COPPE
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
UFRJ

MODELING AND ECONOMIC OPTIMIZATION OF AN INDUSTRIAL SITE FOR NATURAL GAS PROCESSING

Tayná Embiruçu Gonsalves de Souza

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Química, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Química.

Orientadores: Argimiro Resende Secchi
Letícia Cotia dos Santos

Rio de Janeiro
Setembro de 2022

MODELING AND ECONOMIC OPTIMIZATION OF AN INDUSTRIAL SITE FOR NATURAL GAS PROCESSING

Tayná Embiruçu Gonsalves de Souza

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA QUÍMICA.

Orientadores: Argimiro Resende Secchi
Letícia Cotia dos Santos

Aprovada por: Prof. Argimiro Resende Secchi
Drª. Letícia Cotia dos Santos
Dr. Fabio Cesar Diehl
Profª. Karen Valverde Pontes Vater
Prof. Mauricio Bezerra de Souza Junior

RIO DE JANEIRO, RJ - BRASIL
SETEMBRO DE 2022

Souza, Tayná Embiruçu Gonsalves de

Modeling and Economic Optimization of an Industrial Site for Natural Gas Processing/Tayná Embiruçu Gonsalves de Souza. – Rio de Janeiro: UFRJ/COPPE, 2022.

XXI, 125 p.: il.; 29, 7cm.

Orientadores: Argimiro Resende Secchi

Letícia Cotia dos Santos

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Química, 2022.

Referências Bibliográficas: p. 120 – 125.

1. Otimização. 2. Processamento de gás natural. 3. UPGN. 4. Simulação de processos. I. Secchi, Argimiro Resende *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

Acknowledgments

Primeiramente, agradeço à Universidade Pública e a todos que trabalham para mantê-la funcionando apesar da falta de suporte e reconhecimento, na *balbúrdia* necessária para melhorar esse país. Especialmente à UFBA, onde me graduei e tive o primeiro contato com o mundo da pesquisa através da Iniciação Científica.

Aos meus orientadores, Argimiro e Letícia, minha profunda gratidão por todo o apoio durante esse processo e por serem exemplos de como bons trabalhos podem surgir a partir de uma orientação inteligente, humana e respeitosa, como deve ser a Academia.

Aos meus amigos e família, obrigada por todo o apoio e compreensão durante esses longos anos. Vocês foram fundamentais em todas as conversas, risadas, colos, meditações e abraços. À Petrobras, agradeço pelo suporte que viabilizou esse trabalho e pelo reconhecimento de que é necessário investir em ciência e pesquisa. Agradeço ainda aos meus *mestres* e colegas de trabalho por todos os ensinamentos que permitiram o surgimento desse trabalho tão querido.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MODELAGEM E OTIMIZAÇÃO ECONÔMICA DE UM SITE INDUSTRIAL DE PROCESSAMENTO DE GÁS NATURAL

Tayná Embiruçu Gonsalves de Souza

Setembro/2022

Orientadores: Argimiro Resende Secchi
Letícia Cotia dos Santos

Programa: Engenharia Química

O objetivo deste trabalho foi desenvolver um modelo integrado de simulação e otimização para determinar o ponto ótimo econômico de operação de um *site* de processamento de gás natural com múltiplas unidades de processamento e múltiplas fontes de gás rico. Plantas de processamento de gás natural lidam com cenários dinâmicos, sem controle direto acerca das vazões e características do gás de entrada, o que pode gerar uma operação sub-ótima. Uma estrutura integrada de simulação e otimização não linear mista inteira (MINLP) foi proposta, utilizando Aspen HYSYS para construção da simulação rigorosa do processo, Python para a otimização e Microsoft Excel como interface de transferência de dados. O modelo de simulação foi validado utilizando dados industriais de uma planta da Petrobras, indicando boa aderência (desvio médio de 4 % na vazão dos produtos). O problema de otimização MINLP proposto foi subdividido em um problema de programação não linear (NLP), resolvido por uma combinação dos algoritmos de enxame de partículas (PSO) e poliedros flexíveis em série, em conjunto com um problema de programação inteira mista (MIP), resolvido pelo método *branch-and-bound* utilizando a solução de um problema de programação linear (LP) como gerador de estimativas iniciais para os subproblemas NLP. Resultados do NLP mostraram ganho potencial de 7,3 % no lucro variável da planta estudada e indicaram a presença de máximos locais, explorados em detalhe por uma análise do espaço das variáveis utilizando o PSO. Os resultados do MINLP indicaram 9,1 % de aumento potencial no lucro (118.78 R$ / 1000 m^3) através da otimização da distribuição de gás rico para as unidades de processamento e da seleção do seu *status* de operação.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

MODELING AND ECONOMIC OPTIMIZATION OF AN INDUSTRIAL SITE FOR NATURAL GAS PROCESSING

Tayná Embiruçu Gonsalves de Souza

September/2022

Advisors: Argimiro Resende Secchi
Letícia Cotia dos Santos

Department: Chemical Engineering

The objective of this work was to develop an integrated simulation-optimization model to determine optimum economic operating point of an industrial site for natural gas processing, with multiple processing units and feedstock. Natural gas processing facilities deal with dynamic scenarios with no direct control over raw inlet gas flowrates and conditions, which might lead to suboptimal operation. An integrated simulation-optimization framework was proposed, using Aspen HYSYS for rigorous process simulation, Python for optimization and Microsoft Excel as data transfer interface. The simulation model was validated against actual industrial data from a Petrobras facility, indicating good fit (average 4 % deviation in product flowrates). The mixed-integer nonlinear optimization problem (MINLP) proposed was broken down into a nonlinear programming (NLP) contribution, solved by a combination of particle swarm (PSO) and flexible polyhedron methods in series, added to a mixed-integer programming (MIP) problem, solved by branch-and-bound method and using the result of a linear programming (LP) as an initial estimate generator for the NLP subproblems. NLP results showed a potential gain of 7.3 % on variable profit of the plant in study and indicated the presence of local maxima, further explored by an analysis of the space of variables using PSO. The MINLP results indicated 9.1 % (118.78 R$ / 1000 m^3) of potential profit increase by optimizing the raw gas distribution to the process units and selecting their operating status.

Contents

Acknowledgments **iv**

List of Figures **ix**

List of Tables **xii**

List of Symbols **xv**

List of Abbreviations **xix**

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 2

1.3 Dissertation Structure 3

2 Literature review **4**

2.1 Natural Gas 4

2.2 Natural Gas Processing Plants 5

2.2.1 Sales Gas Specification 5

2.2.2 Contaminants 7

2.2.3 The Process 7

2.2.4 Liquid Recovery Processes 10

2.2.5 Natural Gas Processing Plant Configurations 12

2.3 Modeling and Optimization of Natural Gas Processing Units 16

2.3.1 Thermodynamic Model 21

2.3.2 The COSTALD Method for Liquid Density 26

2.3.3 The Inside-Out Method for Distillation 28

2.3.4 Process Optimization Strategies 29

2.3.5 Nonlinear Local Optimization: Nelder-Mead Flexible Polyhedron Method 30

2.3.6 Nonlinear Global Optimization: Particle Swarm Method 31

2.3.7 Integer Optimization: Branch-and-Bound Method 32

2.4 Final Considerations 34

3 Methodology **35**
3.1 Process Description 36
3.2 Process Modeling and Simulation 39
3.2.1 Digital Modeling Approach 40
3.2.2 Component List 40
3.2.3 Plant Representation 40
3.2.4 Thermodynamic Model 45
3.2.5 Simulation Strategies 48
3.2.6 Validation Approach 52
3.3 The Simulation-Optimization Integration Tool 53
3.4 The Optimization Framework 53
3.4.1 Optimization Strategy 54
3.4.2 The Optimization Problem Formulation 58
3.4.3 Decision Variables 64
3.4.4 Inequality Constraints 65
3.4.5 Implementation Strategies 68

4 Results and Discussion **72**
4.1 Process Modeling and Simulation 72
4.1.1 Model Validation and Tuning 74
4.2 The Simulation-Optimization Integration Tool 76
4.3 The Optimization Problem 78
4.3.1 The NLP Problem 78
4.3.2 PSO: MINLP Analysis of the Variables Search Space 86
4.3.3 The MINLP Problem 92
4.3.4 NLP and MINLP Results: Profit Gain 99
4.4 Case Study: Changing Product Price Values 100

5 Conclusions and Suggestions for Future Works **103**

A PSO Python Code **105**

B LP Python Code **113**

C Simulation Mass Balance: Base-Case **118**

Bibliography **120**

List of Figures

2.1 Schematic of a typical processing layout from a natural gas processing plant. Created by the author. 9
2.2 Ranges of liquid extraction in natural gas processing facilities. Adapted from CAMPBELL (2013). 11
2.3 Schematic representation of a typical mechanical refrigeration gas cooling with propane. Created by the author based on POE and MOKHATAB (2017). 12
2.4 Schematic representation of a typical turboexpander refrigeration gas. Created by the author. 13
2.5 Schematic representation of a typical CO_2 removal plant by reactive absorption with alkanolamine solution. Created by the author. 15
2.6 Schematic representation of a typical liquid fractionating unit with LPG and C_5^+ production. Created by the author. 16
2.7 Simplified flowchart of Inside-Out solving method for multistage multicomponent distillation. Adapted from SEADER *et al.* (2011) and BANSAL and MANJARE (2016). 29

3.1 Schematic representation of the simulation-optimization framework built up in this work. 36
3.2 General schematic of the gas processing site in study. Created by the author. 38
3.3 Schematic diagram showing how reactive absorption CO_2 removal treating unit was represented in the simulation model. Created by the author. . 42
3.4 Schematic representation of the NGPU section (part of gas cooling) chosen for result comparison between PR and CPA EoS. Created by the author. 47
3.5 Phase envelope for NGPU-A inlet stream of three-phase vessel calculated with PR (red) and CPA (blue) EoS for comparison. Dashed-dotted region highlights NGPU operating range. 49
3.6 Phase envelope for NGPU-E inlet stream of three-phase vessel calculated with PR (red) and CPA (blue) EoS for comparison. Dashed-dotted region highlights NGPU operating range. 50

3.7 Overall schematic of the methodology proposed in this work. PSO, branch-and-bound and Nelder-Mead methods are employed in three main steps: pre-screening, MINLP solution and post-refinement. 58
3.8 Flowchart showing the steps carried out for initial polyhedron determination. 69

4.1 Example of information contained in the optimization tab of the simulation-optimization integration tool. Data contains lower/upper bounds and current value of all decision variables. 77
4.2 Space of variables investigation for three different levels of flowrate recycle tolerance in HYSYS, 400 simulation points. Objective function S presented as a function of u_1 and u_4. (a): default recycle tolerance, 2h 46min; (b): one-tenth of default recycle tolerance, 2h 41min; (c): one-hundredth of default recycle tolerance, 5h 35min. 81
4.3 Space of variables investigation for three different levels of flowrate recycle tolerance in HYSYS, 400 simulation points. Objective function presented S as a function of u_{11} and u_{12}. (a): default recycle tolerance, 1h 15min; (b) one-tenth of default recycle tolerance, 2h 05min; (c): one-hundredth of default recycle tolerance, 3h 02min. 81
4.4 Graphical results of the NLP problem. Values of total optimization time for each initial estimate set at three levels of simulation recycle tolerance. 82
4.5 Graphical results of the NLP problem with Nelder-Mead method, objective function S per iteration. Each line represents the NLP optimization for one given initial estimate from Table 4.3. 83
4.6 Graphical results of the NLP problem for values of the objective function for each initial estimate set for three simulation recycle tolerances. 84
4.7 Results of PSO pre-screening for the NLP problem. Chart shows the values of the objective function S in each of the 2100 function evaluations during the optimization procedure. 85
4.8 Results of PSO pre-screening coupled with Nelder-Mead local refinement for the NLP problem. Chart shows the values of the objective function S in each iteration of the optimization procedure. 86
4.9 Eleven 2D plots of u_3 (raw gas bypass flowrate from the outlet of SG-A to sales gas) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions. 87

