1 + \beta, z \in \Theta(\bar{y})$, where $\alpha = \frac{\dot{z}_2 - \ddot{z}_2}{\dot{z}_1 - \ddot{z}_1}$, and denote the solution by $\tilde{\beta}$ and $\tilde{\chi} = (\tilde{x}, \bar{y}, \tilde{z})$. Exactly one of the following cases happens:*

1. *$\tilde{\beta} = \ddot{z}_2 - \alpha \ddot{z}_1$. In this case $\dot{z}$ and $\ddot{z}$ are at one face of $\Theta(\bar{y})$, or*

2. *$\tilde{\beta} < \ddot{z}_2 - \alpha \ddot{z}_1$. Either $\tilde{z}$ is an extreme point of $\Theta(\bar{y})$ s.t. $\tilde{z} \neq \ddot{z}$, or $\tilde{z}$ is not an extreme point of $\Theta(\bar{y})$ but there exists an extreme point $\check{z}$ of $\Theta(\bar{y})$ s.t. $\ddot{z}_1 < \check{z}_1 < \tilde{z}_1$.*

PROOF OF LEMMA 2. Note that since $\ddot{z}_1 < \dot{z}_1$, then $\alpha \in \mathbb{R}$. Observe that $\dot{z}$ and $\ddot{z}$ are feasible solution for this LP, then $\tilde{\beta}$ cannot be greater than the objective values of these solutions, i.e. $\ddot{z}_2 - \alpha \ddot{z}_1$ or $\dot{z}_2 - \alpha \dot{z}_1$. Therefore, $\tilde{\beta} \leq \ddot{z}_2 - \alpha \ddot{z}_1$.

1. Suppose that $\tilde{\beta} = \ddot{z}_2 - \alpha \ddot{z}_1 = \dot{z}_2 - \alpha \dot{z}_1$. In this case, $\dot{z}$ and $\ddot{z}$ are two distinct points and optimal for the LP, then $[\dot{z}, \ddot{z}]$ is optimal. Note that any point $\check{z} \in [\dot{z}, \ddot{z}]$ should be a boundary point of $\Theta(\bar{y})$, since if $\check{z}$ is in interior of $\Theta(\bar{y})$, there exists a strictly better solution than $\check{z}$ when going in the opposite direction of the nonzero objective vector. Therefore, $\dot{z}$ and $\ddot{z}$ are at one face of $\Theta(\bar{y})$.

2. Suppose that $\tilde{\beta} < \ddot{z}_2 - \alpha \ddot{z}_1$. Note that $\tilde{z}$ is in boundary of $\Theta(\bar{y})$, between $\dot{z}$ and $\ddot{z}$. Also note that $\tilde{z}$ is not equivalent to $\dot{z}$ and $\ddot{z}$. In this case, $\tilde{z}$ is either

   (a) an extreme point of $\Theta(\bar{y})$, or

   (b) not an extreme point. Note that if $\tilde{z}$ and $\ddot{z}$ are at one face, then $\ddot{z}$ should also be optimal (contradiction); hence, $\tilde{z}$ and $\ddot{z}$ are at different faces of $\Theta(\bar{y})$, meaning that there exists an extreme point between them.

□

LEMMA 3. *Suppose that $\dot{\chi} = (\dot{x}, \bar{y}, \dot{z})$ is a feasible solution of problem P and the next-lower extreme point, $\ddot{\chi} = (\ddot{x}, \bar{y}, \ddot{z})$, with respect to $\dot{\chi}$ exists. If Algorithm 3 is used, when $\tilde{\beta} = \ddot{z}_2 - \alpha \ddot{z}_1$, then $\ddot{\chi}$ is the next-lower extreme point with respect to $\dot{\chi}$.*

PROOF OF LEMMA 3. Suppose that at an iteration of the *repeat*, $\tilde{\beta} = \ddot{z}_2 - \alpha \ddot{z}_1$. We denote $\tilde{z}$ and $\ddot{z}$ in the previous iteration by $\tilde{z}_{old}$ and $\ddot{z}_{old}$, and in this iteration by $\tilde{z}_{new}$ and $\ddot{z}_{new}$. Based on Lemma 2, $\dot{z}$ and $\ddot{z}_{new}$ are at one face. Suppose to the contrary that $\ddot{z}_{new}$ is not extreme point of $\Theta(\bar{y})$. Then, consider the previous iteration. Any point at the same face with $\ddot{z}_{new} = \tilde{z}_{old}$ should have been optimal for the LP. Then, $\dot{z}$ was optimal for the LP in the previous iteration. Since $\dot{z}$ and $\ddot{z}_{old}$ had same objective value, then $\ddot{z}_{old}$ was optimal. It follows that $[\dot{z}, \ddot{z}_{old}] \in \partial\Theta(\bar{y})$ and was optimal. Hence, the *repeat* should have broken at the end of previous iteration (contradiction). $\square$

Now, we prove theorem 7 based on Lemmas 1-3. Using lemma 1, the assumptions of Lemma 2, i.e. $\dot{z}, \ddot{z} \in \partial\Theta(\bar{y})$ and $\ddot{z}_1 < \dot{z}_1$, are always satisfied when FindNep reaches Line (5). Moreover, based on lemma 3, when *repeat* breaks, the next-lower extreme point with respect to $\dot{\chi}$ is found.

Note that based on Lemma 2, at the end of each iteration of *repeat*, either $\ddot{\chi}$ is found and the loop is broken, or an extreme point of $\Theta(\bar{y})$ is passed. Therefore, *repeat* is not executed more than $\#EP\Theta(\bar{y})$. $\square$

The main steps of EnpoBomip is given in Algorithm 4. Recall that our algorithm is always at a boundary point in objective space and moves from right to left, i.e. large to small z_1-value. The current point is denoted as $\chi^{\text{CP}} = (x^{\text{CP}}, y^{\text{CP}}, z^{\text{CP}})$. At the beginning of the algorithm, χ^{CP} is equivalent to χ^{SE} and Φ^* is a singleton containing χ^{SE}. As EnpoBomip moves from right to left, if it is known that for some $\bar{y} \in Y$ the algorithm will not find any nondominated point in $\Theta(\bar{y})$ in the left-side of the current point, $\bar{y}$ is permanently excluded from Y in the remaining of the algorithm. Let Y^{EXC} denote the set of permanently excluded y's. We define a Boolean scalar ξ that is "true" if the current solution is efficient. The *loop* given in Lines (2)-(39) explores the current point and moves to the left as long as reaches a new current point. At any current point χ^{CP}, there are two possibilities (theorem 4) that is distinguished by the *if* in Line (3):

1. $\Theta(y^{\text{CP}})$ has empty intersection with $z_1^{\text{CP}<}$. In this case, Lines (4)-(12) are performed. y^{CP} is added to Y^{EXC} since $\Theta(y^{\text{CP}})$ will not provide nondominated points in the future. Then, the possibility of finding a new point χ^{NP} in $z_1^{\text{CP}\leq}$ is checked in Line (5). If it is possible, χ^{NP} is found (Line (6)), otherwise, the algorithm stops and Φ^* is returned (Line (11)). After finding χ^{NP}, it is checked in Lines (7)-(10) to know if it is a nondominated solution. If z^{NP} is not dominated by z^{CP}, we add χ^{NP} to Φ^* as a new efficient solution and χ^{CP} is updated, otherwise, χ^{CP} is updated and recorded as dominated solution (Line 9).

2. The next-lower extreme point with respect to χ^{CP} exists. In this case, Lines (14)-(37) are performed. The next-lower extreme point is found using FindNep. We define $\tilde{Y}^{\text{EXC}}$ as the set of temporarily excluded y's. The set $\tilde{Y}^{\text{EXC}}$ is used to break ties that happen when $\exists\bar{\chi}$ s.t. $\bar{y} \neq y^{\text{CP}}$ and $\bar{z} = z^{\text{CP}}$. We also define $\tilde{\Phi}$ to break ties that happen if there are overlapping line segments with (z^{CP},z^{NEP}). The *loop* in Lines (15)-(37) is executed to explore all solutions and find the ones with the lowest next-lower extreme point. The possibility of finding a new solution $\check{\chi}$ s.t. $\check{z} \in [z^{\text{CP}}, z^{\text{NEP}}]^{\leqq}$ is examined in Line (16) and found, if possible, in Line (17); otherwise, based on Theorem 2, $(\chi^{\text{CP}}, \chi^{\text{NEP}}]$ with all overlapping line segments, $\tilde{\Phi}$, are added to Φ^* (Line (35)) and χ^{CP} is updated to χ^{NEP} and recorded as an efficient solution by $\xi =$TRUE. In the case of finding $\check{\chi}$, if $\check{z}_1 < z_1^{\text{CP}}$, then the results of Theorem 3 applies; otherwise, i.e. $\check{z}_1 = z_1^{\text{CP}}$, tie happens. In this case, if current point is efficient, $\check{\chi}$ is also added to Φ^* as an efficient solution with alternative y-value. The tie is broken in Lines (23)-(32): If $\Theta(\check{y})$ has empty

intersection with $z_1^{\text{CP}<}$, it is excluded (Line (31)) without any need for other actions, otherwise, the next-lower extreme point $\tilde{\chi}$ with respect to $\check{\chi}$ is found, the slope of the line passing z^{CP} and z^{NEP} is compared to the slope of the line containing z^{CP} and $\tilde{z}$ in Lines (25)-(30):

(a) If the line passing z^{CP} and $\tilde{z}$ is lower, y^{CP} is temporarily excluded, χ^{CP} and χ^{NEP} are updated to $\check{\chi}$ and $\tilde{\chi}$, and $\tilde{\Phi}$ is emptied since the overlapping line segments do not overlap the line passing the updated z^{CP} and z^{NEP}.

(b) If the line passing z^{CP} and z^{NEP} is lower, $\check{y}$ is temporarily excluded.

(c) If the two lines have the same slope:

  i. If $(z^{\text{CP}}, z^{\text{NEP}}) \subseteq (\check{z}, \tilde{z})$, $\check{y}$ is temporarily excluded, but $(\check{\chi}, \bar{\tilde{\chi}})$ is added to $\tilde{\Phi}$ as alternative nondominated line segment, where $\bar{\tilde{\chi}} = (\bar{\tilde{x}}, \check{y}, z^{\text{NEP}})$ and $\bar{\tilde{x}}$ is calculated based on ratios between z-values.

  ii. If $(z^{\text{CP}}, z^{\text{NEP}}) \supset (\check{z}, \tilde{z})$, y^{CP} is temporarily excluded, $\tilde{\Phi}$ is updated and $(\chi^{\text{CP}}, \bar{\chi}^{\text{NEP}})$ is added to $\tilde{\Phi}$. Also χ^{CP} and χ^{NEP} are updated to $\check{\chi}$ and $\tilde{\chi}$. Note that updating $\tilde{\Phi}$ is eliminating the additional part of the line segments in $\tilde{\Phi}$ since all of the line segments in $\tilde{\Phi}$ should have the same length with $(z^{\text{CP}}, z^{\text{NEP}})$ in objective space.

We illustrate Algorithm 4 on the example given in Fig. 3.1.