4.10 Eleven 2D plots of u_8 (NGPU-A feed gas flowrate) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions. 88
4.11 3D plot of u_3 and u_8 against the objective function S. The star symbols represent the five highest objective function values encountered in distinguished PSO executions. 90
4.12 Eleven 2D plots of u_7 (NGPU-E main feed gas flowrate) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions. 91
4.13 Eleven 2D plots of u_{10} (NGPU-D feed gas flowrate) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions. 91
4.14 2D plot of y_2 (raw gas bypass operating status) against y_4 (NGPU-A operating status). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions. 93
4.15 Simplified branch-and-bound algorithm implemented for the MINLP problem. 94
4.16 Schematic diagram of the MINLP calculation phases, listing the values of the main parameters utilized for every method in each phase of problem solution. 95

List of Tables

2.1 Main items of Brazil's sales gas specification, extracted from resolution RANP 16/2008 (ANP, 2008). Values refer to limits of southeast, south and center-west regions. 6
2.2 Common contaminants in raw natural gas, reasons for removal and typical separation processes. Adapted from MONDAL *et al.* (2013). 8
2.3 Typical products of a natural gas processing plant. 10
2.4 Summary on approach of current literature work on simulation and optimization of NGPUs (continue on next page). 19
2.5 Constant values for COSTALD liquid density calculation. Taken from THOMSON *et al.* (1982) 27

3.1 List of products generated in each process unit studied in this work (NGPU-A/B/C/D/E and LFUs): sales gas, NGL (C_2^+), LPG and C_5^+ are final products destined for sale whereas residual gas and NGL (C_3^+) are intermediate streams. 39
3.2 List of equipment present in the digital simulation model and the corresponding solving methods and default specifications. 41
3.3 Gas dehydration/cooling processes used in each NGPU of this study and their products. 43
3.4 Comparison between PR and CPA EoS. Properties calculated for gas-phase outlet stream of NGPU-A three-phase vessel. 48
3.5 Comparison between PR and CPA EoS. Thermodynamic properties calculated for gas-phase outlet stream of NGPU-E three-phase vessel. 48
3.6 PSO default parameters (Equation 2.31) implemented in Python *pyswarm* module and used in this work. 56
3.7 Nelder-Mead method default parameters implemented in Python *scipy.optimize.minimize* module and used in this work. 56
3.8 Typical market price for sales gas, raw gas for external processing, NGL, LPG and C_5^+. Values used to calculate business revenue. Market values from the week of 15 to 21 of May 2021. 59

3.9 List of motocompressors, by process unit, accounted for in the estimation of total electricity demand. 60
3.10 List of furnaces, by process unit, accounted for in the estimation of total fuel gas consumption. 60
3.11 List of chemical products, by process unit, accounted for in the estimation of total chemical product spending. The average values reported utilize unit price data from a Petrobras internal report. Values are presented in $R\$$ per million cubic meters. 61
3.12 Typical unit costs of electricity and fuel gas for Brazilian industry. 62
3.13 Description of the three types of inequality constraints categorized by their distinct treatment during implementation. Table also details how each type was handled and which constraints fit in each category. 68

4.1 Molar composition and process conditions (temperature and pressure of the three simulation inlet streams, which correspond to the entrance of slug catchers SG-A/B/C). 73
4.2 Model validation results: comparison between simulated and actual product flowrates for monthly input data. 76
4.3 Initial estimate values of continuous decision variable u_1 to u_{12} for five distinguished sets (I01 to I05). 80
4.4 Execution time for each phase of the MINLP problem. 94
4.5 Tree-search results of the MINLP problem solution. Table shows the number of executions, respective tree node, values of the binary decision variables (y_1 to y_7, number of iterations and function evaluation of each NLP Nelder-Mead run.) 96
4.6 Optimization results obtained by the MINLP, PSO post-screening and Nelder-Mead refinement steps. Values of continuous and binary variables. 97
4.7 Potential profit gain resulting from the solution of NLP and MINLP problem formulations in comparison to a base case, defined as initial estimate I03 from Table 4.3. 100
4.8 Case study optimization results obtained by the MINLP, PSO post-screening and Nelder-Mead refinement phase. Values of continuous and binary variables considering a scenario of gas streams sales price four times higher than base case. 101

C.1 Mass balance of the natural gas processing site per NGPU. Values expressed in product recovery, calculated as a percentage of the NGPU feed gas flowrate. 118

C.2 Overall mass balance of the natural gas processing site. Values expressed in product recovery, calculated as a percentage of the overall raw gas input flowrate. 119

List of Symbols

K	NGPU design capacity, p. 63
N	Dimension of the NLP problem, p. 66
P	Pressure, p. 21
P_s	Saturation pressure, p. 26
Pen	Penalty function, p. 61
Pen_{conv}	Convergence penalty, p. 62
$Power_{EE}$	Electric power demand, p. 57
$Power_{FG_i}$	Power demand of fuel gas, p. 59
Q	Flowrate, p. 60
Q_i^{max}	Upper bound of continuous variables Q_i, p. 62
Q_i^{min}	Lower bound of continuous variables Q_i, p. 62
Q_{FG}	Fuel gas volumetric flowrate consumption, p. 59
Q_{NGPU}	Vector with the simulated feed flowrate, in m^3/d, of each NGPU that have associated chemical product spending., p. 59
R	Ideal gas constant, p. 21
S	Augmented objective function, p. 61
S_{ref}	Constant of reference utilized to adjust the order of magnitude of the objective function, p. 61
T	Temperature, p. 21
V^*	COSTALD parameter: characteristic volume, p. 27
$V_R^{(0)}$	COSTALD parameter: corresponding states function for normal fluids, p. 27

$V_R^{(\delta)}$ COSTALD parameter: corresponding states deviation function, p. 27

V_m Molar volume, p. 21

V_s COSTALD parameter: molar volume for saturated liquid, p. 26

X_{A_i} CPA parameter: fraction of sites A on molecule i that do not bond with other active sites., p. 23

$\Delta^{A_iB_j}$ CPA parameter: association strength between two sites belonging to two different molecules, p. 23

Ω PR parameter: coefficients of the attractive and co-volume parameters, p. 22

α PR parameter: temperature-dependent function in the attractive term, p. 21

$\bar{Q}_{CP}$ Average consumption of chemical product per gas feed, p. 60

$\bar{\bar{S}}$ Profit in R$/d, p. 60

$\bar{\eta}_i$ Efficiency of equipment i, p. 59

$\bar{\rho}$ Molar density, p. 23

$\bar{x}$ PSO parameter: particle position, p. 31

$\beta^{A_iB_j}$ CPA parameter: association volume, p. 23

δ User-defined step-size used to determine the initial polyhedron of Nelder-Mead method, p. 66

$\epsilon^{A_iB_j}$ CPA parameter: association energy, p. 23

η CPA parameter: reduced density, p. 24

γ Relative density of the gas, p. 6

κ PR parameter: acentric factor-dependent term, p. 22

ω PR parameter: acentric factor, p. 22

a Van der Waals attractive parameter, p. 21

a_c PR parameter: critical conditions function in the attractive term, p. 21

b	Van der Waals covolume, p. 21
c	Cost vector (fuel gas, electricity and chemical products), p. 60
c_1	PSO parameter: cognitive parameter, p. 31
c_2	PSO parameter: social parameter, p. 31
f_{LP}	Objective function of the LP problem, p. 68
g	CPA parameter: radial distribution, p. 23
g_j	Inequality constraints, p. 60
h_j	Equality constraints, p. 60
m	Market price vector for gas processing products, p. 60
n_g	Number of inequality constraints, p. 60
n_h	Number of equality constraints, p. 60
n_p	Number of products from gas processing complex (5), p. 60
r	Scale factor vector, p. 61, 62
sim	Nelder-Mead method initial polyhedron, p. 66
u	Continuous decision variables, p. 34
u_1	Raw gas flowrate for external processing, p. 36
u_1	Raw gas flowrate from SG-B to SG-A, p. 36
u_3	Raw gas bypass from the outlet of SG-C into the sales gas stream, p. 36
u_4	Raw gas from the outlet of SG-A to NGPU-E, p. 36
u_5	Raw gas from the outlet of SG-B to NGPU-E, p. 36
u_6	Raw gas from the outlet of CO_2 removal unit to NGPU-E, p. 36
u_7	Raw gas from the outlet of SG-C to NGPU-E, p. 36
u_8	Raw gas from the outlet of SG-A to NGPU-A, p. 36
u_9	Total NGPU-B feed flowrate, p. 36
u_{10}	Total NGPU-D feed flowrate, p. 36

u_{11} Fraction of the residual gas flowrate from NGPU-E that is reprocessed, p. 36

u_{12} Fraction of the residual gas flowrate from NGPU-A that is reprocessed, p. 36

v PSO parameter: particle velocity, p. 31

w PSO parameter: inertia weight, p. 31

w_0 Initial estimate vertex of Nelder-Mead method, p. 66

x Model variables, p. 34

x_i Molar fraction of component i, p. 7

y Binary decision variables, p. 34

y_1 Operating status of raw gas route for external processing, p. 36

y_2 Operating status of raw gas bypass, p. 36

y_3 Operating status of NGPU-E, p. 36

y_4 Operating status of NGPU-A, p. 36

y_5 Operating status of NGPU-B, p. 36

y_6 Operating status of NGPU-C, p. 36

y_7 Operating status of NGPU-D, p. 36

List of Abbreviations

T_r	Reduced temperature, p. 22
$\bar{c}$	Compound-specific constants for saturation pressure calculation, p. 27
n_{FG}	Number of fuel gas consumers, p. 59
ANP	Brazilian National Agency of Petroleum, Natural Gas and Biofuels, p. 5
COSTALD	Corresponding State Liquid Density, p. 25
CPA	Cubic Plus Association, p. 21
CP	Chemical Products, p. 57
EE	Electricity, p. 57
EoS	Equation of State, p. 20
FG	Fuel Gas, p. 57
GAMS	Commercial optimization software, p. 19
GERG	*Groupe Européen de Recherche Gaziere*, p. 21
GTL	Gas to Liquid, p. 17
HCDP	Hydrocarbon Dew Point, p. 6
HHV	Higher Heating Value, p. 6
HYSYS	Commercial Process Simulator, p. 18
IP	Integer Programming, p. 32
JT	Joule-Thomson, p. 11
LFU	Liquid Fractionation Unit, p. 7

LHV	Lower Heating Value, p. 59
LLE	Liquid-Liquid Equilibrium, p. 23
LNG	Liquefied Natural Gas, p. 17
LPG	Liquefied Petroleum Gas, p. 9
LP	Linear Programming, p. 67
MEG	Monoethylene Glycol, p. 13
MILP	Mixed Integer Linear Programming, p. 32
MINLP	Mixed Integer Non-Linear Programming, p. 18
MIP	Mixed Integer Programming, p. 34
MME	Brazilian Ministry of Mines and Energy, p. 1
MN	Methane Number, p. 6
MON	Motor Octane Number, p. 6
MR	Mechanical Refrigeration, p. 11
NGL	Natural Gas Liquid, p. 9
NGPU	Natural Gas Processing Unit, p. 2
NLP	Nonlinear Programming, p. 34
OAERAP	Outer Approximation Equality Relaxation Augmented Penalty, p. 20
OpEx	Operating Expenses, p. 56
PROSIM	Commercial Process Simulator, p. 19
PR	Peng Robinson, p. 20
PSO	Particle Swarm Optimization, p. 31
ProMax	Commercial Process Simulator, p. 19
RA	Refrigeration Absorption, p. 11
SG	Slug Catcher, p. 7
SRK	Soave Redlich-Kwong equation of state, p. 21

TE	Turbo-expansion, p. 11
UTGCAB	Cabiúnas Gas Processing Site, p. 35
VBA	Visual Basic for Applications, p. 34
VLE	Vapor-Liquid Equilibrium, p. 21
WDP	Water Dew Point, p. 6
WI	Wobbe Index, p. 6
gPROMS	Commercial process simulator, p. 20

Chapter 1

Introduction

1.1 Motivation

Since the discovery of oil in the offshore pre-salt layer, around 2007, Brazil has become an even more relevant worldwide player in oil extraction. Along with this oil comes a gas fraction that must be separated and held properly. Since total flaring is not an option anymore due to environmental impact, the part of this gas that cannot be reinjected in the reservoir must be sent onshore for market use, otherwise oil production might get compromised. For the next decade, gas volume available for processing and end-use in Brazil is expected to grow from roughly 80 to almost 150 millions of cubic meters per day (MME *et al.*, 2020), guaranteeing its importance in the market. Furthermore, natural gas is also getting increased relevance in the last years as a transition energy source from other fossil to renewable fuels. This, added to the fact that the perspective until 2030 is of growth in production, makes natural gas strategic to Brazil.