Line 1: $\chi^{\text{CP}} = (-, y^1, z^1), \Phi^* = \{(-, y^1, z^1)\}, Y^{\text{EXC}} = \{\}, \xi = \text{TRUE}$

Line 3: $\phi = 14 = z_1^1$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^2, z^3)$

Line 9: $\chi^{\text{CP}} = (-, y^2, z^3)$, $\xi =$FALSE

Line 3: $\phi = 14 = z_1^3$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^3, z^5)$

Line 9: $\chi^{\text{CP}} = (-, y^3, z^5)$, $\xi =$FALSE

Line 3: $\phi = 13 < z_1^5$, then lines 14-37 are performed

Line 14: $\chi^{\text{NEP}} = (-, y^3, z^6)$, $\tilde{Y}^{\text{EXC}} = \{\}$, $\tilde{\Phi} = \{\}$

Line 16: $\text{P}_{\text{LX1}}^{\text{NCC}}$ is not feasible

Line 35: $\Phi^* = \{\cdots, ((-, y^3, z^5), (-, y^3, z^6)]\}$,$\chi^{\text{CP}} = (-, y^3, z^6)$, $\xi =$ TRUE

Line 3: $\phi = 13 = z_1^6$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2, y^3\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^4, z^7)$

Line 8: $\Phi^* = \{\cdots, (-, y^4, z^7)\}$, $\xi =$ TRUE, $\chi^{\text{CP}} = (-, y^4, z^7)$

Line 3: $\phi = 8 < z_1^7$, then lines 14-37 are performed

Line 14: $\chi^{\text{NEP}} = (-, y^4, z^{11})$, $\tilde{Y}^{\text{EXC}} = \{\}$, $\tilde{\Phi} = \{\}$

Line 16-17: $\check{\chi} = (-, y^5, z^9)$

Line 19: $\Phi^* = \{\cdots, ((-, y^4, z^7), (-, y^4, z^8)), (-, y^5, z^9)\}$,$\chi^{\text{CP}} = (-, y^5, z^9)$, $\xi =$ TRUE

Line 3: $\phi = 10 = z_1^9$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2, y^3, y^5\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^4, z^{10})$

Line 9: $\chi^{\text{CP}} = (-, y^4, z^{10})$, $\xi =$FALSE

Line 3: $\phi = 8 < z_1^{10}$, then lines 14-37 are performed

Line 14: $\chi^{\text{NEP}} = (-, y^4, z^{11})$, $\tilde{Y}^{\text{EXC}} = \{\}$, $\tilde{\Phi} = \{\}$

Line 16: $\text{P}_{\text{LX1}}^{\text{NCC}}$ is not feasible

Line 35: $\Phi^* = \{\cdots, ((-, y^4, z^{10}), (-, y^4, z^{11})]\}$,$\chi^{\text{CP}} = (-, y^4, z^{11})$, $\xi =$ TRUE

Line 3: $\phi = 8 = z_1^{11}$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2, y^3, y^4, y^5\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^7, z^{12})$

Line 8: $\Phi^* = \{\cdots, (-, y^7, z^{12})\}$, $\xi =$ TRUE, $\chi^{\text{CP}} = (-, y^7, z^{12})$

Line 3: $\phi = 6 = z_1^{12}$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2, y^3, y^4, y^5, y^7\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^8, z^{13})$

Line 9: $\chi^{\text{CP}} = (-, y^8, z^{13})$, $\xi =$FALSE

Line 3: $\phi = 6 = z_1^{13}$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2, y^3, y^4, y^5, y^7, y^8\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^{10}, z^{14})$

Line 8: $\Phi^* = \{\cdots, (-, y^{10}, z^{14})\}$, $\xi = \text{TRUE}$, $\chi^{\text{CP}} = (-, y^{10}, z^{14})$

Line 3: $\phi = 3 < z_1^{14}$, then lines 14-37 are performed

Line 14: $\chi^{\text{NEP}} = (-, y^{10}, z^{19})$, $\tilde{Y}^{\text{EXC}} = \{\}$, $\tilde{\Phi} = \{\}$

Line 16-17: $\check{\chi} = (-, y^9, z^{14})$

Line 21: $\Phi^* = \{\cdots, (-, y^9, z^{14})\}$

Line 24: $\tilde{\chi} = (-, y^9, z^{21})$

Line 27: $\tilde{Y}^{\text{EXC}} = \{y^9\}$

Line 16-17: $\check{\chi} = (-, y^{12}, z^{14})$

Line 21: $\Phi^* = \{\cdots, (-, y^{12}, z^{14})\}$

Line 24: $\tilde{\chi} = (-, y^{12}, z^{18})$

Line 26: $\chi^{\text{NEP}} = (-, y^{12}, z^{18})$, $\tilde{Y}^{\text{EXC}} = \{y^9, y^{10}\}$, $\chi^{\text{CP}} = (-, y^{12}, z^{14})$

Line 16-17: $\check{\chi} = (-, y^{11}, z^{15})$

Line 19: $\Phi^* = \{\cdots, ((-, y^{12}, z^{14}), (-, y^{12}, z^{15})), (-, y^{11}, z^{15})\}$,$\chi^{\text{CP}} = (-, y^{11}, z^{15})$, $\xi = \text{TRUE}$

Line 3: $\phi = 3 < z_1^{15}$, then lines 14-37 are performed

Line 14: $\chi^{\text{NEP}} = (-, y^{11}, z^{16})$, $\tilde{Y}^{\text{EXC}} = \{\}$, $\tilde{\Phi} = \{\}$

Line 16-17: $\check{\chi} = (-, y^{12}, z^{15})$

Line 21: $\Phi^* = \{\cdots, (-, y^{12}, z^{15})\}$

Line 24: $\tilde{\chi} = (-, y^{12}, z^{18})$

Line 28: $\tilde{Y}^{\text{EXC}} = \{y^{12}\}$, $\tilde{\Phi} = \{((-, y^{12}, z^{15}), (-, y^{12}, z^{16}))\}$

Line 16: $\text{P}^{\text{NCC}}_{\text{LX1}}$ is not feasible

Line 35: $\Phi^* = \{\cdots, ((-, y^{11}, z^{15}), (-, y^{11}, z^{16})], ((-, y^{12}, z^{15}), (-, y^{12}, z^{16}))\}$, $\chi^{\text{CP}} = (-, y^{11}, z^{16})$, $\xi = \text{TRUE}$

Line 3: $\phi = 3 = z^{16}_1$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2, y^3, y^4, y^5, y^7, y^8, y^{11}\}$

Line 5-6: $\chi^{\text{NP}} = (-, y^{12}, z^{16})$

Line 8: $\Phi^* = \{\cdots, (-, y^{12}, z^{16})\}$, $\xi = \text{TRUE}$, $\chi^{\text{CP}} = (-, y^{12}, z^{16})$

Line 3: $\phi = 1 < z^{16}_1$, then lines 14-37 are performed

Line 14: $\chi^{\text{NEP}} = (-, y^{12}, z^{18})$, $\tilde{Y}^{\text{EXC}} = \{\}$, $\tilde{\Phi} = \{\}$

Line 16: $\text{P}^{\text{NCC}}_{\text{LX1}}$ is not feasible

Line 35: $\Phi^* = \{\cdots, ((-, y^{12}, z^{16}), (-, y^{12}, z^{18})]\}, \chi^{\text{CP}} = (-, y^{12}, z^{18})$, $\xi = \text{TRUE}$

Line 3: $\phi = 1 = z^{18}_1$, then lines 4-12 are performed

Line 4: $Y^{\text{EXC}} = \{y^1, y^2, y^3, y^4, y^5, y^7, y^8, y^{11}, y^{12}\}$

Line 5: $\text{P}^{\text{NP}}_{\text{LX1}}$ is infeasible

Line 11: Return Φ^*

THEOREM 8. *Given problem P,* ENPOBOMIP *finds* Φ^* *in finite iterations.*

PROOF OF THEOREM 8. The proof consists of two steps:

1. Algorithm 4 finds Φ^*. The proof of this part is referred to Theorems 1-7.

2. The number of iterations for both *loop*'s are finite. Note that the second loop given in Lines (15)-(37) is for breaking ties. For now, assume that ties do not happen, hence we can remove Lines (15)-(37) from Algorithm 4 and use Lines (16, 17, 19, 34, 35, and 36). In this case, only one *loop* remains and there are three possibilities in each iteration:

   (a) y^{CP} is excluded. Note that the number of y's is bounded by $|Y|$.

   (b) The algorithm proceeds to the next-lower extreme point in $\Theta(y^{\text{CP}})$. Note that the number of extreme points of $\Theta(y^{\text{CP}})$ is finite.

   (c) The algorithm moves to another boundary point $\check{z} \in \partial\Theta(\check{y})$. Two cases are possible:

      i. The algorithm returns to the boundary of $\Theta(y^{\text{CP}})$ before the next-lower extreme point with respect to χ^{CP}. In this case, since $\Theta(y^{\text{CP}})$ and $\Theta(\check{y})$ are convex sets, the algorithm will not return back to boundary of $\Theta(\check{y})$ by the time it reaches the next-lower extreme point with respect to χ^{CP}. Note that the number of y's is bounded by $|Y|$. Hence, the next-lower extreme point with respect to χ^{CP} is passed after a finite number of iterations.

ii. The algorithm do not return to the boundary of $\Theta(y^{\text{CP}})$ by the time it reaches the next-lower extreme point with respect to χ^{CP}. Hence, it can be assumed that the next-lower extreme point with respect to χ^{CP} is passed.

Now, we analyze the *loop* given in Lines (15)-(37) and show that this *loop* is broken after a finite number of iterations. Note that in each iteration of this *loop*, either a y-value is excluded or the *loop* is broken. Also note that the number of y's is bounded by $|Y|$. Hence, this *loop* is broken after a finite number of iterations.

$\square$

Thus, EnpoBomip finds the exact nondominated frontier with all possible y-values by finite number of MILP resolutions. As the nondominated frontier of BOMILPs usually include line segments, the number of nondominated points is infinity; hence, $\varphi(\text{EnpoBomip}) = \frac{\#\text{NDP}}{\#\text{MILP}} = \frac{\infty}{\#\text{MILP}} = \infty$. Therefore, considering the efficiencies of Augmecon ($\approx$1) and Can ($\approx$0.5), EnpoBomip is very efficient than these algorithms. Also note that Augmecon and Can do not find all alternative y-values associated with each nondominated point.

Algorithm 4 $\Phi^* \Leftarrow$ ENPOBOMIP(P)

$\chi^{\text{CP}} \Leftarrow \text{LX}(\text{P}^{\text{SE}}_{\text{LX1}}, \text{P}^{\text{SE}}_{\text{LX2}})$, Set $\Phi^* \leftarrow \{\chi^{\text{CP}}\}$, $Y^{\text{EXC}} \leftarrow \{\}$, $\xi \leftarrow$ TRUE
loop
  if $\phi\,(\Leftarrow \text{LP}^{\text{NEP}}(y^{\text{CP}})) = z_1^{\text{CP}}$ **then**
    Set $Y^{\text{EXC}} \leftarrow Y^{\text{EXC}} \cup \{y^{\text{CP}}\}$
    if $z_2^{\text{NP}} \Leftarrow \text{P}^{\text{NP}}_{\text{LX1}}(Y^{\text{EXC}}, z_1^{\text{CP}})$ is feasible **then**
      $\chi^{\text{NP}} \Leftarrow \text{P}^{\text{NP}}_{\text{LX2}}(z_2^{\text{NP}})$
      if $z_1^{\text{NP}} < z_1^{\text{CP}}$ **or** $z^{\text{NP}} = z^{\text{CP}}$ **then**
        Set $\Phi^* \leftarrow \Phi^* \cup \{\chi^{\text{NP}}\}$, $\xi \leftarrow$ TRUE , $\chi^{\text{CP}} \leftarrow \chi^{\text{NP}}$
      else Set $\chi^{\text{CP}} \leftarrow \chi^{\text{NP}}$, $\xi \leftarrow$ FALSE
      end if
    else RETURN Φ^*
    end if
  else
    $\chi^{\text{NEP}} \Leftarrow$ FINDNEP(χ^{CP}), Set $\tilde{Y}^{\text{EXC}} \leftarrow \{\}$, $\tilde{\Phi} \leftarrow \{\}$
    loop
      if $\check{z}_2 \Leftarrow \text{P}^{\text{NCC}}_{\text{LX1}}(Y^{\text{EXC}} \cup \tilde{Y}^{\text{EXC}} \cup y^{\text{CP}}, z^{\text{CP}}, z^{\text{NEP}})$ is feasible **then**
        $\check{\chi} \Leftarrow \text{P}^{\text{NCC}}_{\text{LX2}}(Y^{\text{EXC}} \cup \tilde{Y}^{\text{EXC}} \cup y^{\text{CP}}, \check{z}_2)$
        if $\check{z}_1 < z_1^{\text{CP}}$ **then**
          Update $\tilde{\Phi}$, Set $\Phi^* \leftarrow \Phi^* \cup (\chi^{\text{CP}}, \bar{\chi}) \cup \{\check{\chi}\} \cup \tilde{\Phi}$, $\chi^{\text{CP}} \leftarrow \check{\chi}$, $\xi \leftarrow$ TRUE, **break loop**
        else
          if $\xi =$ TRUE **then** Set $\Phi^* \leftarrow \Phi^* \cup \{\check{\chi}\}$
          end if
          if $\phi\,(\Leftarrow \text{LP}^{\text{NEP}}(\check{y})) < z_1^{\text{CP}}$ **then**
            $\tilde{\chi} \Leftarrow$ FINDNEP$(\check{\chi})$
            if $\frac{\tilde{z}_2 - z_2^{\text{CP}}}{\tilde{z}_1 - z_1^{\text{CP}}} > \frac{z_2^{\text{NEP}} - z_2^{\text{CP}}}{z_1^{\text{NEP}} - z_1^{\text{CP}}}$ **then**
              Set $\tilde{Y}^{\text{EXC}} \leftarrow \tilde{Y}^{\text{EXC}} \cup \{y^{\text{CP}}\}$, $\chi^{\text{CP}} \leftarrow \check{\chi}$, $\chi^{\text{NEP}} \leftarrow \tilde{\chi}$, $\tilde{\Phi} \leftarrow \{\}$
            else if $\frac{\tilde{z}_2 - z_2^{\text{CP}}}{\tilde{z}_1 - z_1^{\text{CP}}} < \frac{z_2^{\text{NEP}} - z_2^{\text{CP}}}{z_1^{\text{NEP}} - z_1^{\text{CP}}}$ **then** Set $\tilde{Y}^{\text{EXC}} \leftarrow \tilde{Y}^{\text{EXC}} \cup \{\check{y}\}$
            else if $z_1^{\text{NEP}} \geq \tilde{z}_1$ **then** Set $\tilde{Y}^{\text{EXC}} \leftarrow \tilde{Y}^{\text{EXC}} \cup \{\check{y}\}$, $\tilde{\Phi} \leftarrow \tilde{\Phi} \cup (\check{\chi}, \bar{\tilde{\chi}})$
            else Set $\tilde{Y}^{\text{EXC}} \leftarrow \tilde{Y}^{\text{EXC}} \cup \{y^{\text{CP}}\}$, update $\tilde{\Phi}$, Set $\tilde{\Phi} \leftarrow \tilde{\Phi} \cup (\chi^{\text{CP}}, \bar{\chi}^{\text{NEP}})$, $\chi^{\text{CP}} \leftarrow \check{\chi}$, $\chi^{\text{NEP}} \leftarrow \tilde{\chi}$,
            end if
          else Set $Y^{\text{EXC}} \leftarrow Y^{\text{EXC}} \cup \{\check{y}\}$
          end if
        end if
      else
        Set $\Phi^* \leftarrow \Phi^* \cup (\chi^{\text{CP}}, \chi^{\text{NEP}}] \cup \tilde{\Phi}$, $\chi^{\text{CP}} \leftarrow \chi^{\text{NEP}}$, $\xi \leftarrow$ TRUE, **break loop**
      end if
    end loop
  end if
end loop

Chapter 5

ϵ-OA

In this chapter, we present our effective algorithm, ϵ-OA, to solve biobjective mixed-integer nonlinear programming problems for the synthesis of process networks that are formulated using generalized disjunctive programming. First, the augmented ϵ-constraint model of this problem is given, then the foundation of logic-based OA with respect to the biobjective nonlinear process networks problem is outlined. The novel ϵ–OA algorithm in presented afterwards.

5.1 Augmented ϵ-constraint

The augmented ϵ-constraint method converts MOOP to a single objective problem considering one criterion, usually the most important one, as the objective function and including the others into the set of constraints [Mavrotas, 2009]. The objective function also incorporates the slack variables for the newly added constraints to eliminate the possibility of generating dominated solutions. In this section, we present the augmented ϵ-constraint model for the GDP of bi-objective nonlinear process networks and prove the efficiency of the obtained solutions. Note that this proof holds if the nonlinear functions (i.e. $f_1(x)$, $f_2(x)$, $g(x)$, and $h_i(x)$) are convex.

In the augmented ϵ-constraint method, the range of the second objective (i.e. $z_2^U - z_2^L$) is divided into Q equivalent intervals, and the first objective is optimized for each. If the value of Q is sufficiently large, solving the subproblems of the augmented ϵ-constraint method results in a fine approximation of the efficient frontier. Note that, however, more computational effort is required as Q increases. The j^{th} subproblem

(SP_j) of the augmented ϵ-constraint method is written as follows ($j = 0, \cdots, Q$).

$$\min \quad z^{aug} = \sum_i c_i + f_1(x) - \mu \frac{s}{r} \tag{5.1}$$

$$\text{s.t.} \quad \sum_i \rho_i + f_2(x) + s = z_2^U - j\epsilon, \tag{5.2}$$

$$\text{and Eqs. (3.5)-(3.8).} \tag{5.3}$$

where μ is an adequately small number, usually between 10^{-3} and 10^{-6}, s is the slack of the constraint that is added to the problem because of the second objective, z_2^L and z_2^U are the lower and upper bounds of the second objective, r is the range of the second objective, and ϵ is the length of each interval (i.e. $\epsilon = \frac{z_2^U - z_2^L}{Q}$).

THEOREM 9. *Solving subproblem SP_j produces efficient solutions for problem P^{GDP}.*

PROOF OF THEOREM 9. First, we prove that solving SP_j produces a solution that is efficient over the feasible region of the subproblem. Suppose that $\mathcal{X}_A = (x^A, c^A, \rho^A, \mathcal{Y}^A)$ and $\mathcal{X}_B = (x^B, c^B, \rho^B, \mathcal{Y}^B)$ are feasible for SP_j and $\mathcal{X}_A$ dominates $\mathcal{X}_B$.

$$z_A^{aug} = \sum_i c_i^A + f_1(x^A) - \frac{\mu}{r}\left(z_2^U - j\epsilon - \sum_i \rho_i^A - f_2(x^A)\right) \tag{5.4}$$

$$z_B^{aug} = \sum_i c_i^B + f_1(x^B) - \frac{\mu}{r}\left(z_2^U - j\epsilon - \sum_i \rho_i^B - f_2(x^B)\right) \tag{5.5}$$

Then,

$$\begin{aligned} z_A^{aug} - z_B^{aug} &= \left(\sum_i c_i^A + f_1(x^A) - \sum_i c_i^B - f_1(x^B)\right) \\ &+ \frac{\mu}{r}\left(\sum_i \rho_i^A + f_2(x^A) - \sum_i \rho_i^B - f_2(x^B)\right) \\ &= z_1^A - z_1^B + \frac{\mu}{r}(z_2^A - z_2^B) \end{aligned} \tag{5.6}$$

There are two possibilities:

- $z_1^A \leq z_1^B$ and $z_2^A < z_2^B$. Consider Eq. (5.6). Since $z_1^A - z_1^B \leq 0$, $\mu > 0$, $r > 0$, and $z_2^A - z_2^B < 0$, then the right-hand-side of Eq. (5.6) is negative. Then, $z_A^{aug} - z_B^{aug} < 0$ and $z_A^{aug} < z_B^{aug}$. Therefore, only $\mathcal{X}_A$ can be found due to a better objective value.

- $z_1^A < z_1^B$ and $z_2^A = z_2^B$. In this case, Eq. (5.6) becomes $z_A^{aug} - z_B^{aug} = z_1^A - z_1^B < 0$. Therefore, only $\mathcal{X}_A$ can be found.

Now, we prove that the efficient solution $\mathcal{X}_A$ cannot be dominated by the feasible solutions of problem P^{GDP} that are outside of the feasible region of the subproblem SP_j. Since $\mathcal{X}_A$ is feasible for subproblem SP_j, then $\mathcal{X}_A$ is feasible for Eqs. (3.5)-(3.8) and $\sum_i \rho_i^A + f_2(x^A) \leq z_2^U - j\epsilon$. In the above, we proved that $\mathcal{X}_A$ cannot be dominated by feasible solutions of the same subproblem. So, suppose that $\mathcal{X}_C$ is a feasible solution of problem P^{GDP}, but is not feasible for subproblem SP_j. Therefore, $\mathcal{X}_C$ is feasible for Eqs. (3.5)-(3.8), but it does not satisfy $\sum_i p_i^C + f_2(x^C) \leq z_2^U - j\epsilon$. Suppose that, on the contrary, $\mathcal{X}_C$ dominates $\mathcal{X}_A$, thus $\sum_i \rho_i^C + f_2(x^C) \leq \sum_i \rho_i^A + f_2(x^A)$. Since $\sum_i \rho_i^A + f_2(x^A) \leq z_2^U - j\epsilon$, then $\sum_i \rho_i^C + f_2(x^C) \leq \sum_i \rho_i^A + f_2(x^A) \leq z_2^U - j\epsilon$. It follows that $\sum_i \rho_i^C + f_2(x^C) \leq z_2^U - j\epsilon$, meaning that $\mathcal{X}_C$ is feasible for subproblem SP_j. This is a contradiction. Hence, any efficient solution of subproblem SP_j is efficient for problem P^{GDP}. $\square$

The efficient frontier that can be approximated by solving subproblems SP_j's for $j = 1, \cdots, Q$ provides useful trade-off information between objectives and can be used by decision maker to choose from the possible efficient alternatives. The impediment, however, is solving these NP-hard subproblems, specially when Q is very large. The state-of-the-art MINLP solvers, such as DICOPT, can be used to solve the subproblems SP_j. Algorithm 5 represents this method to produce efficient solutions for the bi-objective nonlinear network synthesis problems (ϵ-MINLP). In section 6.2, we show that ϵ-MINLP using DICOPT is very slow in solving the 3 and 8 process network synthesis examples. In the remaining, we present an efficient algorithm to find the efficient solutions for very large values of Q in a short computing time.

Algorithm 5 the ϵ-MINLP for the biobjective discrete GDPs for nonlinear convex network synthesis problems

```
Input: biobjective discrete nonlinear convex problem, and Q
Output: efficient solutions
for m = 1, 2 do
    Solve single objective MINLP problem minimizing objective m
    Calculate pay-off table values using the optimal solution.
end for
for j = 0 → Q do
    Solve augmented MINLP subproblem SP_j
    Calculate efficient solution using the optimal solution.
end for
```

5.2 Foundations of logic-based OA

In this section, we present the logic-based OA [Türkay and Grossmann, 1996b] to solve the subproblems of the augmented ϵ-constraint method. The OA was first developed by [Duran and Grossmann, 1986] to solve convex MINLPs. The logic-based OA starts with solving a set covering problem to find the minimum number of alternatives that cover all of the processing units. The NLP subproblem is solved for each of these alternatives and the best upper bound is obtained. The Maser problem is constructed by linearizing the nonlinear functions, and a lower bound is found by solving the Master problem. If the optimality gap is small enough, the algorithm terminates and the solution of the NLP subproblem is optimal, otherwise, the NLP subproblem is solved using the Boolean variable values found by the Master problem. If the remaining optimality gap is within tolerance limits, the algorithm stops, otherwise the Master problem is constructed and solved. This procedure continues until the lower and upper bounds converge. The NLP subproblem of SP_j for a fixed choice of Boolean variables, $\bar{\mathcal{Y}}$, is as follows.

$$\min \quad z_U^{aug} = \sum_i c_i + f_1(x) - \frac{\mu}{r}s \tag{5.7}$$

$$\text{s.t.} \quad \sum_i \rho_i + f_2(x) + s = z_2^U - j\epsilon, \tag{5.8}$$

$$g(x) \leq 0, \tag{5.9}$$

$$\left.\begin{array}{l} h_i(x) \leq 0 \\ c_i = \gamma_i \\ \rho_i = \eta_i \end{array}\right\} \forall \bar{\mathcal{Y}}_i = True, \tag{5.10}$$

$$\left.\begin{array}{l} B^i x = 0 \\ c_i = 0 \\ \rho_i = 0 \end{array}\right\} \forall \bar{\mathcal{Y}}_i = False, \tag{5.11}$$

$$x \in \mathbb{R}^n, c \geq 0, \rho \geq 0. \tag{5.12}$$

Solving the NLP subproblem provides a feasible solution and an upper bound on the optimal value of SP_j. The solution of this problem is used as the input to construct the MILP Master problem. Suppose that L values have been obtained for x from solving the NLP subproblems by the current iteration, i.e. x^l, $l = 1, \cdots, L$.