Between the wellhead and the transmission system, natural gas needs to be conditioned offshore to allow pipeline transmission and then further processed onshore to meet market specifications. These midstream processing plants are responsible for receiving natural gas from oil production sites, via gathering pipelines, and guaranteeing sales gas specifications for transportation and end use. To achieve that, they carry out processes for removal of contaminants and heavy liquid fractions. This intrinsic dependence between oil extraction and natural gas production makes gas processing plants key structures to guarantee Brazil's pre-salt oil production.

Since gas processing plants are in midstream, they inherently need to deal with dynamic changes in the inlet streams, due to varying upstream conditions. Typical changes include: flowrate, liquid slug, composition, pressure and level of contaminants. Furthermore, midstream layout is subject to long-term modifications in the upstream,

such as the opening of a new well, new process units, slug catchers and offsite facilities. Over time, this means that the single-unit natural gas processing plant might turn into a multi-unit and multi-feedstock gas processing site. Each single natural gas processing unit (NGPU) has its specificities, performance, maximum and minimum capacities, operating modes and even resulting products. Furthermore, natural gas from each entrance point arrives at different conditions, flowrates and compositions, needing particular levels of treating and processing to achieve sales gas specifications. This extremely dynamic scenario might lead to sub-optimal operating conditions in terms of performance and profit throughout the time (CAMPOS *et al.*, 2012).

Therefore, in this context of:

1. Perspective of growth in associated natural gas production;
2. Increased importance of natural gas as transition alternative from other fossil to clean fuels;
3. Imminent opening of Brazil gas market, allowing natural gas processing plants to receive feedstock from multiple contracting parties;
4. Increased complexity of natural gas processing plants due to higher demand. Natural gas processing sites with multiple distinguished units built over time.

There is an increasing importance in studying natural gas processing sites as integrated facilities, modeled and optimized as multiple-unit site, considering high-level economical and business aspects. Single-unit optimizations, although common in the literature, do not cover the needs for this upcoming industrial scenario and that is the context in which this work presents itself.

1.2 Objectives

The main objective of this work is to propose a strategy for simulation and optimization of a multi-unit and feedstock natural gas processing site from a high-level perspective, taking into account economic and business aspects, by using an actual industrial plant as case study.

Specific objectives are:

1. Build a rigorous digital static model in a process simulation software to virtually reproduce the behavior of the gas processing site;

2. Validate and tune the model with actual plant data;

3. Identify the main decision variables, constraints and objective function of the optimization problem;

4. Formulate and solve the NLP and MINLP optimization problems utilizing flexible polyhedron, particle swarm optimization and branch-and-bound methods in the open-source frameworks from Python;

5. Evaluate the optimized results and estimate potential profit gain.

1.3 Dissertation Structure

This work is divided into five chapters. After this brief introduction, the second chapter contains a literature review, focusing mainly on natural gas processing plants and studies on their optimization. Then, in the third chapter, the methodology applied in this work is detailed, including modeling approaches, the optimization problem formulation, economic considerations and their respective equations. In Chapter 4, the results of simulation and optimization are presented and discussed for the NLP and MINLP problems. At last, Chapter 5 brings the conclusions of the work and further suggestions.

Chapter 2

Literature review

This chapter will approach the state-of-the-art and key concepts that are essential for better understanding of this work. First, a description of natural gas fundamentals will be shown, followed by an overview of what are natural gas processing plants and their role in the Oil & Gas value chain. Then, a discussion is presented regarding previous works on modeling and optimization of NGPUs.

2.1 Natural Gas

Natural gas is a non-renewable mixture of hydrocarbons typically found in underground reservoirs, either in gas (non-associated) or oil (associated) wells (MME *et al.*, 2020; MONDAL *et al.*, 2013). Major components of natural gas are paraffin hydrocarbons up to 10 carbon atoms (C_1 to C_{10}) (CAMPBELL, 2013), mainly methane (major component), ethane, propane and butane and non-hydrocarbons nitrogen, carbon dioxide, water and hydrogen sulfide (MME *et al.*, 2020). In Brazil, 80 % percent of the natural gas produced is associated to oil extraction, coming offshore (MME *et al.*, 2020). As a consequence, oil industry shows a strong dependence of the gas industry.

Natural gas primary use is as fuel (MONDAL *et al.*, 2013). Even though it is fossil-based, natural gas has been drawing attention in the last few years as a transition fuel from oil to renewable energy, since it is considered the most energy-efficient fossil fuel in terms of emissions (MOKHATAB *et al.*, 2012; ZHENG *et al.*, 2010). Other uses include power generation, feedstock for the petrochemical industry and source of elemental sulfur (MOKHATAB *et al.*, 2012; MONDAL *et al.*, 2013). In Brazil, another relevant use of natural gas is as fuel in thermal power plants, which are requested to operate on demand, particularly when hydroelectric reservoir levels lower due to dry season (FRANCO *et al.*, 2020).

Between extraction and market use, natural gas goes through gathering, processing and transportation (CAMPBELL, 2013; POE and MOKHATAB, 2017). Although some field processing usually takes place near the wellhead, further processing is needed onshore to guarantee transport and market specification — that is where the importance of natural gas processing plants comes in.

2.2 Natural Gas Processing Plants

The natural gas extracted from wells first goes through offshore field processing. This treatment aims to condition the natural gas for exportation to shore. To achieve that, the level of H_2S and CO_2 contaminants are reduced to acceptable levels — pre-salt natural gas, for instance, might have its CO_2 mole content reduced from up to 40 % to around 3 % in this step (BIGDOLI, 2018) — and there are also units for gas dehydration, hydrocarbon dew point adjustment and lastly compression (BIDART *et al.*, 2015).

However conditioned, this export gas is still not suitable for market use and requires further processing — for this reason, and in accordance with the literature terminology, the natural gas exported after field conditioning will be hereafter called *raw gas*. This gas is then directed through pipelines to an onshore natural gas processing facility, usually installed in the vicinities of the production field. Those facilities are mainly designed to condition gas for sale, by further removal of contaminants, impurities and heavy hydrocarbons, in order to meet sales gas requirements of transmission, distribution and purchase (CAMPBELL, 2013; MAZUMDER and XU, 2020; MONDAL *et al.*, 2013; ZHANG *et al.*, 2016). The processes required for that are typically sweetening (removal of acid gases), dehydration, compression and hydrocarbon dew point control (MAZUMDER and XU, 2020) — similar processes from field conditioning, but requiring further refinement to meet market specifications.

2.2.1 Sales Gas Specification

In Brazil, sales gas must comply with the regulation of the Brazilian National Agency of Petroleum, Natural Gas and Biofuels (ANP), resolution RANP 16/2008 (ANP, 2008), which controls both composition and properties of natural gas. The main items of specification are listed in Table 2.1.

The so called *heavy hydrocarbons* are the ones liquefied by the refrigeration process. Their presence in sales gas is undesirable for fuel application, because heavy hydrocarbons might liquefy in such conditions and burners are specifically designed to handle

Table 2.1: Main items of Brazil's sales gas specification, extracted from resolution RANP 16/2008 (ANP, 2008). Values refer to limits of southeast, south and center-west regions.

Type	Characteristic	Specification
Composition	Methane	$\geq$ 85.0 $mol.\ \%$
	Ethane	$\leq$ 12.0 $mol.\ \%$
	Propane	$\leq$ 6.0 $mol.\ \%$
	Butanes and heavier	$\leq$ 3.0 $mol.\ \%$
	CO_2	$\leq$ 3.0 $mol.\ \%$
	Inerts (CO_2 + N_2)	$\leq$ 6.0 $mol.\ \%$
	H_2S	$\leq$ 10 mg/m^{3*}
	Total sulfur	$\leq$ 70 mg/m^{3*}
Property	Higher Heating Value (HHV)	$\geq$ 35000 and $\leq$ 43000 kJ/m^{3*}
	Wobbe Index (WI)	$\geq$ 46500 and $\leq$ 53500 kJ/m^{3*}
	Methane Number (MN)	$\geq$ 65
	Water Dew Point (WDP) at 1 atm	$\leq -45\ ^oC$
	Hydrocarbon Dew Point (HCDP) at 4.5 MPa	$\leq$ 0 ^oC

* volumetric-based properties are referred at 20 ^{o}C and 1 atm in dry basis

gas phase. They are typically the fractions with molar mass higher than ethane (C_2^+) or propane (C_3^+), depending on the minimum temperatures reached during refrigeration. The heavier fraction (usually C_5^+) must necessarily be removed from raw gas, in order to meet hydrocarbon dew point (HCDP) specification. Further removal (C_2^+ or C_3^+) is motivated by economic reasons: pure natural gas liquid products usually have higher sales price than its value as sales gas (CAMPBELL, 2013). In terms of sales gas specification (RANP 16/2008), heavy hydrocarbons have specific segregated composition limits (namely ethane, propane and butane and higher) and will also interfere in regulated properties such as higher heating value (HHV), Wobbe Index and HCDP.

HHV is regulated to assure compatibility with customers' combustion equipment (CAMPBELL, 2013). It corresponds to the heat produced during gas combustion with theoretical amount of air, when the water formed is condensed at the reference temperature of 293.15 K, pressure of 101.325 kPa and dry basis (ANP, 2008).

Wobbe Index is a parameter additional to HHV, used for gas containing inerts such as N_2 and CO_2 as a measure of fuel interchangeability (CAMPBELL, 2013; POE and MOKHATAB, 2017). It is defined by the quotient between HHV and the square root of

the gas relative density (γ), measured at the same temperature and pressure:

$$WI = \frac{HHV}{\sqrt{\gamma}} \tag{2.1}$$

Lastly, Methane Number (MN) is useful for vehicle application, indicating the gas antidetonating capacity. This property is calculated from gas composition (Equation 2.2), using gasoline parameter MON (Motor Octane Number) as intermediate calculation (Equation 2.3).

$$MN = 1.445\ MON - 103.42 \tag{2.2}$$

in which MON is given by the following equation in terms of the molar fractions of the light gas compounds:

$$\begin{aligned} MON = {} & 137.78\ x_{methane} + 29.948\ x_{ethane} - 18.193\ x_{propane} \\ & - 167.062\ x_{butane} + 181.233\ x_{CO_2} + 26.994\ x_{N_2} \end{aligned} \tag{2.3}$$

2.2.2 Contaminants

The raw natural gas stream might carry several contaminants from the well to the processing facility. Those often need to be removed or at least reduced to meet gas sales requirements. They usually have deleterious effects either in terms of material (corrosion) or safety integrity. Table 2.2 provides a list of main natural gas contaminants, their associated problems and typical removal processes.

2.2.3 The Process

The natural gas processing facility is constituted by a series of unit operations in order to deliver products to the market, namely sales gas and natural gas liquids. The main equipment used is: two- and three-phase separation vessels, heat exchangers, compressors, pumps and distillation towers (FRANCO *et al.*, 2020).

The feed stream of a natural gas processing facility is a mixture of raw natural gas, condensate, water and contaminants. To handle this mixture, the feed stream first encounters an inlet liquid separation facility — the slug catcher (SG) — that roughly separates free water, liquid hydrocarbons and gas phase. Downstream to the slug catcher are NGPUs

Table 2.2: Common contaminants in raw natural gas, reasons for removal and typical separation processes. Adapted from MONDAL *et al.* (2013).

Contaminant	Reasons for removal	Typical removal processes
Water	• Pipeline corrosion (liquid water). • Solid hydrate formation and downstream blockage.	• Injection in cryogenic heat exchangers. • Molecular sieve dehydration, for higher removal.
CO_2	• Pipeline corrosion (in the presence of water). • Sales gas specification (molar percentage of CO_2 and HHV). • CO_2 freezing.	• Membrane separation. • Amine chemical absorption.
H_2S	• Pipeline corrosion (in the presence of water). • Sales gas specifications. • Health issues (toxic if burned).	• Amine chemical absorption. • Iron oxide-based adsorption.
Mercury	• Damage to aluminum heat exchangers. • Accumulation in glycol dehydration and amine regeneration vent gas. • Health issues (toxic).	• Adsorption with regenerative adsorbents. • Adsorption with non-regenerative adsorbents.

to handle gas phase and LFUs (Liquid Fractionation Units) to handle liquid fraction. A typical generic processing layout is shown in Figure 2.1 and contains three main process steps:

Gas treating: previous to gas processing, might be necessary in order to remove contaminants, such as water, acid gases (CO_2 and H_2S) and mercury (refer to Table 2.2 for list of contaminants, reasons for removal and typical processes).