Then, the master problem is written as follows.

$$\begin{aligned}
\min \quad & z_L^{aug} = \sum_i \gamma_i y_i + \alpha_{oa} - \frac{\mu}{r} s && (5.13)\\
\text{s.t.} \quad & \alpha_{oa} \geq f_1(x^l) + \nabla f_1(x^l)^\top (x - x^l) \quad \forall l = 1, \cdots, L && (5.14)\\
& g(x^l) + \nabla g(x^l)^\top (x - x^l) \leq 0 \quad \forall l = 1, \cdots, L && (5.15)\\
& \sum_i \eta_i y_i + f_2(x^l) + \nabla f_2(x^l)^\top (x - x^l) + s \\
& \qquad = z_2^U - j\epsilon \quad \forall l = 1, \cdots, L && (5.16)\\
& \nabla h_i(x^l)^\top x \leq \left(-h_i(x^l) + \nabla h_i(x^l)^\top x^l\right) y_i \\
& \qquad \forall l = 1, \cdots, L, i \in D && (5.17)\\
& B^i x \leq M_i y_i \quad \forall i \in D && (5.18)\\
& Ay \leq a && (5.19)\\
& \alpha_{oa} \in \mathbb{R}^1, x \in \mathbb{R}^n, y \in \{0,1\}^m. && (5.20)
\end{aligned}$$

Eq. (5.19) represents the logic relations between y_i's. Solving the MILP Master problem provides a lower bound on the optimal value of SP_j. The Master problem finds a new values for Boolean variables to generate and solve NLP subproblem.

The logic-based OA can be applied to each subproblem of the augmented ϵ-constraint method SP_j when solving a biobjective discrete nonlinear problem to find efficient solutions; however, this requires significant computational effort since many NLP subproblems and Master problems should be solved in each iteration that leads to very large number of problem resolutions in total. We refer this method as ϵ-con+OA (see Algorithm 6). In section 6.2, we show that ϵ-con+OA is very slow and requires resolution of many NLP subproblems and Master problems.

Algorithm 6 the ϵ-con+OA for the biobjective discrete GDPs for nonlinear convex network synthesis problems

1: **Input:** biobjective discrete nonlinear convex problem, and Q
2: **Output:** efficient solutions
3: **for** $m = 1, 2$ **do**
4: Use logic-based OA to solve single objective network synthesis problem minimizing objective m
5: Calculate pay-off table values using the optimal solution.
6: **end for**
7: **for** $j = 0 \to Q$ **do**
8: Use logic-based OA to solve the j^{th} subproblem of the augmented ϵ-constraint problem SP_j
9: Calculate efficient solution using the optimal solution.
10: **end for**

5.3 The ϵ–OA

In this section, we present our novel ϵ–OA algorithm that consists of two phases: calculation of the pay-off table, and iteratively solving the subproblems of the augmented ϵ-constraint method to determine efficient solutions.

5.3.1 Phase 1 of the ϵ–OA

The first phase includes the optimization of two single objective problems that gives the best possible values of each objective function; then, the worst values of each objective can be calculated using the optimal solution obtained from optimizing the other objective. The best and worst values of objectives form the pay-off table. The optimization of objective m ($m = 1, 2$) starts with assigning very small and large values to z_{LB} and z_{UB}, respectively. Solving the set covering problem determines the alternatives that should be solved at the beginning of the algorithm. For each of these alternatives, the NLP subproblem is solved that is min. z_k subject to Eqs. (3.5)-(3.8), where $\mathcal{Y}$-values are known based on the present units in the alternative. The best optimal solution is saved as $(z^{opt}, \mathcal{Y}^{opt}, x^{opt})$, and z_{UB} is updated to z^{opt}, after solving the NLP subproblem for all required alternatives.

We have a starting point $(z^{opt}, \mathcal{Y}^{opt}, x^{opt})$ to initialize the main iterations of the OA by solving the Master problems and NLP subproblems iteratively as long as the optimality gap is not satisfied, i.e. $|z_{UB} - z_{LB}| > \delta$. The Master problem can be constructed using x^{opt}. Note that, in this phase, the Master problem is constructed for the single objective problem, meaning that the objective function does not include the augmented part as in Eq. (5.13), and there is no constraint for the second objective function. Observe that the outer approximations of the nonlinear constraints remain valid for the ϵ–OA; hence, we maintain these approximations and include them when solving a Master problem. After solving the Master problem, z_{LB} is updated if the optimal value provides a better lower bound. A no-good constraint is also added to the Master problem to prevent reproducing the same $\mathcal{Y}$-values in the subsequent iterations of the OA.

The solution of the Master problem generates $\mathcal{Y}$-values that is used as input for the NLP subproblem. After solving the NLP subproblem, if it is possible, we update $(z^{opt}, \mathcal{Y}^{opt}, x^{opt})$ and z_{UB}. The x-values are used to construct and solve the Master problem. This procedure is performed repeatedly until an optimal solution is found. Note that when solving the first single objective problem finishes, before applying the same process for the 2^{nd} objective, no-good constraints and the outer approximations that are related to the first objective function are removed from the Master problem; however, the outer approximations for other constraints are still valid.

5.3.2 Phase 2 of the ϵ–OA

In the second phase of the ϵ–OA, the SP$_j$, given by Eqs. (5.1)-(5.3), is optimized for all $j = 0, \cdots, Q$. Here, z_2^U and z_2^L are the best and worst values of the second objective and are known from the pay-off table.

The iterations of the ϵ–OA starts from $j = 0$. We assign very small and large values to z_{LB} and z_{UB}, respectively. Moreover, we set z^{opt} (the optimal solution of SP_j) to a very large value. The Master problem, given by Eqs. (5.13)-(5.20), and the NLP subproblem, given by Eqs. (5.7)-(5.12) are solved until the optimality gap becomes less than δ and an optimal solution is found. The Master problem includes some of the outer approximations from phase 1 that are valid in addition to the ones that are generated from the optimal solutions of the NLP subproblems.

After solving the Master problem, we store the optimal solution $(\hat{z}, \hat{\mathcal{Y}}, \hat{x})$ to use in the further iterations. Using the optimal solution, we update z_{LB} to $\hat{z}$ if it is a better bound. Moreover, a no-good constraint is added to the Master problem to prevent generating the same $\hat{\mathcal{Y}}$ in the further OA iterations. Except for Eq. (5.16), the outer approximations of other nonlinear constraints are maintained, and are active when solving a Master problem. The reason is that Eq. (5.16) becomes invalid when j increases, hence any of its outer approximations may become invalid; however, other nonlinear constraints remain the same throughout the ϵ–OA.

The $\hat{\mathcal{Y}}$-values that is resulted from solving Master problem is used as the input for NLP subproblem given by Eqs. (5.7)-(5.12). After optimizing the NLP subproblem, the optimal solution $(\tilde{z}, \tilde{\mathcal{Y}}, \tilde{x}, j_{max})$ is stored. The j_{max} is the maximum of index j for which Eq. (5.8) is feasible for $(\tilde{\mathcal{Y}}, \tilde{x})$.

$$j_{max} = \max_{\hat{j}=0,\cdots,Q} \{\sum_i \rho_i + f_2(\tilde{x}) \leq z_2^U - \hat{j}\epsilon\} \tag{5.21}$$

Based on the result of NLP subproblem, z_{UB} is updated to $\tilde{z}$ if it provides a better upper bound. Moreover, if $z^{opt} > \tilde{z}$, we update $(z^{opt}, \mathcal{Y}^{opt}, x^{opt})$ to $(\tilde{z}, \tilde{\mathcal{Y}}, \tilde{x})$. The $\tilde{x}$ can be used as input for the Master problem in the continuum of the ϵ-OA. The linear approximations that are added in each iteration to the Master problem is checked to prevent duplicity; they should not be redundant with respect to the previous approximations.

The usage of the stored solutions is to prevent redundant resolutions of the Master problems and NLP subproblems. Before solving the Master problem, all of the stored solutions of the Master problem should be investigated to see if there is a qualified solution; hence, it can be considered as the optimal solution of the Master problem. To be qualified, a stored solution should satisfy all of the current linear approximations in the Master problem, as well as the present no-good constraints. Moreover, all of the no-good constraints that were present at the time of stored solution should be also valid. Similarly, before solving NLP subproblem, the stored solutions of the NLP subproblem should be searched for a solution that satisfies all of the requirements to be optimal for the current NLP subproblem. These requirements are having the same $\mathcal{Y}$-values as input, and $j \leq j_{max}$.

Observe that the no-good constraints and the outer approximations of Eq. (5.16) may not be valid from one iteration of ϵ-OA to the next, i.e. from j to $j+1$; hence, they are removed in the beginning of each iteration. However, the outer approximations for the other constraints of the Master problem are preserved in the execution of the algorithm.

The efficient solutions (z_1^e, z_2^e) are calculated at the end of each iteration using the best Y and x-values that are found in that iteration. Based on the value of z_2^e, it may be possible to skip some iterations of the ϵ-OA. Suppose that for $\hat{j}$, the corresponding iteration returns the efficient solution $(\hat{z}_1^e, \hat{z}_2^e)$ with the best $\hat{\mathcal{Y}}$ and $\hat{x}$-values. Assume that when $\hat{j}$ increases by 1, Eq. (5.2) is still satisfied. Moreover, observe that when going from iteration $\hat{j}$ to $\hat{j}+1$, the only change of the $\mathrm{SP}_{\hat{j}+1}$ is in Eq. (5.2). Hence, $\hat{\mathcal{Y}}$ and $\hat{x}$-values are also feasible for $\mathrm{SP}_{\hat{j}+1}$. Since the feasible region of $\mathrm{SP}_{\hat{j}+1}$ is a subset of the feasible region of $\mathrm{SP}_{\hat{j}}$, then $\hat{\mathcal{Y}}$ and $\hat{x}$-values are the optimal solutions of $\mathrm{SP}_{\hat{j}+1}$ and give the same efficient solution $(\hat{z}_1^e, \hat{z}_2^e)$. Therefore, at the end of each iteration of ϵ-OA, the $\mathcal{Y}$ and x-values that are known to be the best in that iteration, are checked to see if they satisfy Eq. (5.2) of the next iteration; if it is, the next iteration is skipped.

Algorithm 3 is the ϵ-OA to solve bi-objective nonlinear network synthesis problems. This algorithm uses logic-based OA to solve SP_j's, and also incorporates our innovative speed-up ideas to eliminate unnecessary operations. In section 6.2, we show that ϵ-OA is very effective, specially for very large values of Q, and solves the bi-objective 3 and 8 process network synthesis instances in a very short CPU time.

Algorithm 7 the ϵ-OA for the bi-objective discrete GDPs for nonlinear convex network synthesis problems

Input: biobjective discrete nonlinear convex problem, and Q
Output: efficient solutions
for $m = 1, 2$ **do**
    Eliminate no-good constraints
    Set $z_{UB} = bigM$ and $z_{LB} = -bigM$
    Solve SCP and determine w_i's (w_i is 1 if alternative i is determined by the SCP, and 0 otherwise)
    for i:$w_i = 1$ **do**
        Solve single objective NLP Subproblem minimizing objective m
        If possible, update $(z^{Opt}, \mathcal{Y}^{Opt}, x^{Opt})$
    end for
    $x^{Best} = x^{Opt}$
    while $z_{UB} - z_{LB} > \delta$ **do**
        Add approximations using x^{Best}
        Solve single objective Master problem minimizing objective $m \Rightarrow \mathcal{Y}$-values
        If Master problem is integer infeasible, break the *while* loop
        Add a no-good constraint to Master problem using Y-values
        If possible, update z_{LB}
        If $z_{UB} - z_{LB} \leq \delta$, break *while* loop
        Solve single objective NLP Subproblem minimizing objective $m \Rightarrow x^{Best}$
        If possible, update z_{UB} and $(z^{Opt}, \mathcal{Y}^{Opt}, x^{Opt})$
    end while
    Calculate pay-off table values using $(\mathcal{Y}^{Opt}, x^{Opt})$.
end for
for $j = 0 \rightarrow Q$ **do**
    Set $z_{UB} = bigM$ and $z_{LB} = -bigM$
    Eliminate no-good constraints and approximations of Eq. (5.16)
    while $z_{UB} - z_{LB} > \delta$ **do**
        for stored solutions **do**
            If the solution satisfies all linear approximations, Eq (5.16), and all no-good constraints, and if the no-good constraints at the time of stored solution exist now, then use this stored solution instead of solving Master problem
        end for
        If no stored solution is qualified, solve augmented Master problem and store the solution$\Rightarrow$ $\mathcal{Y}$-values
        If Master problem is integer infeasible, break the *while* loop
        Add no-good constraint
        If possible, update z_{LB}
        If $z_{UB} - z_{LB} \leq \delta$, break *while* loop

Algorithm 8 Algorithm 7 continued.

```
        Solve augmented NLP subproblem, and store the solution ⇒ x^Best
        If possible, update z_UB and (z^Opt, 𝒴^Opt, x^Opt)
        If linear approximations for x^Best has not been added before, add them
    end while
    Calculate efficient solution (z_1^e, z_2^e) using (𝒴^Opt, x^Opt)
    while z_2^e ≤ z_2^U − jϵ do
        j++
    end while
end for
```

Chapter 6

RESULTS

This chapter presents the results of two experimental works: (1) a comparative performance analysis of AUGMECON, CAN, and ENPOBOMIP to show the effectiveness of ENPOBOMIP in finding the exact nondominated points with all alternative values of integer variables for BOMILPs, and (2) a comparison of ϵ-con+OA, ϵ–OA, ϵ-MINLP, and T-ϵ-con on 3 and 8 process network synthesis problems to highlight the speed of ϵ–OA in finding an approximation of nondominated frontier of BOMINLPs that can be formulated as P^{GDP}. The algorithms are implemented in GAMS 24.0.2, and ILOG CPLEX 12.5 Optimizer is used with a PC Pentium IV at 3.33 GHz and with 32.0 GB of RAM.

6.1 BOMILPs

In this section, we present a comparative performance analysis of AUGMECON, CAN, and ENPOBOMIP. The parameters Q and μ in AUGMECON are set to 10,000 and 10^{-4}. For the sake of simplicity, $Q = 50$ and z-values are scaled to [0,1] in drawing figures. Three well-known problems are considered:

1. **Synthesis of Complex Production Networks (SCPN):** The benchmark example of BOMILP for the synthesis of complex production networks outlined in [E GROSSMANN et al., 1982] is solved in this thesis. The objective functions are maximizing net present value and minimizing toxicity that is calculated based on the amount of substances handled in the process networks. De-

tailed illustration of the model is available in [Grossmann and Santibanez, 1980, E GROSSMANN et al., 1982].

2. **Facility Location Problem (FLP):** In this problem, a number of potential facilities with unlimited capacities should serve a given number of customers. Instances are generated similar to [Stidsen et al., 2014] with modifications in objective functions to incorporate general cases of biobjective FLPs. Let c_i^1 and p_i^1 be coefficients of binary variables, and c_{ij}^2 and p_{ij}^2 be coefficients of continuous variables in the first and second objective functions. The parameters c_i^1 and c_{ij}^2 are generated similar to [Stidsen et al., 2014], p_{ij}^2 is generated uniformly between 0 and 10, and p_i^1 is generated similar to c_i^1 using p_{ij}^2. Seven instances of biobjective FLPs with $m = 5, 10, 15, 20, 25, 30, 35$ are generated where m denotes the number of potential facilities.

3. **Fixed Charge Network Design Problem (FCNDP):** This problem connects a set of nodes with a volume of 1 shipment from a source node to a terminal node. The approach in [Stidsen et al., 2014] is followed with modifications in objective functions to incorporate general cases of biobjective FCNDPs. We let c_{ij}^1 and c_{ij}^2 be coefficients of binary and continuous variables in the first objective function, and p_{ij}^1 and p_{ij}^2 be coefficients of binary and continuous variables in the second objective function. The parameter c_{ij}^2 is generated similar to [Stidsen et al., 2014], and c_{ij}^1, p_{ij}^1, and p_{ij}^2 are uniformly generated between 0 and 100. Seven instances of biobjective FCNDPs with $n = 3, 4, \cdots, 9$ are generated where n is the number of nodes. We also generate $n = 3, 4, \cdots, 9$ biobjective FCNDP($\times$.1) scaling c_{ij}^2 and p_{ij}^2 by factor 0.1, i.e. the n=3 biobjective FCNDP($\times$.1) is obtained from n=3 biobjective FCNDP by multiplying c_{ij}^2 and p_{ij}^2 by 0.1.

We present an analysis of the shapes of nondominated frontiers in each problem and the resulting frontier by each algorithm. One instance from each problem is taken, i.e. the benchmark instance in [E GROSSMANN et al., 1982], the m=5 instance from biobjective FLPs, and the n=5 instance from biobjective FCNDPs, and solved using the three algorithms, Augmecon, Can, and EnpoBomip. As shown in Fig. 6.1, Augmecon provides an approximation of nondominated frontier using a number of nondominated points. The Can finds all ESN points and gives a very rough approximation of nondominated frontier. The EnpoBomip finds exact nondominated frontiers for all instances solved. Based on Fig. 6.1, Augmecon gives a better approximation than Can since Can does not show the complexities in nondominated frontier. For instance in n=5 biobjective FCNDP, the nondominated frontier has very complex shape while the approximation by Can is very smooth and includes only a few number of points. Hence, Can is good only for problems with very smooth and convex-like nondominated frontiers. For example in m=5 biobjective FLP, Can gives a good approximation since the nondominated frontier is very smooth. The approximation by Augmecon does not represent the shape of the nondominated frontier very well, and its quality is highly dependent on Q; when Q is very large, Augmecon provides a good approximation. The EnpoBomip always finds the exact nondominated frontier for any shape of nondominated frontier and all of the problems considered in this thesis. Among the three algorithms, only EnpoBomip is able to distinguish nondominated line segments while Augmecon and Can are only able to find nondominated points.

Table 6.1 presents the results of solving all instances of biobjective SCPN, FLPs, and FCNDPs using three algorithms. For example, the exact nondominated frontier of the n=3 biobjective FCNDP consists of 4 line segments and 4 isolated points, and includes 6 ESN points; the Augmecon approximates using 1,317 nondominated points, Can finds the 6 ESN points, and EnpoBomip finds the exact nondominated

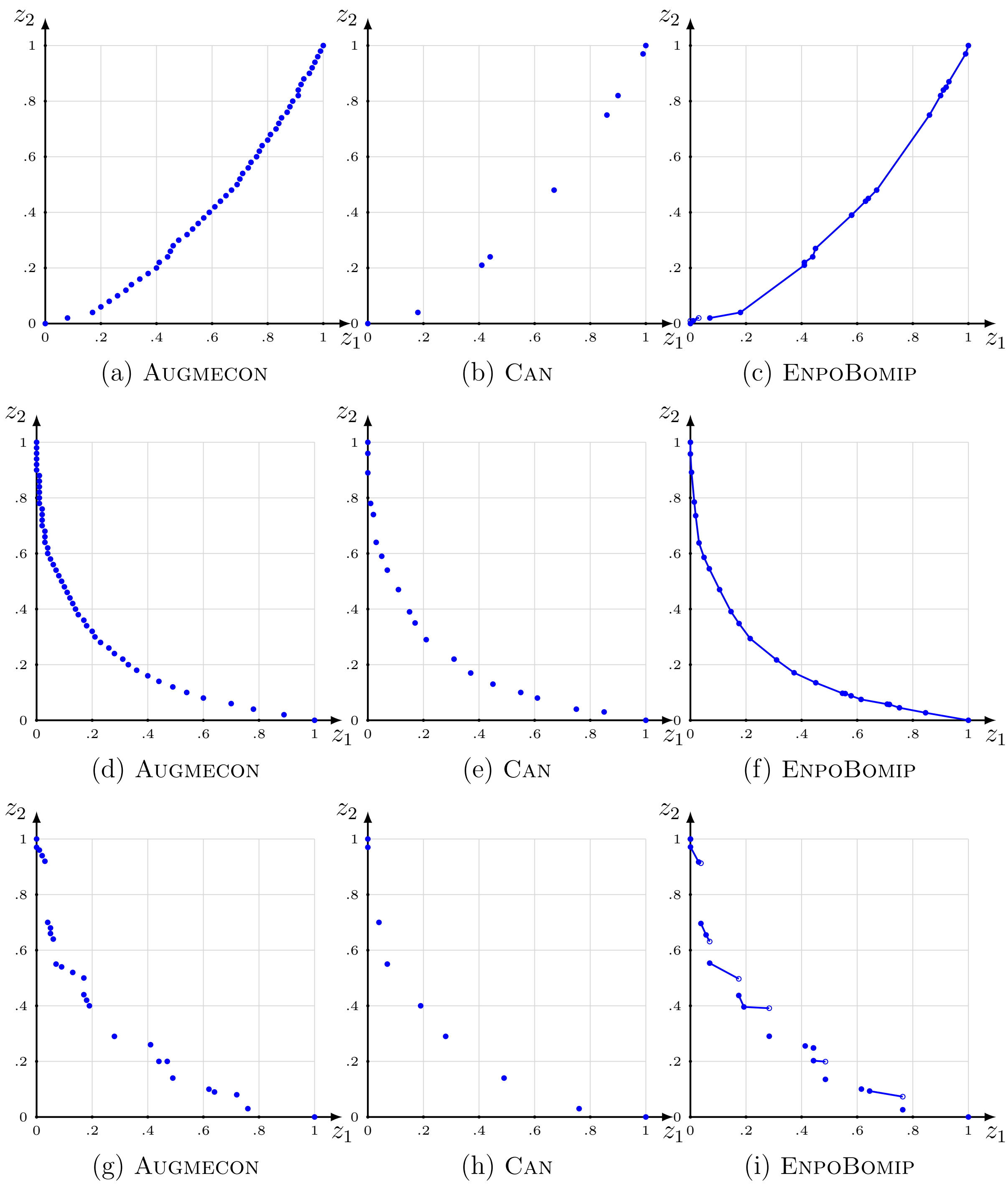


Figure 6.1: The resulting nondominated frontiers using AUGMECON (a,d, and g), CAN (b,e, and h), and ENPOBOMIP (c,f, and i), found for instances of biobjective SCPN (a-c), FLP (d-f), and FCNDP (g-i)

frontier. The Augmecon cannot solve the n=9 biobjective FCNDP($\times$.1) in 15,000 seconds. Based on Table 6.1, Augmecon finds the exact nondominated points for three instances, i.e. n=4 biobjective FCNDP and n=3,4 biobjective FCNDPs($\times$.1); however, the nondominated frontiers of these instances only consists of a few number of points and does not include line segments. Also note that Q is sufficiently large, however, with small Q, the Augmecon may dismiss some of the nondominated points. The Can always finds all ESN points and gives a very rough approximation of nondominated frontier. And EnpoBomip always finds exact nondominated frontiers. The nondominated frontiers of instances of biobjective SCPN and FLPs only consists of line segments while the small instances of biobjective FCNDPs include a few nondominated isolated points. The biobjective FCNDPs($\times$.1) possess the most discrete-like nondominated frontiers; the n=3,4 instances include only nondominated points while the remaining instances consists of both nondominated points and line segments. Hence, if coefficients of continuous variables in objective functions, i.e. c_{ij}^2 and p_{ij}^2, are diminished, the nondominated frontier becomes more discrete-like.