Gas processing: use of refrigeration processes to liquefy and separate heavy hydrocarbons and water. Feed stream is the gas fraction that exits the slug catcher plus residual gas from liquid fractionation.

Liquid fractionation: use of distillation towers to separate liquid fractions into desired products. Feed stream is the liquid fraction that exits the slug catcher (condensate) plus liquid formed during gas processing refrigeration and in knockout drums.

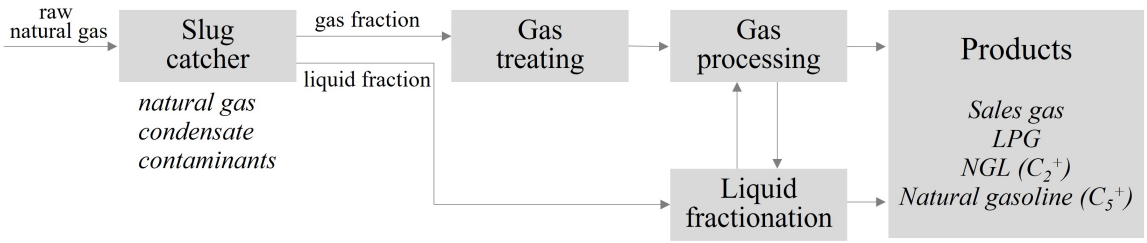

Figure 2.1: Schematic of a typical processing layout from a natural gas processing plant. Created by the author.

The actual products of natural gas processing and their properties will depend on the units' topology, process operating conditions and raw gas composition. Design will define processing philosophy, taking into account the gap between raw gas feed and desired products — the latter depend on available market, end-user applications and economic evaluation. This will define the design requirements for treating and processing facilities, plant complexity, the refrigeration mechanism and which components will need to be recovered from the rich gas (POE and MOKHATAB, 2017). A typical product profile of an NGPU — which is also the one that will be considered for this work — is shown in Table 2.3.

The operation of NGPUs is very peculiar when compared to other industries, even in the Oil & Gas chain. For being in midstream, NGPUs have the main primary goal of guaranteeing total destination of raw gas exported from upstream platforms. Therefore, there is no direct decision power over an NGPU feed — as long as it is within design limits, raw gas flowrate and composition might and do vary significantly within days or months.

For this reason, utilization of the plant might vary from days in which all NGPUs of the natural gas processing complex are in maximum capacity to others with underutilization. MOKHATAB *et al.* (2012) points out that operation agility is essential in this dynamic

Table 2.3: Typical products of a natural gas processing plant.

Market product type	Product	Main usages
Final	Sales Gas	Energy source for heating, cooking and electricity generation; vehicle fuel.
	Liquefied Petroleum Gas (LPG)	Energy source for cooking, fireplaces and heater.
Intermediate	Natural Gas Liquid (NGL)	Raw material for ethane, propane, butane and gasoline production
	Natural Gasoline (C_5^+)	Raw material for gasoline production, after stabilization.

scenario.

2.2.4 Liquid Recovery Processes

The previous section discussed the reasons for heavy hydrocarbons removal, namely: 1) sales gas specification and 2) economic advantage due to higher sales price of liquid products in comparison with sales gas. Figure 2.2 illustrates the different recovery-levels of natural gas liquids that can be achieved by an NGPU.

The level of liquid extraction is defined during plant design and is guided by economics: the higher the recovery, the higher the quantity of liquid products. However, there are also associated costs with increasing NGL extraction, since it requires more efficient processes — higher capital costs and operating expenses (mainly energy consumption). Therefore market value difference between liquid products and sales gas must be enough to cover this cost increase (CAMPBELL, 2013).

Extraction of natural gas liquids is usually performed by a refrigeration process. Each technology reaches certain minimum temperature levels for the raw gas stream, promoting an associated liquefaction. The lower the temperatures, the higher is the liquid recovery

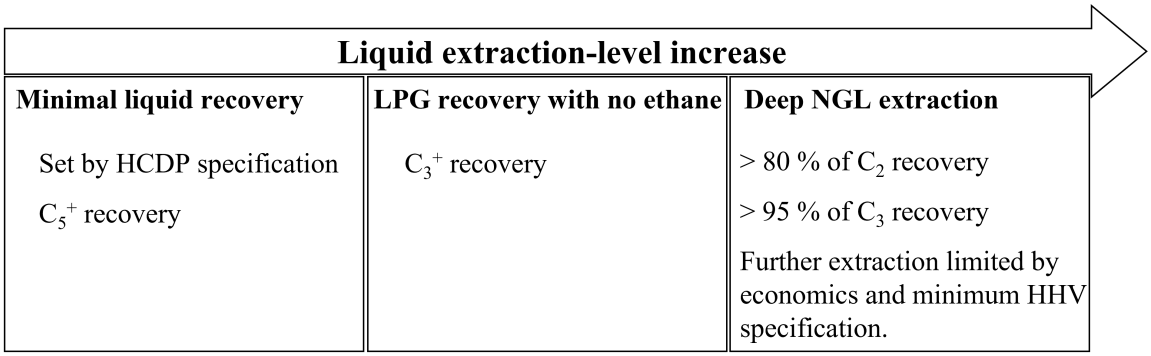

Figure 2.2: Ranges of liquid extraction in natural gas processing facilities. Adapted from CAMPBELL (2013).

and methane content in sales gas. There are four main refrigeration processes employed in NGPUs. They are, in ascendant order of complexity (CAMPBELL, 2013):

Joule-Thomson (JT): cooling through isenthalpic expansion in a valve. Works for HCDP control of non-associated gas. It is simple and low-cost, but requires high pressure difference availability.

Mechanical Refrigeration (MR): use of an external refrigerant (the most common being propane) to cool down the gas. Used both for HCDP control and further liquid recovery. Minimum temperature is limited by the atmospheric boiling point of the refrigerant ($-40\ ^oC$ for propane). Illustrated in Figure 2.3.

Turbo-expansion (TE): cooling via isoentropic expansion in a turbine (Figure 2.4). Reaches lower temperature when compared to the same pressure drop in a valve (JT). Minimum temperatures achieved are below $-100\ ^oC$, therefore this process is the most used for deep liquid extraction.

Refrigeration absorption (RA): use of hydrocarbon solvent to absorb heavy hydrocarbons at low temperatures. Used for higher recoveries, however presents complex operation. It is rarely used nowadays — replaced by TE.

Definition of the most adequate refrigeration technology is a multi-discipline design choice, which considers technical options, characteristics of expected raw gas feed (C_3^+ content) together with market and logistic information (POE and MOKHATAB, 2017).

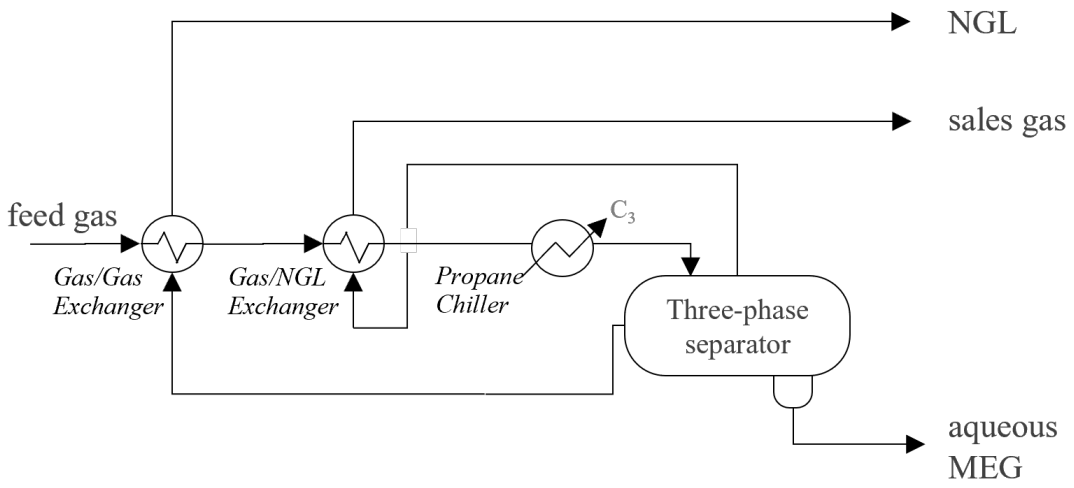

Figure 2.3: Schematic representation of a typical mechanical refrigeration gas cooling with propane. Created by the author based on POE and MOKHATAB (2017).

Design might consider one of those or a combination of more than one technology for a same NGPU. A natural gas processing site might be made up by NGPUs using different refrigeration technologies among one another.

2.2.5 Natural Gas Processing Plant Configurations

Slug Catcher

The slug catcher is the inlet separation facility for plant raw gas and the first unit that the feed stream encounters (POE and MOKHATAB, 2017). It works as a three-phase separation equipment to capture and buffer liquid slugs, segregating the raw gas into the fractions:

1. **Gas hydrocarbon:** inlet stream of gas treating and gas processing units.
2. **Liquid hydrocarbon (condensate):** inlet stream of liquid stabilization/fractionation units.
3. **Aqueous phase:** directed to wastewater treatment for contaminant removal.

Traditionally, slug catchers are of finger-type, with multiple long pieces of pipes. They allow high-pressure operation and provide surge volume for the uncertain incoming liquid flow.

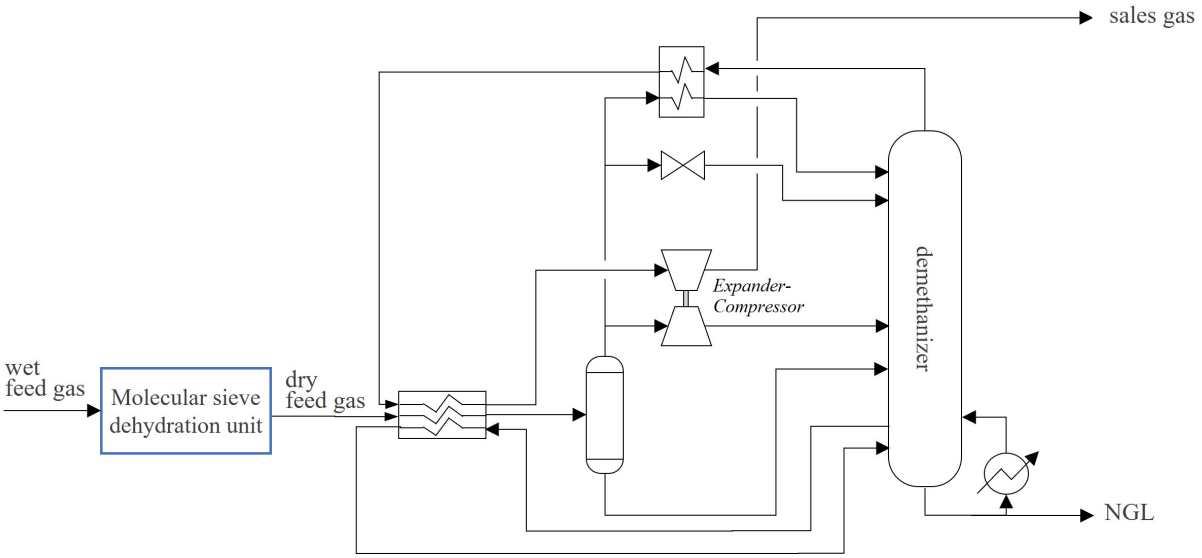

Figure 2.4: Schematic representation of a typical turboexpander refrigeration gas. Created by the author.

Gas Dehydration

In onshore natural gas processing facilities, there are two main processes for water removal (CAMPBELL, 2013; POE and MOKHATAB, 2017):

Inhibition by MEG injection: water is removed by temperature decrease during gas cooling and hydrate formation is prevented by inline monoethylene glycol (MEG) injection upstream to gas cooling heat exchangers, which works as a thermodynamic inhibitor, lowering hydrate formation temperature. Aqueous phase, containing MEG, is separated in a three-phase vessel, sent to MEG regeneration and reinjection.

Adsorption with molecular sieve: water is removed by adsorption, when wet gas flows through vessels filled with molecular sieve. Outlet gas reaches extremely low water concentrations (< 1 ppmv), therefore this is the most used process in plants with deep natural gas liquid recovery (Figure 2.4).