As the Can does not provide reasonable approximation of nondominated frontiers, we do not include it in CPU time analysis. Also note that since Can only finds a very small number of nondominated points, its running time is expected to be very small. Table 6.2 presents a comparative analysis of running times of Augmecon and EnpoBomip on instances of biobjective SCPN, FLPs and FCNDPs. The number of MILP resolutions and the total CPU time are presented for Augmecon. For EnpoBomip, the number of MILP and LP problems solved, the total time spent for solving MILP and LP problems, and the total running time of the algorithm are provided. For example, the Augmecon solves 3,494 MILP problems and runs for 12,848.55 seconds to provide an approximation of nondominated frontier of n=8 biobjective FCNDP($\times$.1), while EnpoBomip solves 130 MILP problems in 495.32 seconds and 195 LP problems in 27.62 seconds and, in total, runs for 523.35 seconds to find the exact nondominated frontier of n=8 biobjective FCNDP(×.1). The Augmecon cannot solve the n=9 biobjective FCNDP(×.1) in 15,000 seconds, while EnpoBomip solves this problem in 902.48 seconds. The total CPU time of EnpoBomip, especially in large instances, is significantly smaller than Augmecon, hence EnpoBomip outperforms Augmecon from the view of solution quality and CPU time.

Table 6.1: Quality of the solutions by the three algorithms

	AUG (NDP)	CAN (ESN)	ENP(NDLS,NDP)
SCPN	10,001	9	(22,0)
FLP			
m=5	10,001	20	(24,0)
m=10	10,001	47	(68,0)
m=15	9,885	89	(113,0)
m=20	9,962	121	(201,0)
m=25	10,001	176	(255,0)
m=30	9,854	207	(293,0)
m=35	9,766	273	(380,0)
FCNDP			
n=3	1,317	6	(4,4)
n=4	8	3	(0,8)
n=5	2,512	9	(11,6)
n=6	8,346	20	(69,7)
n=7	7,745	24	(94,0)
n=8	8,354	32	(169,0)
n=9	7,708	44	(144,0)
FCNDP($\times$.1)			
n=3	1	1	(0,1)
n=4	5	3	(0,5)
n=5	398	7	(5,10)
n=6	6,595	11	(29,11)
n=7	6,916	8	(31,4)
n=8	3,492	10	(47,6)
n=9	—*	14	(78,11)

AUG (NDP): number of nondominated points found by AUGMECON, CAN (ESN): number of ESN points found by CAN, ENP (NDLS,NDP): number of nondominated line segments and nondominated points found by ENPOBOMIP

$*$ the instance is not solved in 15,000 seconds

Table 6.2: Effectiveness analysis of Augmecon v.s. EnpoBomip

	Augmecon		EnpoBomip				
	MILP	TT	MILP	T(MILP)	LP	T(LP)	TT
SCPN	10,003	1,483.76	45	7.15	114	12.45	19.66
FLP							
m=5	10,003	924.24	33	2.79	131	9.70	12.57
m=10	10,003	945.49	98	9.82	515	41.75	51.86
m=15	9,887	1,081.00	145	15.59	869	80.79	96.98
m=20	9,964	1,316.60	275	40.34	1,807	195.65	237.39
m=25	10,003	1,573.24	345	61.98	2,420	306.59	370.90
m=30	9,856	1,818.51	391	82.77	2,899	429.28	515.41
m=35	9,768	2,721.94	495	231.97	3,776	1,415.21	1,807.29
FCNDP							
n=3	1,319	225.67	21	2.24	16	1.61	3.86
n=4	10	2.02	17	2.20	8	0.87	3.08
n=5	2,514	359.36	40	5.94	39	4.09	10.06
n=6	8,348	1,396.80	189	40.23	338	40.31	80.90
n=7	7,747	2,031.47	212	68.32	494	65.19	134.18
n=8	8,356	4,153.38	405	231.72	1,289	192.91	426.86
n=9	7,710	3,925.49	305	228.00	992	182.99	413.25
FCNDP($\times$.1)							
n=3	2	0.74	3	0.35	1	0.11	0.46
n=4	7	1.32	11	1.36	5	0.48	1.84
n=5	400	66.18	32	4.98	27	2.56	7.57
n=6	6,597	2,066.65	98	44.74	144	16.15	61.05
n=7	6,918	4,697.94	82	71.37	135	19.21	90.78
n=8	3,494	12,848.55	130	495.32	195	27.62	523.35
n=9	—	—*	209	842.31	328	59.23	902.48

MILP: number of MILP resolutions, TT: total running time, T(MILP): total CPU time spent for MILP resolutions, LP: number of LP resolutions, T(LP): total CPU time spent for LP resolutions

$*$ the instance is not solved in 15,000 seconds

A sensitivity analysis on n=5 biobjective FCNDP is presented to analyze the effects of objective functions coefficients on the shape of nondominated frontiers and the solution quality of the three algorithms. We consider four cases that are obtained by multiplying p^2 and c^2 by 0.01, 0.1, 1 and 100. The third case, i.e. $p^2, c^2 \times 1$, is shown in Fig. 6.1 while the other three cases are presented in Fig. 6.2. Since p^2 and c^2 are coefficients of continuous variables, scaling them by a factor $<$1 results in discrete-like nondominated frontiers while multiplying them by a factor $>$1 makes a continuous-like nondominated frontier. In $p^2, c^2 \times 0.01$ and 0.1, the discrete variables in objective functions are dominant since they have large coefficients, while in $p^2, c^2 \times 100$, continuous variables are determinant. As shown in Figs. 6.1-6.2, when scaling factor of p^2, c^2 increases, the nondominated frontier includes more line segments than isolated points; when the factor is 0.01, the nondominated frontier only consists of isolated points, while when the factor is 100, the nondominated frontier only includes line segments.

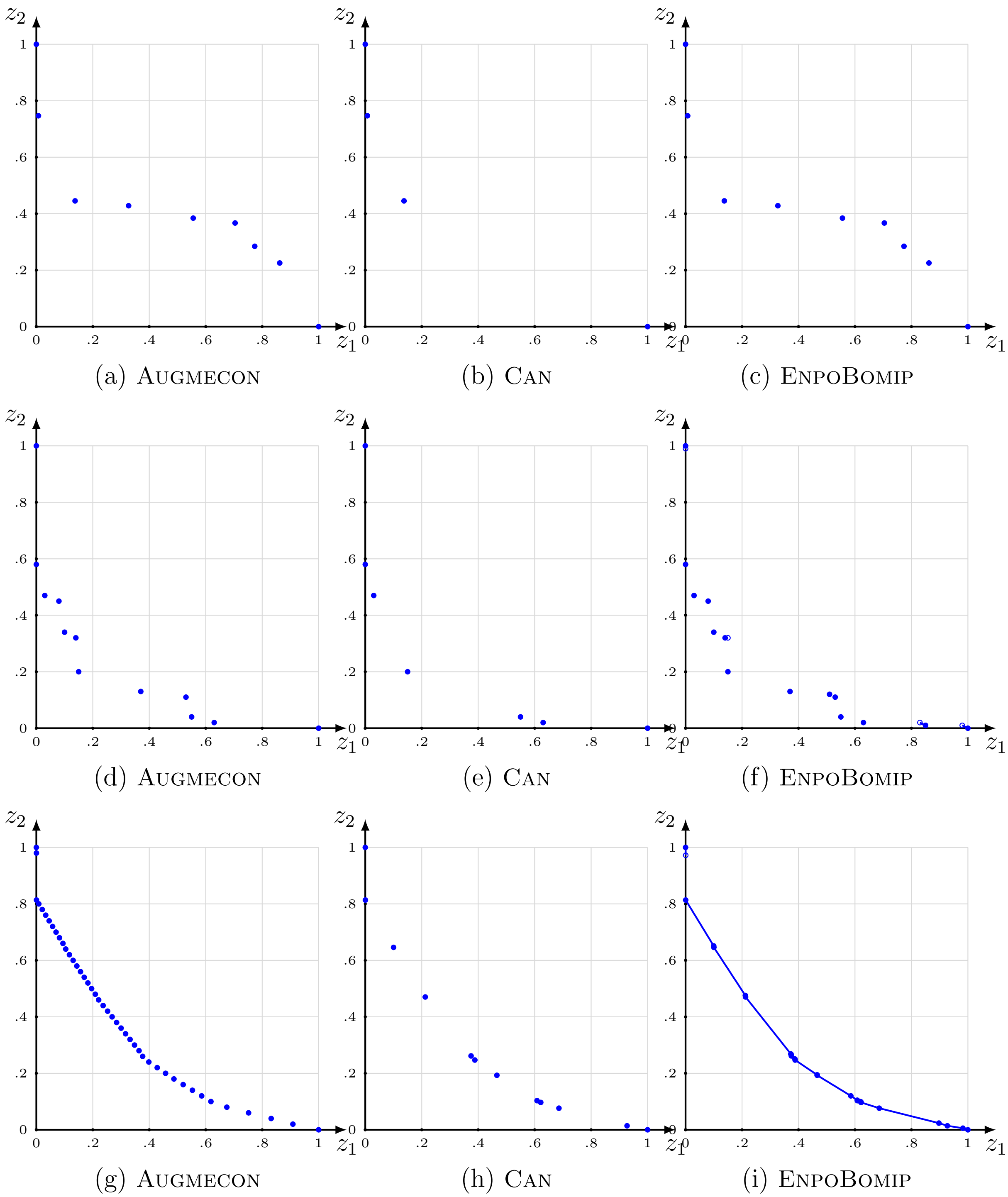


Figure 6.2: Nondominated frontiers for n=3 instance of biobjective FCNDP with $p^2, c^2 \times .01$ (a-c), $p^2, c^2 \times .1$ (d-f), and $p^2, c^2 \times 100$ (g-i), found using Augmecon (a,d, and g), Can (b,e, and h), and EnpoBomip (c,f, and i).

6.2 Biobjective nonlinear process networks

In this section, we present analysis of results to provide a comparison of ϵ-con+OA, ϵ–OA, ϵ-MINLP, and T-ϵ-con. We use ILOG CPLEX 12.5 Optimizer and CONOPT 3.15I. In both ϵ-MINLP and T-ϵ-con, the MINLP subproblems are solved using DICOPT. Note that T-ϵ-con differs from ϵ-MINLP as it does not include the augmented part in the objective function ($\mu\frac{s}{r}$). We solve the 3 and 8-process network problems for Q's between 10 and 10.000 using the four algorithms. The value of μ is set to 10^{-4} in our experiment.

Table 6.3 provides a comparative analysis on performance of the four algorithms for the examples considered in this thesis. This table shows the number of NLP subproblems, the total and average time of NLP subproblems, the number of Master problems, and the total and average time of Master problems for ϵ-con+OA and ϵ–OA, and the total running time for different values of Q for the four algorithms. As shown in Table 6.3, the number of NLP subproblems and Master problems that are solved, specially for large values of Q, is significantly smaller in ϵ–OA compared to ϵ-con+OA. This is also represented in Fig. 6.3 that are drawn on *log-log* scale. Total CPU times took to solve the bi-objective 3 and 8-process problems for different values of Q are shown in Fig. 6.4. As it is seen, the running times of ϵ-con+OA, ϵ-MINLP, and T-ϵ-con dramatically increase with the large values of Q; however, the ϵ–OA solves the problems in a very short computing time even for very large values of Q. Therefore, based on the results of the number of NLP subproblems and Master problems and the total running time, the ϵ–OA algorithm is very efficient and significantly outperforms the other three algorithms. The comparison between T-ϵ-con and ϵ–OA highlights the contribution of this thesis as the existing work on MOMINLPs mostly use T-ϵ-con [Martinez and Eliceche, 2008, Martínez and Eliceche, 2011, Čuček et al., 2012b, Čuček et al., 2012a, Chakraborty and Linninger, 2002]. The ϵ–OA enhances the quality of the solutions compared to T-ϵ-con since T-ϵ-con may produce dominated solutions. Moreover, ϵ–OA is significantly faster (see Table 6.3 and Fig. 6.4). Hence, we propose a very effective algorithm to generate guaranteed efficient solutions of the bi-objective nonlinear network synthesis problem.

Note that the ϵ-con+OA is dominated by ϵ-MINLP with respect to CPU time since it involves a straightforward use of OA in each iteration of the augmented ϵ-constraint method. When running ϵ-MINLP, GAMS uses the results of one MINLP subproblem in the next iteration as advanced basis. In ϵ-con+OA, on the other hand, the algorithm starts with no suggesting solution at each iteration. The T-ϵ-con and ϵ-MINLP perform almost similar from the view of CPU time; however, ϵ-MINLP is superior as it guarantees efficiency without additional computational cost.

Table 6.3: Analysis of performance of the algorithms on 3 and 8-process network problems.

			$Q = 10$	$Q = 100$	$Q = 1000$	$Q = 10000$
3 Process Network	ϵ-con + OA	N	16	106	1006	10006
		$N.T.T$	1.35	8.51	88.13	717.27
		$N.A.T$	0.08	0.08	0.09	0.07
		M	14	117	1011	10547
		$M.T.T$	1.28	12.66	109.96	1024.31
		$M.A.T$	0.09	0.11	0.11	0.10
		$T.T$	2.87	21.40	198.63	1748.91
	ϵ-OA	N	9	23	100	151
		$N.T.T$	0.71	1.53	6.68	10.65
		$N.A.T$	0.08	0.07	0.07	0.07
		M	7	34	108	155
		$M.T.T$	0.68	2.80	8.74	13.87
		$M.A.T$	0.10	0.08	0.08	0.09
		$T.T$	1.56	4.66	18.46	32.26
	ϵ-MINLP	$T.T$	2.04	13.28	130.92	1209.90
	T-ϵ-con	$T.T$	1.76	13.54	132.47	1318.26
	E		4	18	95	146
8 Process Network	ϵ-con + OA	N	21	123	1154	11464
		$N.T.T$	1.50	8.37	87.58	987.87
		$N.A.T$	0.07	0.07	0.08	0.09
		M	16	129	1266	12644
		$M.T.T$	1.35	10.53	126.14	1976.34
		$M.A.T$	0.08	0.08	0.10	0.16
		$T.T$	3.03	19.15	216.78	3009.50
	ϵ-OA	N	14	28	87	149
		$N.T.T$	0.96	1.99	6.08	10.49
		$N.A.T$	0.07	0.07	0.07	0.07
		M	9	33	139	250
		$M.T.T$	0.75	2.88	12.99	23.41
		$M.A.T$	0.08	0.09	0.09	0.09
		$T.T$	1.9	5.44	33.06	101.49
	ϵ-MINLP	$T.T$	1.62	13.27	136.69	1428.57
	T-ϵ-con	$T.T$	1.82	14.2	140.18	1401.07
	E		4	14	61	110

N: number of NLP subproblems, $N.T.T$: total time of solving NLP subproblems, $N.A.T$: average time of solving NLP subproblems, M: number of Master problems, $M.T.T$: total time of solving Master problems, $M.A.T$: average time of solving Master problems, $T.T$: total CPU time, E: number of efficient solutions found.

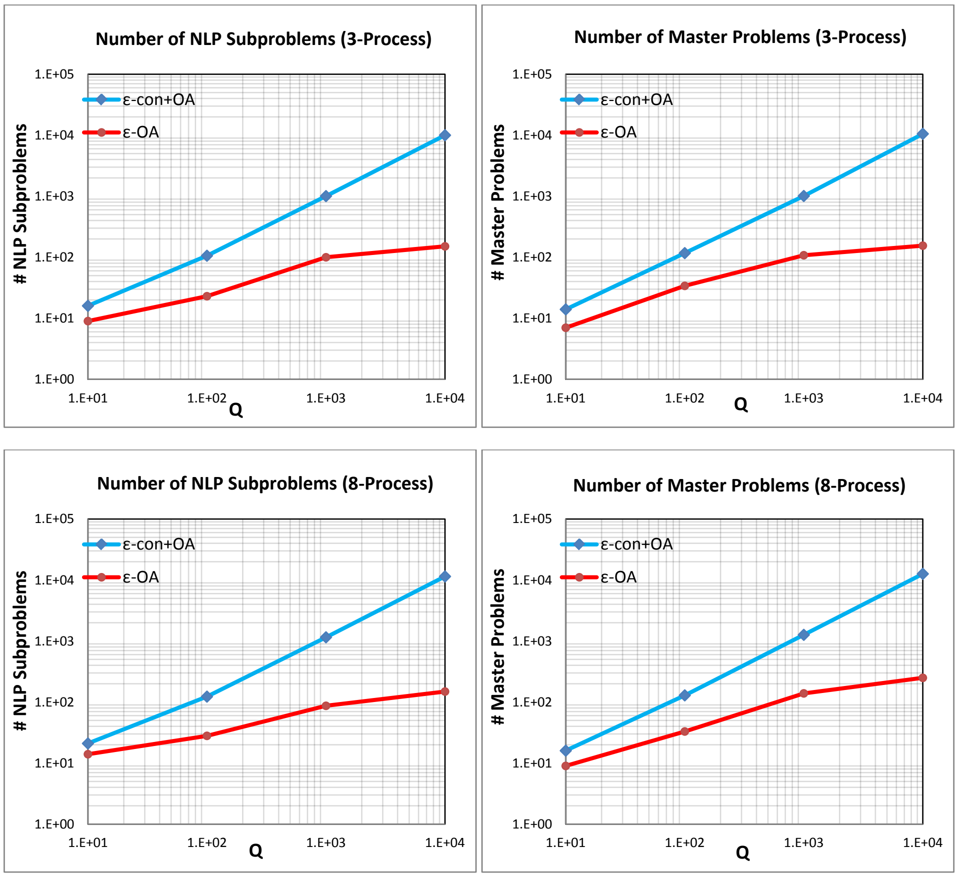


Figure 6.3: Number of NLP subproblems and Master problems solved using ϵ-OA and ϵ-con+OA for the 3 and 8-process networks

Table 6.3 also represents the number of efficient solutions that are found for different values of Q. Note that the same efficient frontiers are found using ϵ-OA, ϵ-con+OA, ϵ-MINLP, and T-ϵ-con. The ϵ-con+OA and ϵ-MINLP both perform the augmented ϵ-constraint method; however, they use OA and DICOPT, respectively, to solve the MINLP subproblems in each iteration. The ϵ-OA is an improved version of ϵ-con-OA after modifications and speed-up techniques. In ϵ-OA, the unnecessary operations throughout the algorithm are identified and skipped. Hence, ϵ-OA, ϵ-con+OA,

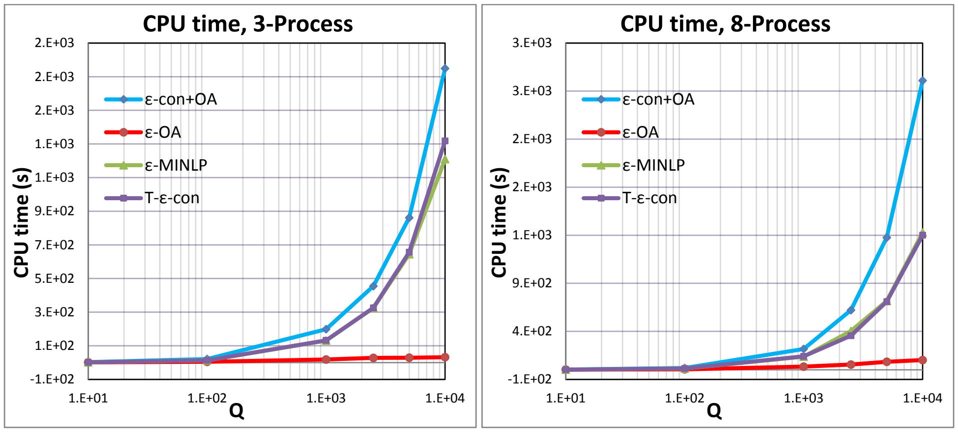


Figure 6.4: CPU times of the four algorithms for the 3 and 8-process networks

and ϵ-MINLP identify the same efficient solutions. The T-ϵ-con, however, perform a different approach that involves solving subproblems without the augmented part in the objective function. Although it is possible that T-ϵ-con finds weekly efficient solutions, it does not happen in the two examples considered in our experiment due to the structure of the objective spaces.

The number of efficient solutions that are found increases when Q increases (as seen in Table 6.3); hence, this provides an accurate representation of the efficient frontier. However, more computational effort is required as Q increases as shown in Fig. 6.4. Although the running times for all of the algorithms increases, the CPU time for ϵ–OA exhibits a modest increase while the solution times of other three algorithms dramatically increase as Q increases. Hence, ϵ–OA does not suffer from the dimensionality of the problem compared to other competing algorithms. Therefore, it is fair to say that large-scale bi-objective nonlinear network synthesis problems can be solved more effectively by ϵ–OA compared to other methods.

The efficient frontiers of the 3 and 8-process networks are drawn in Fig. 6.5. The efficient frontier of 3-process network comprises 3 sections. Section (I) is when there is very tight constraint on the value of the second objective function. In this case, it is not possible to use units 2 or 3 while the limitation on the value of the second objective is satisfied; hence, the material B is purchased from outside. Section (II) includes units 1 and 2, and Section (III) involves units 1 and 3. A smooth change occurs in the efficient frontier due to the change in the values of continuous variables x within each section. The topologies of these three sections are represented in Fig. 6.6. Similarly, the efficient frontier of the 8-process problem involves two sections. The units 2, 4, and 6 are present in section (I), and section (II) comprises units 2, 4, 6, and 8. Fig. 6.7 shows the topologies that are found in these sections.