Choosing between them depends on the minimum temperatures reached during the gas cooling process: MEG injection is simpler and more economical, however molecular sieve performs a deep dehydration, leading to lower outlet water contents, which is necessary for higher liquid recovery (POE and MOKHATAB, 2017).

CO_2 Removal

Natural gas often contains some amount of acid gases such as CO_2 or H_2S. They might need to be removed from gas stream either for operational, economical or environmental reasons (DOS SANTOS *et al.*, 2015a; SELVAN and PANDA, 2018). In upstream facilities, part of CO_2 content is removed to avoid offshore pipeline corrosion. In natural gas processing plants, further CO_2 removal might be necessary when very low temperatures are achieved for liquid recovery — for example, when turboexpander refrigeration is used in the NGPU. In those cases, even small amounts of CO_2 might freeze due to the low temperatures, causing clogging and obstruction in the processes downstream.

Technologies for acid gas removal include adsorption, membrane retention, cryogenic separation, physical absorption and chemical absorption (SELVAN and PANDA, 2018; SULEMAN *et al.*, 2015). Selection depends on factors such as the concentration of CO_2 in the feed stream, level of removal required, plant size and temperature/pressure conditions (selection guidelines are available on SULEMAN *et al.* (2015)).

Chemical absorption in alkanolamine-based solvents is one of the most commonly used and well-established processes for CO_2 removal (DOS SANTOS *et al.*, 2015a). Its preferential application for natural gas streams in onshore processing facilities is justified for being suitable for gas streams with low CO_2 concentrations, having lower overall costs, greater process efficiency and ease in operation (SULEMAN *et al.*, 2015).

In this process, the CO_2 in the gas stream reacts with the alkanolamine in the aqueous solution forming weak salts (NUCHITPRASITTICHAI and CREMASCHI, 2011); the reaction is exothermic and reversible, being carried out in a counter current gas-liquid absorber at low temperature and high pressure (NUCHITPRASITTICHAI and CREMASCHI, 2011). Alkanolamine and CO_2 are then separated in a second step called regeneration, in which the solution is stripped at low pressure and high temperature, dissociating the weak salts back into alkanolamine and acid gas (SULEMAN *et al.*, 2015).

Figure 2.5 shows a general schematic diagram of the process. First, sour gas stream is pre-heated to guarantee minimum temperature to prevent condensation inside the absorber. The stream is then fed at the bottom of an absorption column and flows counter current to the lean alkanolamine aqueous solution that is fed to the top. Sweet natural gas leaves from the top. From here, all the steps concern regeneration of the amine aqueous solution loaded with CO_2 — so called *rich amine*.

This stream leaves the column from the bottom and is sent to a vessel that separates

light hydrocarbons that might have been absorbed along with the CO_2. Rich amine is then pre-heated before entering the regeneration column. In this column the weak salts formed during absorption are reversed into lean amine and acid gas by increasing temperature and reducing pressure. From the top exits a sour gas stream that can be either vented to the atmosphere or used as input to other process units. From the bottom exits the lean amine stream that is then cooled down and sent back to the process in a closed loop. First, this hot stream is used to pre-heat the cold rich amine; it is then pumped back into the absorber column. Make up of lean amine solution is also possible and expected to compensate for normal evaporation losses in the regeneration column.

It is important to point out that, due to environmental reasons, atmosphere release of CO_2 streams such as the sour gas from alkanolamine regeneration is turning into a not-acceptable solution. This scenario is increasing the relevance of carbon capture technologies, as discussed in SILVA *et al.* (2021).

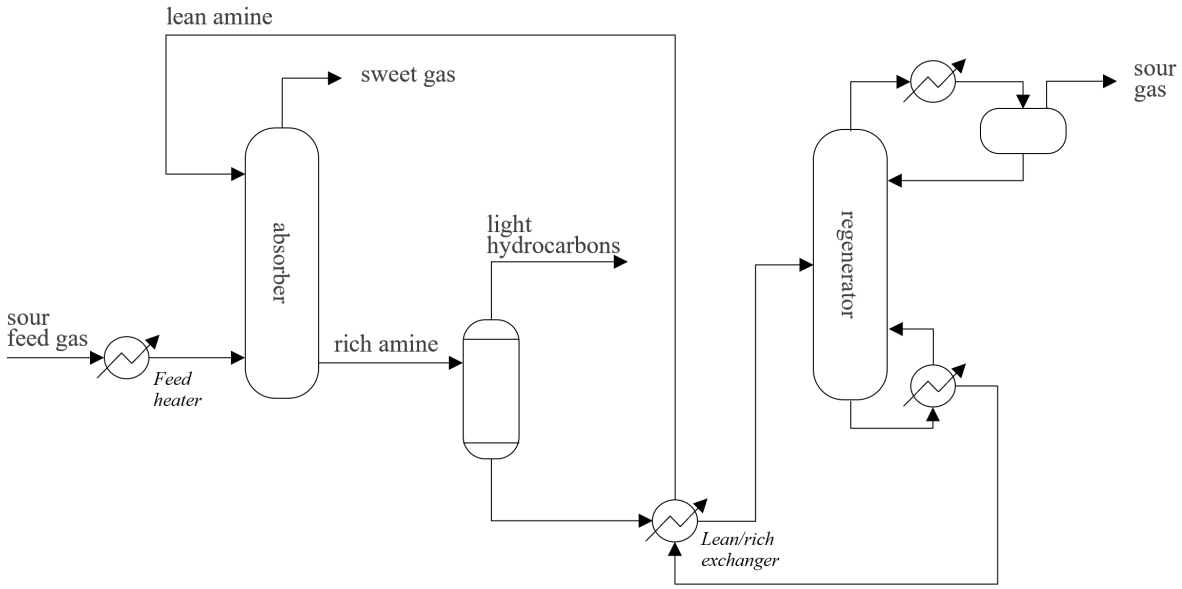

Figure 2.5: Schematic representation of a typical CO_2 removal plant by reactive absorption with alkanolamine solution. Created by the author.

Liquid Fractionating Units (LFUs)

The condensate fraction separated in the slug catcher needs further processing in order to remove dissolved light hydrocarbons. This stabilization is achieved in the inlet flash drum of an LFU, as shown in the typical schematic from Figure 2.6.

The stabilized condensate can still be fractionated by distillation to separate liquid fractions into distinguished products (CAMPBELL, 2013). For instance, LPG and C_5^+

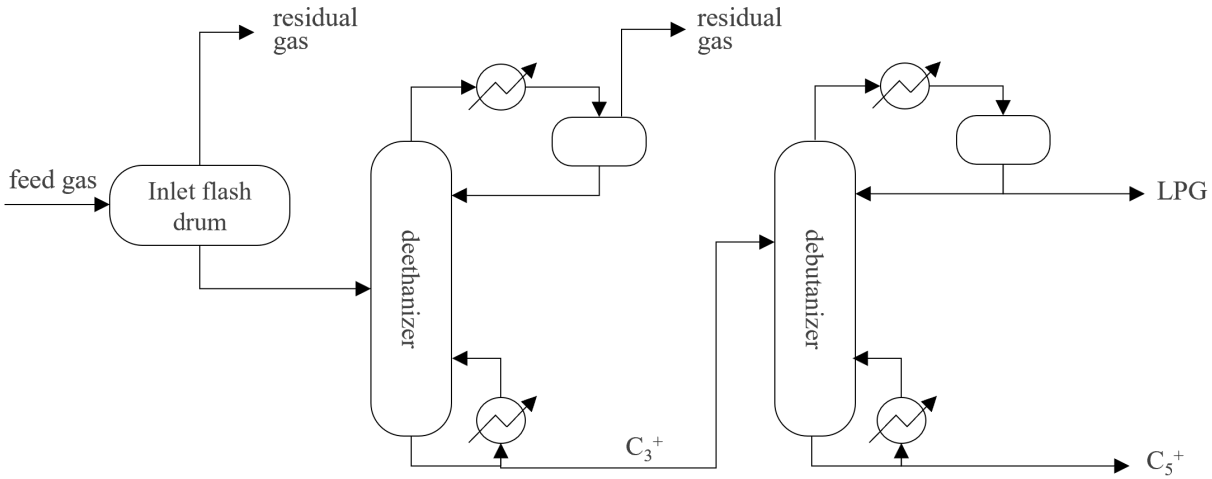

Figure 2.6: Schematic representation of a typical liquid fractionating unit with LPG and C_5^+ production. Created by the author.

production requires two distillation columns in series, deethanizer and debutanizer, vide Figure 2.6. In the deethanizer, most of ethane and lighter hydrocarbons are separated from the heavier hydrocarbons and exit at the top of the column in a stream called *residual gas*. This stream is analogous to the one that exits the top of the inlet flash drum and is composed by a non-specified gas fraction that is produced during liquid stabilization and fractionation. This residual gas has two possible destinations: 1) reprocessing in an NGPU or 2) direct mixing into the sales gas stream. The second option is limited by sales gas specification limits. The part of the residual gas stream that is sent for reprocessing in the NGPUs defines a relevant recycle stream in the plant, generating an intrinsic correlation between NGPUs and LFUs. It also increases the total amount of raw gas to be processed, also increasing liquid products yield.

Lastly, the debutanizer is the final distillation column from an LFU and separates LPG (top exit) from C_5^+ (bottom exit). The cut is usually defined by LPG sales specification in terms of heavy hydrocarbons content (C_5^+), which defines the operating condition of the column. The percentage of ethane in the LPG, on the other hand, is determined by the operating conditions of the first column, deethanizer.

2.3 Modeling and Optimization of Natural Gas Processing Units

Historically, NGPUs have been seen as destination facilities to associated raw gas, making oil extraction feasible. Process and economical optimization were not key parameters to

be looked at, as long as raw gas receivement capacity was guaranteed and all products left the NGPU specified. However, this scenario is changing since market is getting more competitive and demanding, and specifications more strict. Nowadays, onshore gas processing modeling and optimization is a field of growing attention, however, as ZHANG *et al.* (2016) and ZHENG *et al.* (2010) point out, most previous efforts on optimization are still limited to:

1. Natural gas production/processing — recovery enhancement.
2. Natural gas transportation.
3. Natural gas market.

Hence, natural gas processing optimization is generally associated only to liquid recovery enhancement within a single NGPU. The traditional mindset is that, the higher the liquid recovery, the higher the profit. However, as BULLIN and HALL (2000) points out, product price changes play an important role in gas processing plants optimization — the high frequency of those changes is such that maximizing liquid recovery does not always lead to the highest profit. On the other hand, optimization studies on gas transportation and market do not focus on processing and, when necessary, use overly simplified models to represent this step — mostly a set of pipelines, valves and compressors (ZHANG *et al.*, 2016).

Also, several papers on plant design of natural gas processing plants have been published on: optimization of refrigeration technology selection, plant topology, gas treating process technologies and so on. There are also many works that approach optimization from a process control perspective, proposing advanced control and real-time optimization techniques to operating variables, so to reduce variability, enable operation closer to process limits, and open opportunities for process optimization (CAMPOS *et al.*, 2012; DELOU *et al.*, 2021; RIBEIRO *et al.*, 2020).

From an economic perspective, most of previous works focused on improving plant energy efficiency, however looking at process variables such as temperatures and pressures of a single NGPU as manipulated variables. For instance, FRANCO *et al.* (2020) worked on energy consumption optimization for three different feed stream compositions varying in C_3^+ content; manipulated variables were such as temperatures (heat exchanger outlets and reboilers) and pressures (compressors, turboexpander and fractionating towers).

This interest comes from the fact that NGPUs are energy intensive. Both cold (raw gas cryogenic cooling) and hot (liquid fractionation) areas of a gas processing plant consume high amount of energy both as electricity and as fuel gas. Gas compression stations are

also energy intensive. Therefore, energy consumption is a key parameter to consider when attempting to enhance process efficiency in NGPUs (FRANCO *et al.*, 2020).

AL-SOBHI and ELKAMEL (2015) published a paper on optimization of gas processing with Liquefied Natural Gas (LNG), Gas to Liquid (GTL) and methanol facilities. This work reported the lack of the so called *enterprise-wide* optimization approach for this field in the literature. The work of BULLIN and HALL (2000) was one of the first in the literature that brought up the concept of optimizing an entire natural gas processing site with multiple feedstock, considering what they called *business aspects*, therefore looking at the plant from an integrated high-level perspective. That paper considered gas plants in a scenario of multiple inlet gas streams with different ownership, which is towards where Brazil is moving in the next few years with the approval of *New Gas Law* (AGENCY, 2020) and imminence of gas market opening with third-party access to natural gas processing facilities (further details on MOLNAR and BERKENWALD (2021)). They pointed out the importance of structured economic optimization in this scenario, in which intuitively recognizing optimum process conditions becomes nearly impossible due to the variety of raw gas feed streams and customer contracts.