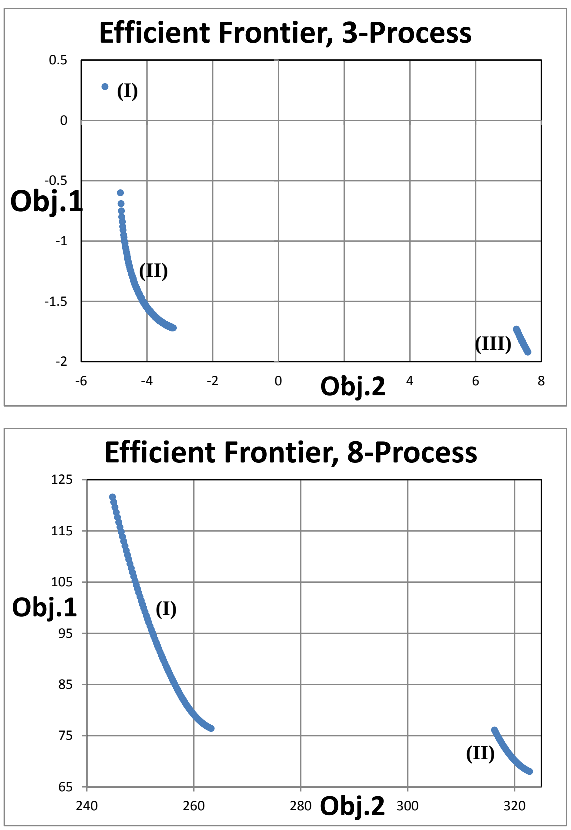


Figure 6.5: Efficient frontiers for the 3 and 8-process networks

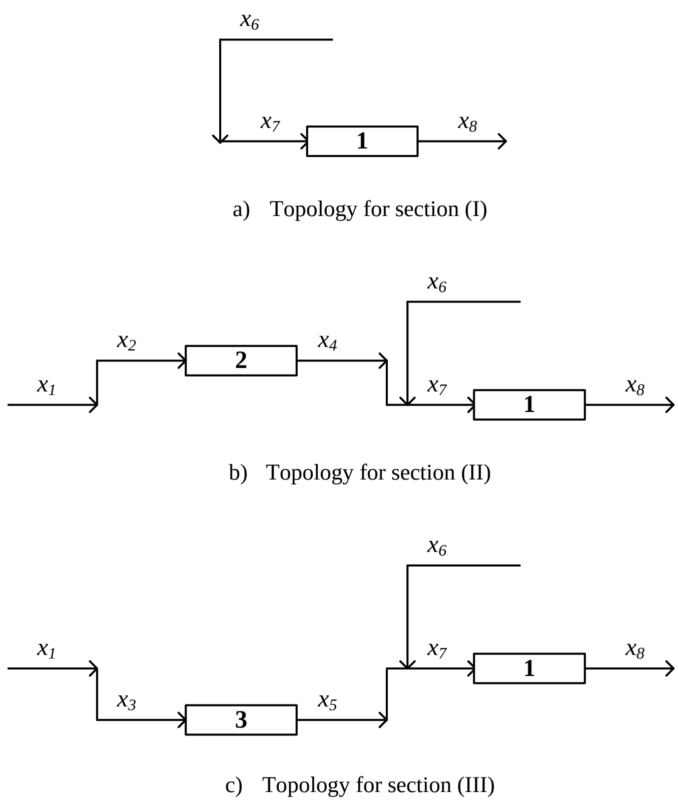
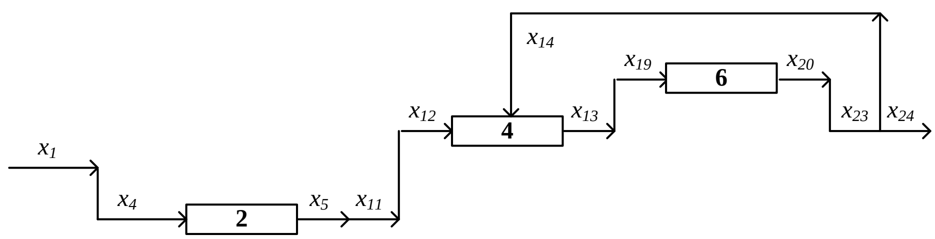
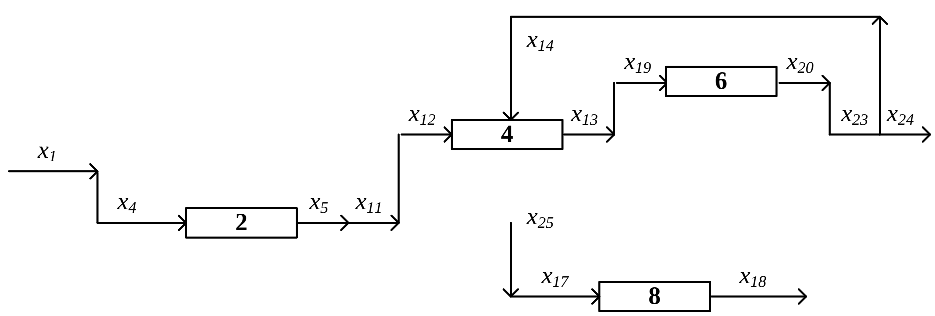


Figure 6.6: Topologies for sections (I), (II), and (III) in the efficient frontier of the 3-process network synthesis problem



a) Topology for section (I)



b) Topology for section (II)

Figure 6.7: Topologies for sections (I) and (II) in the efficient frontier of the 8-process network synthesis problem

Chapter 7

CONCLUSION

Some possible complexities in nondominated frontiers of BOMILPs is described using an illustrative example. A novel algorithm, EnpoBomip, is presented that is, to the best of our knowledge, the first algorithm in literature that finds the exact nondominated frontier of general BOMILPs with all alternative values of integer variables associated with each nondominated point. An experiment is conducted to present a solution quality and computational performance analysis between EnpoBomip, Augmecon and Can on three well-known problems; biobjective synthesis of complex production networks, biobjective facility location problems, and biobjective fixed charge network design problems. Based on our experiment, Can produces a few number of nondominated points and provides a very rough approximation of nondominated frontier. The solution quality of Augmecon can be improved by increasing Q, however, it can never find guaranteed exact nondominated frontier and distinguish line segments. The EnpoBomip is the most effective algorithm from the view of solution quality, as it finds the exact nondominated frontier, and computational performance, as it is very faster than Augmecon.

Our sensitivity analysis shows that when the coefficients of continuous variables scaled by a factor >1, the nondominated frontier becomes more continuous-like and contains more line segments while if the factor is <1, the frontier becomes more discrete-like and contains more isolated points. The Can has better solution quality when applied to continuous-like nondominated frontiers since the feasible set in objective space is more similar to convex sets, while Augmecon works better with discrete-like nondominated frontiers since they have more isolated points than line segments. The EnpoBomip finds exact nondominated frontier in all cases.

We also present a novel method based on augmented ϵ-constraint and logic based OA to solve bi-objective nonlinear network synthesis problems. The theoretical analysis of the proposed algorithm (ϵ-OA) proves that the solutions generated are efficient for convex functions. We illustrate the effectiveness of the ϵ-OA algorithm on two well-known benchmark problems (3 process and 8 process). The analysis of the results show that the ϵ-OA algorithm is very effective than straightforward use of OA with augmented ϵ-constraint algorithm, the augmented ϵ-constraint without OA, and the traditional ϵ-constraint.

BIBLIOGRAPHY

[Aneja and Nair, 1979] Aneja, Y. P. and Nair, K. (1979). Bicriteria transportation problem. *Management Science*, 25(1):73–78.

[Chakraborty and Linninger, 2002] Chakraborty, A. and Linninger, A. A. (2002). Plant-wide waste management. 1. synthesis and multiobjective design. *Industrial & engineering chemistry research*, 41(18):4591–4604.

[Cohon, 2013] Cohon, J. L. (2013). *Multiobjective programming and planning*. Courier Dover Publications.

[Čuček et al., 2012a] Čuček, L., Klemeš, J. J., Varbanov, P. S., and Kravanja, Z. (2012a). Reducing the dimensionality of criteria in multi-objective optimisation of biomass energy supply chains. *CHEMICAL ENGINEERING*, 29.

[Čuček et al., 2012b] Čuček, L., Varbanov, P. S., Klemeš, J. J., and Kravanja, Z. (2012b). Total footprints-based multi-criteria optimisation of regional biomass energy supply chains. *Energy*, 44(1):135–145.

[Duran and Grossmann, 1986] Duran, M. A. and Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339.

[E GROSSMANN et al., 1982] E GROSSMANN, I., Drabbant, R., and Jain, R. K. (1982). Incorporating toxicology in the synthesis of industrial chemical complexes. *Chemical Engineering Communications*, 17(1-6):151–170.

[Figueira et al., 2005] Figueira, J., Greco, S., and Ehrgott, M. (2005). *Multiple criteria decision analysis: state of the art surveys*, volume 78. Springer.

[Geoffrion, 1972] Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260.

[Goicoechea et al., 1976] Goicoechea, A., Duckstein, L., and Fogel, M. M. (1976). Multiobjective programing in watershed management: A study of the charleston watershed. *Water Resources Research*, 12(6):1085–1092.

[Grossmann and Lee, 2003] Grossmann, I. E. and Lee, S. (2003). Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, 26(1):83–100.

[Grossmann and Santibanez, 1980] Grossmann, I. E. and Santibanez, J. (1980). Applications of mixed-integer linear programming in process synthesis. *Computers & Chemical Engineering*, 4(4):205–214.

[Grossmann and Trespalacios, 2013] Grossmann, I. E. and Trespalacios, F. (2013). Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE Journal*, 59(9):3276–3295.

[Haimes et al., 1971] Haimes, Y. Y., Ladson, L., and Wismer, D. A. (1971). Bicriterion formulation of problems of integrated system identification and system optimization.

[Hooker, 1994] Hooker, J. N. (1994). Logic-based methods for optimization. In *Principles and Practice of Constraint Programming*, pages 336–349. Springer.

[Hwang et al., 1979] Hwang, C.-L., Masud, A. S. M., Paidy, S. R., and Yoon, K. P. (1979). *Multiple objective decision making, methods and applications: a state-of-the-art survey*, volume 164. Springer Berlin.

[Lee and Grossmann, 2005] Lee, S. and Grossmann, I. E. (2005). Logic-based modeling and solution of nonlinear discrete/continuous optimization problems. *Annals of Operations Research*, 139(1):267–288.

[Marler and Arora, 2004] Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395.

[Martínez and Eliceche, 2011] Martínez, P. and Eliceche, A. M. (2011). Bi-objective minimization of environmental impact and cost in utility plants. *Computers & Chemical Engineering*, 35(8):1478–1487.

[Martinez and Eliceche, 2008] Martinez, P. E. and Eliceche, A. M. (2008). Minimization of life cycle greenhouse emissions and cost in the operation of steam and power plants. *Computer Aided Chemical Engineering*, 25:1107–1112.

[Mavrotas, 2009] Mavrotas, G. (2009). Effective implementation of the e-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2):455–465.

[Mavrotas and Florios, 2013] Mavrotas, G. and Florios, K. (2013). An improved version of the augmented e-constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18):9652–9669.

[Nabar and Schrage, 1991] Nabar, S. and Schrage, L. (1991). Modeling and solving nonlinear integer programming problems. In *annual AIChE meeting, Chicago*.

[Özpeynirci and Köksalan, 2010] Özpeynirci, Ö. and Köksalan, M. (2010). An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. *Management Science*, 56(12):2302–2315.

[Pedersen et al., 2008] Pedersen, C. R., Nielsen, L. R., and Andersen, K. A. (2008). The bicriterion multimodal assignment problem: introduction, analysis, and experimental results. *INFORMS Journal on Computing*, 20(3):400–411.

[Przybylski et al., 2008] Przybylski, A., Gandibleux, X., and Ehrgott, M. (2008). Two phase algorithms for the bi-objective assignment problem. *European Journal of Operational Research*, 185(2):509–533.

[Przybylski et al., 2010] Przybylski, A., Gandibleux, X., and Ehrgott, M. (2010). A recursive algorithm for finding all nondominated extreme points in the outcome set of a multiobjective integer programme. *INFORMS Journal on Computing*, 22(3):371–386.

[Przybylski et al., 2011] Przybylski, A., Gandibleux, X., and Ehrgott, M. (2011). The two-phase method for multiobjective combinatorial optimization problems.

[Raith and Ehrgott, 2009] Raith, A. and Ehrgott, M. (2009). A two-phase algorithm for the biobjective integer minimum cost flow problem. *Computers & Operations Research*, 36(6):1945–1954.

[Raman and Grossmann, 1994] Raman, R. and Grossmann, I. E. (1994). Modelling and computational techniques for logic based integer programming. *Computers & Chemical Engineering*, 18(7):563–578.

[Sawaya and Grossmann, 2007] Sawaya, N. W. and Grossmann, I. E. (2007). Computational implementation of non-linear convex hull reformulation. *Computers & chemical engineering*, 31(7):856–866.

[Solanki et al., 1993] Solanki, R. S., Appino, P. A., and Cohon, J. L. (1993). Approximating the noninferior set in multiobjective linear programming problems. *European Journal of Operational Research*, 68(3):356–373.

[Stadler, 1988] Stadler, W. (1988). Fundamentals of multicriteria optimization. In *Multicriteria Optimization in Engineering and in the Sciences*, pages 1–25. Springer.

[Stidsen et al., 2014] Stidsen, T., Andersen, K. A., and Dammann, B. (2014). A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4):1009–1032.

[Türkay and Grossmann, 1996a] Türkay, M. and Grossmann, I. E. (1996a). Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions. *Industrial & engineering chemistry research*, 35(8):2611–2623.

[Türkay and Grossmann, 1996b] Türkay, M. and Grossmann, I. E. (1996b). Logic-based minlp algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering*, 20(8):959–978.

[Turkay and Grossmann, 1998] Turkay, M. and Grossmann, I. E. (1998). Structural flowsheet optimization with complex investment cost functions. *Computers & chemical engineering*, 22(4):673–686.

[Ulungu, 1995] Ulungu, E. L., J. T. (1995). The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations Comput. Decision Sci.*, 20(2):149–165.

[Van Veldhuizen and Lamont, 2000] Van Veldhuizen, D. A. and Lamont, G. B. (2000). Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary computation*, 8(2):125–147.

[Zhang and Reimann, 2014] Zhang, W. and Reimann, M. (2014). A simple augmented e-constraint method for multi-objective mathematical integer programming problems. *European Journal of Operational Research*, 234(1):15–24.