Five years later, the previous work (BULLIN and HALL, 2000) was continued in BULLIN and CHIPPS (2005), with the development of an Excel interface to help combine process and economic models, in order to automatize plant economic performance prediction due to changes in process conditions. Later, ZHANG *et al.* (2016) proposed a superstructure to optimize a natural gas refining process by solving a Mixed Integer Non-linear Programming (MINLP) model for global optimality. Reformulation techniques were used to reduce the number of nonconvex terms and GAMS software was used for the optimization problem solving. The formulation and approach of ZHANG *et al.* (2016) is the closest one found in the literature from this work's proposal. However, on the contrary of present work, they perform a series of simplifications for model formulation — for instance, thermodynamic is not modeled rigorously and uses an empirical correlation instead —, do not use a process simulation software, and even though they consider a multi-unit and multi-feedstock natural gas processing complex, all NGPUs are identical and do not have different performances, which simplifies the problem.

Furthermore, optimization strategies on natural gas processing systems are reported to encounter difficulties for implementation due to nonlinear and nonconvex nature, which might make computation solutions hard to find (ZHENG *et al.*, 2010). According to ZHANG *et al.* (2016), thermodynamic modeling is responsible for most of those nonconvex nonlinearities, whereas SELOT *et al.* (2008) adds the existence of nonlinear pressure-flowrate relationships.

In all approaches, steady-state simulations are the default choice, unless a specific dynamic assessment is required. FRANCO *et al.* (2020) states that Aspen HYSYS ® (hereafter HYSYS) is the most employed software for modeling and optimization of NGPUs. Table 2.4 compiles several works in the area, listing the simulation approach utilized, their scope, the optimization problem considered and the method/software used for problem solution.

As far as this author's knowledge goes, very few attempts have been made into business economic optimization of an integrated natural gas processing site, considering multiple NGPUs and multiple feedstock with distinct inlet conditions. Most of the literature approaches mentioned in this section, although relevant to the natural gas processing knowledge area, do not take into account an integrated view of the whole natural gas processing site, with its distinguished NGPUs and multiple feedstock. Moreover, they do not look at the process from a high-level perspective, being limited to local NGPU optimization, without guaranteeing optimum feed allocation and the best economical operating point of the integrated facility. This constitutes a gap in the literature that this work aims to fulfill.

Table 2.4: Summary on approach of current literature work on simulation and optimization of NGPUs (continue on next page).

Reference	**Scope**	**Simulation approach**	**Optimization problem**	**Optimization software and method**
BULLIN and HALL (2000)	Single NGPU	Reduced-order model generated from PROSIM rigorous simulations	Profit maximization	Sensitivity analysis from case study simulations
BULLIN and CHIPPS (2005)	Single NGPU	ProMax ®	Profit maximization	Sensitivity analysis from case study simulations using Microsoft Excel integration

Table 2.4: Summary on approach of current literature work on simulation and optimization of NGPUs (continued from previous page).

SELOT *et al.* (2008)	Malaysia Sarawak Gas Production System (3 LNG plants)	Simplified first principles process models (no commercial simulator)	Raw gas feed maximization or well production maximization (production planning)	GAMS (BARON solver)
MONDAL *et al.* (2013)	Bangladesh industrial single NGPU	HYSYS v3.2	Profit maximization	Sensitivity analysis from case study simulations.
ZHANG *et al.* (2016)	Multi-feedstock gas processing site from South China	Simplified first principles process models (no commercial simulator)	Maximize profit (global optimization for MINLP)	GAMS 24.5.3 (ANTIGONE, BARON and LINDOGLOBAL solvers)
GONG and YOU (2017)	Shale gas processing unit	HYSYS	Net present value maximization, global warming potential minimization, water footprint minimization	GAMS 24.8.3 (CPLEX and BARON/SCIP solvers)
FRANCO *et al.* (2020)	Single NGPU	HYSYS v9	Minimize energy consumption	Aspentech Optimizer built-in tool
MAZUMDER and XU (2020)	Single NGPU with gas treating and compression	HYSYS v10	Total NGPU cost	Sensitivity analysis from simulations plus economic evaluations with Aspen process economic analyzer
MURALI *et al.* (2020)	Single NGPU	gPROMS process builder	MINLP to maximize profit	gPROMS (OAERAP solver)

Next subsection presents a literature review on the thermodynamic model that will be used in this work. Further details on the implementation will be discussed in Chapter 3.

2.3.1 Thermodynamic Model

When talking about gas processing simulations, Equations of State (EoS) are the primary choice for thermodynamic modeling (KONTOGEORGIS and FOLAS, 2009). Those have become popular for their simplicity and sufficient accuracy for this application, which needs a model that handles well vapor-liquid equilibrium calculations for a mixture of hydrocarbons up to ten (C_{10}) or to twenty (C_{20}) carbons plus water, CO_2, N_2 and H_2S at high pressure and low temperature. Typical temperature ranges between $-100\ ^oC$ and $250\ ^oC$, with pressures up to $100\ bar$. Cubic Equations of State are the classical models for this scenario: high-pressure plus hydrocarbon mixture (KONTOGEORGIS and FOLAS, 2009). Among them, the Soave Redlich-Kwong (SRK) and the Peng-Robinson (PR) EoS are the primary choice for gas processing (KONTOGEORGIS and FOLAS, 2009).

Works such as from FRANCO *et al.* (2020), GONG and YOU (2017) and MAZUMDER and XU (2020) used Peng Robinson (PR) EoS for gas processing. FRANCO *et al.* (2020) list simplicity and good performance for vapor-liquid equilibrium (VLE) prediction as reasons for its popularity and successful use. According to LOPEZ-ECHEVERRY *et al.* (2017), it is actually considered a fine cubic EoS for VLE calculations and determination of thermodynamic and volumetric properties for the problem studied in this work and will be further discussed in the following Subsection.

Another relevant EoS in the literature is the Peng Robinson modification proposed by the GERG (*Groupe Européen de Recherche Gaziere*) group (ISO, 2004), which focuses on the relationship between water content and water dew point of natural gas at high pressures (DOS SANTOS *et al.*, 2015b). Another approach to account for the presence of water was utilized by the Cubic Plus Association (CPA) EoS and considers the hydrogen bonds present in these systems. The CPA EoS, proposed by KONTOGEORGIS *et al.* (1996), considers the effects of chemical association and, therefore, was considered in this work to be compared with Peng Robinson. CPA will be presented later in this Section and the comparison with Peng Robinson EoS will be discussed in Section 3.2.4.

Peng Robinson EoS

The PR EoS was originally introduced in 1976 in the work of PENG and ROBINSON (1976), which was the result of a project for the Natural Gas Processors Association. At the time, the goal was to find an improved expression of the Soave-Redlich-Kwong model

in terms of predictive capability, due to SRK inaccuracy for liquid density calculations and property prediction near the critical points, as pointed by LOPEZ-ECHEVERRY *et al.* (2017) review paper. This resulted in a new two-parameter EoS described by the following set of equations:

$$P = \frac{RT}{V_m - b} - \frac{a}{V_m(V_m + b) + b(V_m - b)} \tag{2.4}$$

with the attractive parameter a given by:

$$a = a_c \alpha \tag{2.5}$$

In Equation 2.5, α is a temperature-dependent parameter, calculated by Equation 2.6.

$$\alpha = [1 + \kappa(1 - \sqrt{T_r})]^2 \tag{2.6}$$

where T_r is the reduced temperature and κ is a parameter dependent of the acentric factor (ω) and calculated from Equation 2.7.

$$\kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2 \tag{2.7}$$

Also in Equation 2.5, a_c is a parameter calculated from the critical temperature (T_c) and pressure (P_c), Equation 2.8.

$$a_c = \Omega_a \frac{R^2 T_c^2}{P_c} = 0.45724 \frac{R^2 T_c^2}{P_c} \tag{2.8}$$

Lastly, the co-volume parameter b from Equation 2.4 is also calculated from the critical properties, as stated in Equation 2.9.

$$b = \Omega_b \frac{RT_c}{P_c} = 0.07780 \frac{RT_c}{P_c} \tag{2.9}$$

Throughout the years, since it was published, many modifications of the PR EoS have been proposed with different objectives. LOPEZ-ECHEVERRY *et al.* (2017) compiles several of those and categorizes them into four types of modification: (1) introduction of new expressions for α, a or b parameters, (2) proposal of new deviation functions, (3) incorporation of new terms/parameters, and (4) modification of mixing/combining rules.

Cubic Plus Association EoS

The CPA equation of state was introduced as a new approach to describe the behavior of systems containing associating fluids. Those are species that form hydrogen bonds, either between the same molecules (self-association) or between different molecules (cross-association). These strong interactions interfere greatly on the thermodynamic properties of the fluids by forming molecular clusters within the system. Many attempts have been made throughout the years to model associating fluids (ECONOMOU and DONOHUE, 1996) using different theories to account for hydrogen bonding effect: Chemical, Perturbation and Lattice/Quasi-Chemical models. All of those are essentially built up of two contributions: 1) a physical term (cubic or noncubic EoS) plus 2) an association term, being the latter the crucial part in describing associating compounds (PESCHEL and WENZEL, 1984).

In this context, CPA EoS comes as a theoretically consistent model that can describe both VLE and LLE of one given component in a multicomponent system considering the same parametric set. As seen in Equation 2.10 for mixtures, CPA combines a simple cubic equation of state (originally SRK) to account for the physical term plus an advanced association term from Perturbation Theory, proposed by Wertheim (KONTOGEORGIS and FOLAS, 2009; SULEMAN *et al.*, 2015). In the absence of hydrogen bonds, CPA model reduces to SRK, achieving a good balance between accuracy and simplicity without adding significant computation time. CPA expression in terms of pressure is presented in Equation 2.10.

$$P = \frac{RT}{V_m - b} - \frac{a(T)}{V_m(V_m + b)} - \frac{1}{2}\frac{RT}{V_m}\left(1 + \bar{\rho}\frac{\partial \ln g}{\partial \bar{\rho}}\right)\sum_i x_i \sum_{A_i}(1 - X_{A_i}) \quad (2.10)$$

in which V_m is the molar volume, $\bar{\rho} = 1/V_m$ is the molar density, x_i is the mole fraction of component i. The first two terms refer to SRK contribution, whereas the last term accounts for association effects. X_{A_i} is the key element of that association term and represents the fraction of sites A on molecule i that do not form bonds with other active sites. It is calculated from Equation 2.11.

$$X_{A_i} = \frac{1}{1 + \bar{\rho}\sum_j x_j \sum_{B_j} X_{B_j}\Delta^{A_i B_j}} \quad (2.11)$$

in which $\Delta^{A_i B_j}$ is the association strength between two sites belonging to two different

molecules (Equation 2.12).

$$\Delta^{A_iB_j} = g(\bar{\rho}) \left(\exp \frac{\epsilon^{A_iB_j}}{RT} - 1 \right) b_{ij} \beta^{A_iB_j} \tag{2.12}$$

in which $g(\bar{\rho})$ is the radial distribution, given by Equation 2.13.

$$g(\bar{\rho}) - \frac{1}{1 - 1.9\eta} \tag{2.13}$$

and η is the reduced density, given by Equation 2.14

$$\eta = \frac{1}{4} b \bar{\rho} \tag{2.14}$$

in which b is the temperature dependent covolume parameter, the same utilized in the Peng Robinson EoS. The energy parameter a is temperature dependent and calculated from Equation 2.15.

$$a(T) = a_0 \left(1 + c_1 \left(1 - \sqrt{T_r} \right) \right)^2 \tag{2.15}$$

in which $T_r = T/T_c$ is the reduced temperature and T_c is the critical temperature.

For multicomponent systems, calculations are performed using parameters estimated from binary experimental data. For the physical term, conventional mixing rules can be applied to determine a and b from Equations 2.16 and 2.17.

$$a = \sum_i \sum_j x_i x_j a_{ij} \tag{2.16}$$

$$b = \sum_i x_i b_i \tag{2.17}$$

where a_{ij} is given by Equation 2.18.

$$a_{ij} = \sqrt{a_i a_j} \left(1 - k_{ij} \right) \tag{2.18}$$

CPA EoS has five pure compound adjustable parameters. The first three (a_0, b, c_1) are well-known SRK-term parameters, whereas the last two ($\epsilon^{A_iB_j}$ and $\beta^{A_iB_j}$) are so

called association energy and association volume, respectively, which are only used for associating compounds. In physical terms, when using the combining rules proposed by KONTOGEORGIS *et al.* (1996), they are related to the enthalpy and entropy of hydrogen bond, respectively (KONTOGEORGIS and FOLAS, 2009).

For the association term, the parameters $\epsilon^{A_iB_j}$ and $\beta^{A_iB_j}$ are calculated from combining rules. For mixtures of two associating compounds, also called cross-associating mixtures, CR-1 (Equation 2.19) and Elliott's combining rule (ECR, Equation 2.20) are the most widely and successfully used.

$$\epsilon^{A_iB_j} = \frac{\epsilon^{A_iB_i} + \epsilon^{A_jB_j}}{2} \quad , \quad \beta^{A_iB_j} = \sqrt{\beta^{A_iB_i}\beta^{A_jB_j}} \tag{2.19}$$

$$\epsilon^{A_iB_j} = \frac{\epsilon^{A_iB_i} + \epsilon^{A_jB_j}}{2} \quad , \quad \beta^{A_iB_j} = \sqrt{\beta^{A_iB_i}\beta^{A_jB_j}}\frac{\sqrt{b_ib_j}}{b_{ij}} \tag{2.20}$$

For binaries, b_{ij} is expressed by Equation 2.21 when considering classical combining rules for mixtures.

$$b_{ij} = \frac{b_i + b_j}{2} \tag{2.21}$$

For a self-associating compound, the so-called *association scheme* represents the number and type of association sites and is related to its physical nature and expected form of self-association. This concept was firstly introduced by HUANG and RADOSZ (1991) and the choice of the association scheme for a given compound will determine the expression that will be used to calculate X_{A_i} and, therefore, is of crucial importance since it changes the functional form of CPA EoS (KONTOGEORGIS and FOLAS, 2009).

For water, extensive investigation has been done in the literature and 4C scheme is typically used (ABUNAHMAN *et al.*, 2020; HUANG and RADOSZ, 1991; KONTOGEORGIS and FOLAS, 2009). The same is true for glycols such as MEG. For alkanolamines, KONTOGEORGIS and FOLAS (2009) recommend 4C scheme, even though other association schemes have also been tested and proven to yield good results, such as 6A and 4D (KONTOGEORGIS and FOLAS, 2009). Section 3.2.4 shows a comparison between PR and CPA EoS for the system in study, supporting the thermodynamic model choice for the following steps of this work.

2.3.2 The COSTALD Method for Liquid Density

The COSTALD (Corresponding State Liquid Density) method was developed by HANKINSON and THOMSON (1979) and THOMSON *et al.* (1982), for liquid density calculation. The work from 1979 proposed a correlation for saturated liquid density. In 1982, this was further expanded for compressed liquids.

The molar volume is calculated as a function of the liquid characteristic volume, reduced temperature and acentric factor (HANKINSON *et al.*, 1982). The set of equations that describe the COSTALD method is (HANKINSON and THOMSON, 1979; HANKINSON *et al.*, 1982):

$$V = V_s \left(1 - C \ln \frac{B + P}{B + P_s}\right) \tag{2.22}$$

with B given by:

$$\frac{B}{P_c} = -1 + a\,(1 - T_r)^{1/3} + b\,(1 - T_r)^{2/3} + d\,(1 - T_r) + e\,(1 - T_r)^{4/3} \tag{2.23}$$

The variable e is an exponential expression described by:

$$e = \exp\left(f + g\,\omega_{SRK} + h\,\omega_{SRK}^2\right) \tag{2.24}$$

Variable C is a linear function of acentric factor, calculated as:

$$C = j + k\,\omega_{SRK} \tag{2.25}$$

Constants a to k have the values given in Table 2.5.

Table 2.5: Constant values for COSTALD liquid density calculation. Taken from THOMSON *et al.* (1982)

Parameter	Value
a	-9.070217
b	62.45326
d	-135.1102
f	4.79594
g	0.250047
h	1.14188
j	0.0861488
k	0.0344483

In Equation 2.22, V_s is the saturated molar volume, determined by the correlations:

$$V_s = V^* V_R^{(0)} \left[1 - \omega_{SRK} V_R^{(\delta)}\right] \tag{2.26}$$

where V^* is COSTALD characteristic volume, a compound-specific parameter reported in HANKINSON and THOMSON (1979). $V_R^{(0)}$ is the corresponding states function for normal fluids and $V_R^{(\delta)}$ is the corresponding states deviation function. $V_R^{(0)}$ is given by:

$$\begin{aligned} &V_R^{(0)} = 1 + a\,(1 - T_r)^{1/3} + b\,(1 - T_r)^{2/3} + C\,(1 - T_r) + d\,(1 - T_r)^{4/3}, \\ &for\ 0.25 < T_r < 0.95 \end{aligned} \tag{2.27}$$

in which a, b and d are constants given in Table 2.5 and C is calculated from Equation 2.25. $V_R^{(\delta)}$ is then given by:

$$V_R^{(\delta)} = \frac{e + f(T_r) + gT_R^2 + hT_r^3}{T_r - 1.00001}, \qquad for\ 0.25 < T_r < 1.0 \tag{2.28}$$

in which e is the exponential expression described above and f, g and h are constants given in Table 2.5. Lastly, P_s is calculated as:

$$\ln P_s = \bar{c}_1 + \frac{\bar{c}_2}{T} + \bar{c}_3\, T + \bar{c}_4 \ln T \tag{2.29}$$

In the equation above, $\bar{c}_1$, $\bar{c}_2$, $\bar{c}_3$ and $\bar{c}_4$ are specific constants for each compound. For

multi-component systems, mixing rules are available in HANKINSON and THOMSON (1979).

This COSTALD method is widely used in the gas industry, being recognized by CAMPBELL (2013) as one of the most general and accurate methods for liquid density prediction with specified composition, presenting errors lower than 1 %. In conclusion, the combination of PR EoS (for VLE calculations) and COSTALD (for liquid density calculations) presents itself as an accurate and simple approach, object of interest of the natural gas industry. For those reasons, this combination was selected to be evaluated in this work.

2.3.3 The Inside-Out Method for Distillation

Solving a distillation column requires the calculation of temperature, pressure, flowrates, compositions and heat transfer at each stage. This requires the solution of mass balance, phase equilibrium, mole fraction summation and energy balance, the so-called MESH equations (SEADER *et al.*, 2011). Since those relations are nonlinear algebraic equations with strong interaction between one another, a numerical solution procedure is required. From the methods available, the bubble-point and sum-rates methods are simpler and restricted to nearly ideal mixtures (SEADER *et al.*, 2011). On the other hand, Newton-Raphson (simultaneous solution of the MESH equations) and Inside-Out methods are more complex, robust and flexible, being suitable for nonideal sistems and widely available in process simulators (SEADER *et al.*, 2011).

Inside-Out is a numerical method for solving steady-state multicomponent multistage non-ideal distillation column problems (BANSAL and MANJARE, 2016). It was first proposed by Boston and Sullivan as an alternative to reduce the time needed to compute thermodynamic properties during calculations (SEADER *et al.*, 2011). The method is characterized by dividing the problem in two loops: inner and outer. The inner loop is solved more frequently and therefore uses a more simplified and faster approach: empirical equations are used to converge the calculations along with MESH equations. The outer loop is demanded less often and therefore is more rigorous. There is also a previous initialization step procedure, which goal is to provide estimates of liquid/vapor component mole fractions, temperature and liquid/vapor flowrates (BANSAL and MANJARE, 2016; SEADER *et al.*, 2011). Simpler methods such as bubble-point or sum-rates might be used in this step (BANSAL and MANJARE, 2016).

Since its development in 1974, the Inside-Out has been extensively used and is the default method in HYSYS to solve distillation columns (TECHNOLOGY,

2019), due to its robustness and agility. Hence, it was chosen for use in this study. Figure 2.7 shows a simplified flowchart of Inside-Out method, highlighting inner and outer loops.

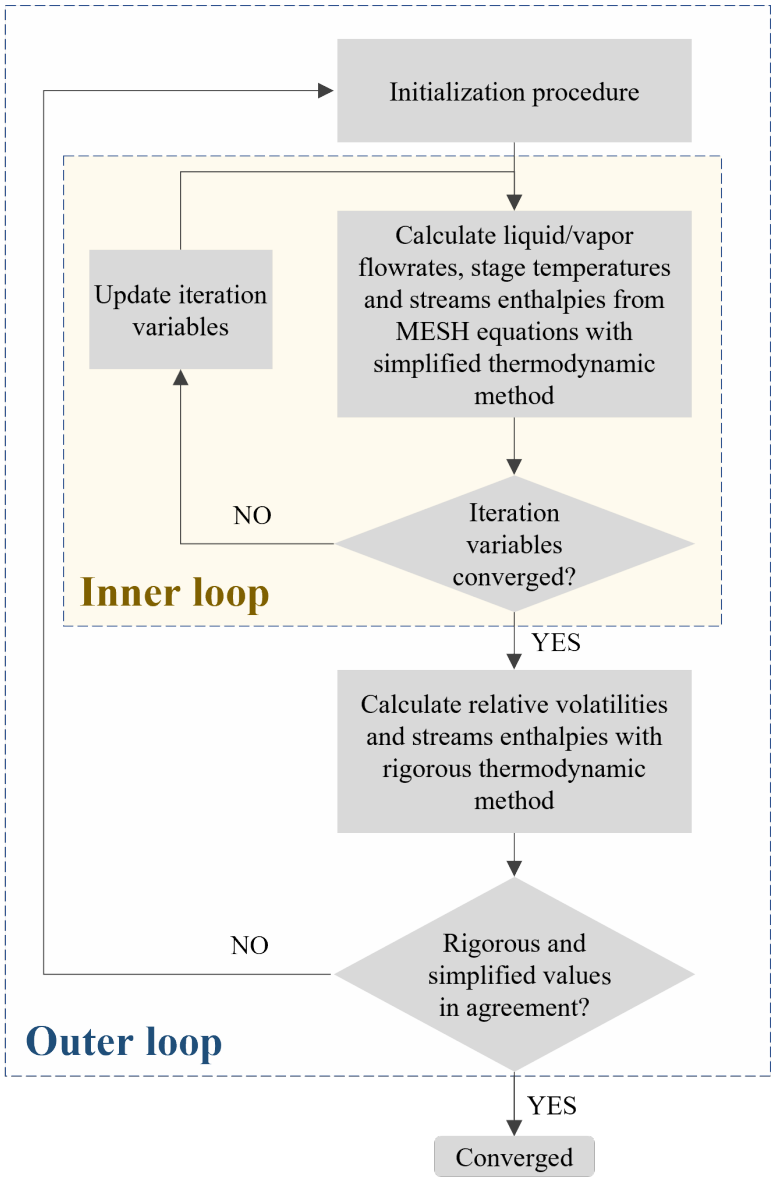

Figure 2.7: Simplified flowchart of Inside-Out solving method for multistage multicomponent distillation. Adapted from SEADER *et al.* (2011) and BANSAL and MANJARE (2016).

2.3.4 Process Optimization Strategies

As BIEGLER (2010) points out, carrying out an optimization procedure that couples the optimization algorithm with a simulation model developed in a commercial process simulator is challenging and requires more than just putting these two elements together. It is necessary to ensure other software functions such as: communication links for data

transfer, estimation of process parameters, data processing/reconciliation/validation and a database for process variables, operating conditions, disturbances and costs.

In this simulation-optimization framework, the simulation model contains the bulk of the constraints, represented by a set of equality constraints, for instance: mathematical relations (material and energy balances), phase relations, connection variables and rate equations. In some cases, the simulation model might also account for optimization binary variables, by alternating flowsheet topography, changing parameters or modifying specific stream flowrates (on-off) (BIEGLER, 2010). On the other hand, the inequality constraints are usually not directly accounted for in the process simulator, and include material flow limits, upper and lower bounds of temperatures, pressures and compositions.

In this context, there are two major types of optimization algorithms for NLP (BIEGLER, 2010): **1)** ***feasible path***, which satisfies equality and active inequality constraints at the end of every calculation step. This approach might require higher computation time, however ensures feasible solution even in case of non-convergence. On the other hand, **2)** ***infeasible path*** only satisfies equality and inequality active constraints when optimal solution is reached.

In this work, the feasible path approach will be utilized. The equality constraints will be solved by the simulation model, developed in a sequential-modular process simulator (HYSYS). The inequality constraints will be accounted for by the optimization algorithm, ensuring that at every intermediate stage both equality and inequality constraints are satisfied.

2.3.5 Nonlinear Local Optimization: Nelder-Mead Flexible Polyhedron Method

The Nelder-Mead algorithm was firstly introduced in 1965 by NELDER and MEAD (1965) and has become widely utilized since then. It is a direct search method for unconstrained multidimensional local optimization. The method was developed for nonlinear optimization and, as a direct search method, does not require derivative information on the objective function (BARKER and CONWAY, 2007).

This method attempts to find local minimum for an n-dimensional problem using a flexible polyhedron, a geometric figure of $(N + 1)$ vertices that can be modified at each iteration according to the pre-established criteria. At each iteration, the objective function is evaluated at each vertex and the result is used to determine search direction in an attempt

to improve the worst vertex of the polyhedron (BARKER and CONWAY, 2007), guided by 4 processes, as follows:

1. **Reflection:** the algorithm replaces the worst vertex — corresponding to the highest values of the objective function in a minimization problem — by another co-linear to it and to the remaining polyhedron centroid. By definition, this operation does not modify the size of the original polyhedron.

2. **Expansion:** occurs when the algorithm finds a promising direction, i.e., when the new points resulting from reflection is better than the best vertex from the original polyhedron.

3. **Contraction:** has the opposite idea of expansion and is performed when the new reflection point lays between the worst and the second worst vertices from the original polyhedron.

4. **Reduction:** used when the new reflected point is worse than the worst vertex from the original polyhedron. In this case, the size of the polyhedron reduces, maintaining only the best original vertex.

The deformation ability of the polyhedron improves rate of convergence, allowing the polyhedron to adapt to the objective function topography (BARKER and CONWAY, 2007).

Convergence criterion is associated with distance between vertices and centroid, i.e., the algorithm reaches convergence when the polyhedron becomes too small.

Direct search methods such as the Nelder-Mead algorithm might have more difficulty in converging, due to the absence of derivative information. However, as pointed out by BIEGLER (2010), in the last decade computational power increased significantly, and so did the need to couple simulation with optimization models. In this context, direct search approaches gained visibility, since they allow *black-box* optimization in association with complex simulation models developed in commercial process simulators and therefore presenting itself as a good choice for this work.

2.3.6 Nonlinear Global Optimization: Particle Swarm Method

Particle swarm optimization (PSO) is a non-deterministic global optimization algorithm proposed by KENNEDY and EBERHART (1995). Its wide use is in great part due to the efficient and simple concept that lies behind among the non-deterministic methods, as

well as the simplicity in implementing the computer code algorithm.

PSO treats every potential solution as an individual particle in a swarm population. Its basic concept is that the behavior of each individual particle inside the swarm can be modeled with simple vectors that take into account the influence of the best values of the single particle (cognition term) and of the whole population (social term) (LEE and EL-SHARKAWI, 2008; MORAES *et al.*, 2014). This means that each individual exchanges previous experiences with the whole group and updates its position ($\bar{x}$) according to Equation 2.30.

$$\bar{x}_i^{k+1} = \bar{x}_i^k + v_i^{k+1} \tag{2.30}$$

with v_i^{k+1} expressed by Equation 2.31.

$$v_i^{k+1} = w\, v_i^k + c_1\, rand_1[pbest_i - \bar{x}_i^k] + c_2\, rand_1[gbest - \bar{x}_i^k] \tag{2.31}$$

in which $\bar{x}_i$ and v_i are, respectively, the position and velocity of particle i, $pbset_i$ is the best position ever found by particle i, $gbest$ is the best value so far in the groups, among all $pbest_i$. c_1 and c_2 are, respectively, the cognitive and social parameters that account for the weight of each term relative to the other in the movement of the particle. $rand_1$ and $rand_2$ are random decimal numbers between 0 and 1 that are generated at each iteration for every particle. w is the inertia weight.

PSO is a robust method for global optimization problems, despite its drawback: the large number of objective function evaluations usually required (MORAES *et al.*, 2014). This might be especially critical for daily large-scale engineering problems, in which objective function evaluation tends to be a costly step in the optimization algorithm and a large number of evaluations might mean impractical computation time and effort (MORAES *et al.*, 2014). However, in the last few years computational processing capacity has been substantially increased and this type of method has been gaining prominence for engineering problems. This method is especially suitable for parallel computing. Chapter 4 will present the overall computation demand for the system in study.

2.3.7 Integer Optimization: Branch-and-Bound Method

The branch-and-bound method was originally proposed by LAND and DOIG (1960) as an automatable routine for discrete and mixed programming problems. Since then, literature

works have approached different strategies to obtain the *bounds* and to select the vertices and variable for *branching* (KIANFAR, 2011):

Bounds: Typically, the lower bound — for minimization problems — is determined by relaxing the integer variables and solving for any of the subproblems.

Branching: The classical and most simple approach is to select the integer value with the most fractional value and add linear constraints to split the feasible region.

This method is used today in almost all integer programming solvers of Integer Programming (IP) problems, including MILP and MINLP (KIANFAR, 2011).

Branch-and-bound approach is to employ branching and pruning strategies to avoid complete explicit enumeration of binary solution space, since even small problems might generate a large number of solution points when integer variables are added (KIANFAR, 2011). In general terms, this method systematically decomposes the original problem into subproblems (*branching*) and performs a tree-search over the space of binary variables — the tree represents 0-1 combinations (KIANFAR, 2011; MORRISON *et al.*, 2016). Each subproblem is a node of the tree that can be further branched by partition of the solution space into smaller regions (MORRISON *et al.*, 2016). The optimal solution is found when the entire tree has been explored.

In order to reduce the number of subproblems to be solved, pre-defined criteria guide tree pruning. When a node is pruned, its subproblem is not partitioned anymore and there is no deeper branching. Upper and lower bounds are calculated and continuously updated to guide and guarantee smart tree pruning, which can happen in three cases (KIANFAR, 2011):

1. Infeasibility: Node subproblem has an empty region of feasibility.

2. Bound: Node subproblem has its best objective value worse than a solution already known.

3. Optimality: Node subproblem has its optimal solution found.

Another important strategy is node selection, which defines the priority for node evaluation. A widely used criteria is *depth first node selection* (KIANFAR, 2011; MORRISON *et al.*, 2016), which states that the next node is:

1. **For current node not pruned:** one of the children of current node.
2. **For current node pruned:** the child of the first node, with an unconsidered child node, found by backtracking from the current to root node.

2.4 Final Considerations

Literature review for modeling, simulation and optimization of natural gas processing showed that focus on single NGPUs is the most occurring. Works with integrated approach on multi-unit gas processing sites are scarce. Because of that, there is no well-established methodology looking at business optimization of natural gas processing facilities.

In that context an economic optimization of a multi-unit multi-feedstock gas processing site, considering business aspects, is proposed. Instead of traditional literature approach using temperatures and pressures as decision variables, here main process feed flowrates and NGPUs operating status are considered.

Chapter 3

Methodology

To achieve the objective of this work, an actual industrial site with several distinguished units was chosen as case study and is considered for modeling and business optimization. The developed framework combines process simulation of the gas processing facility with Python-implemented solution for the MINLP optimization problem, resulting in the operating point of maximum business profit.

In this approach, the MINLP model was broken down into two contributions: MIP and NLP. A branch-and-bound algorithm was used to solve the MIP problem, whereas for the underlying NLP problem two methods in series were employed: global PSO for pre-screening followed by local Nelder-Mead method for refinement. The optimization framework was coupled with the simulation model developed in HYSYS v11. To communicate Python with HYSYS and facilitate data visualization and customization, Microsoft Excel was used as a data transfer interface.

At each iteration, the process simulator receives the decision variables (u and y) passed by the optimization module via an Excel VBA (visual basic for applications) macro and runs mass/energy balances to calculate output model variables $x(u, y)$. Those model variables are passed back from HYSYS to Excel and then Python, which uses them to compute the objective function. At the end of the iterative process, upon convergence, a maximum value of overall business profit is determined, associated with optimum values for u and y (u^{opt} and y^{opt}). Figure 3.1 shows a schematic representation of this communication, flow of variables and final results with optimum feed allocation among NGPUs (continuous variables u) and best economic NGPU operating status (integer variables y).

The following steps were executed as part of this work, in this order:

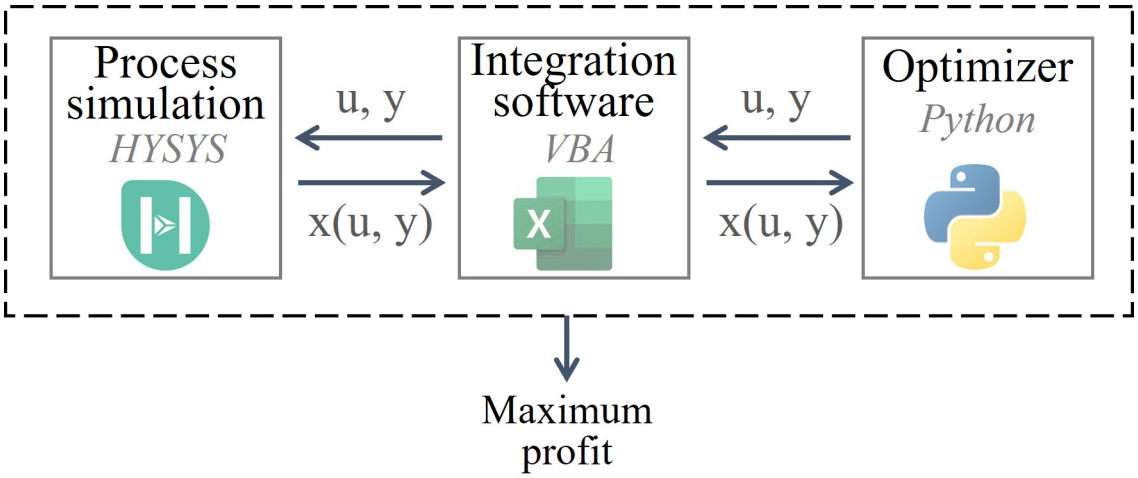

Figure 3.1: Schematic representation of the simulation-optimization framework built up in this work.

1. Simulation model implementation in HYSYS.
2. Model validation and tuning with actual plant data.
3. Definition of main variables of interest (decision and model variables).
4. Definition of the most suitable optimization framework and method.
5. Optimization problem formulation and computer implementation.
6. Optimization problem solution and result discussion.

Next subsections explain the approach and assumptions considered for process simulation and optimization, as well as a comprehensive description of the industrial site chosen for study.

3.1 Process Description

The system in study is Cabiúnas Natural Gas Processing Site (UTGCAB), a Petrobras facility located in Macaé, Rio de Janeiro. UTGCAB is responsible for receiving non-processed raw gas from three different main offshore sources and processing it into four products: sales gas, NGL (C_2^+), LPG and C_5^+. There are also intermediate products, which destination will be detailed below; they are: NGL from NGPU-A (C_3^+) and residual gas.

The core of the industrial site are the five NGPUs (NGPU-A/B/C/D/E), which may differ from one another in terms of refrigeration process, liquid recovery, performance,

minimum and maximum allowable feed rates, feed sources, operating modes and products.

There are also three liquid fractionation units (LFUs), responsible for receiving unstabilized condensate from slug catchers, intermediate NGL stream from NGPU-A (C_3^+) and a lighter liquid generated inside NGPUs knockout drums by pressure drop and/or gas cooling. Furthermore, auxiliary treating units are present to remove the contaminants water, CO_2, H_2S and mercury. A general schematic of UTGCAB is presented in Figure 3.2.

Raw gas comes into three slug catchers (SG-A/B/C) that work with different gas flowrates/quality and process conditions (temperature and pressure). Also, due to UTGCAB's complexity, there are many possible configurations for sending the multi-feedstock raw gas to the process units. Multiple pipe routings exist as a result of the fact that process units have been gradually built over time, which means that raw gas from one SG might be processed in one or more NGPUs and that one NGPU might receive raw gas from one or more SGs. Those possible routes are the continuous decision variables (u) of the optimization problem, as indicated in Figure 3.2.

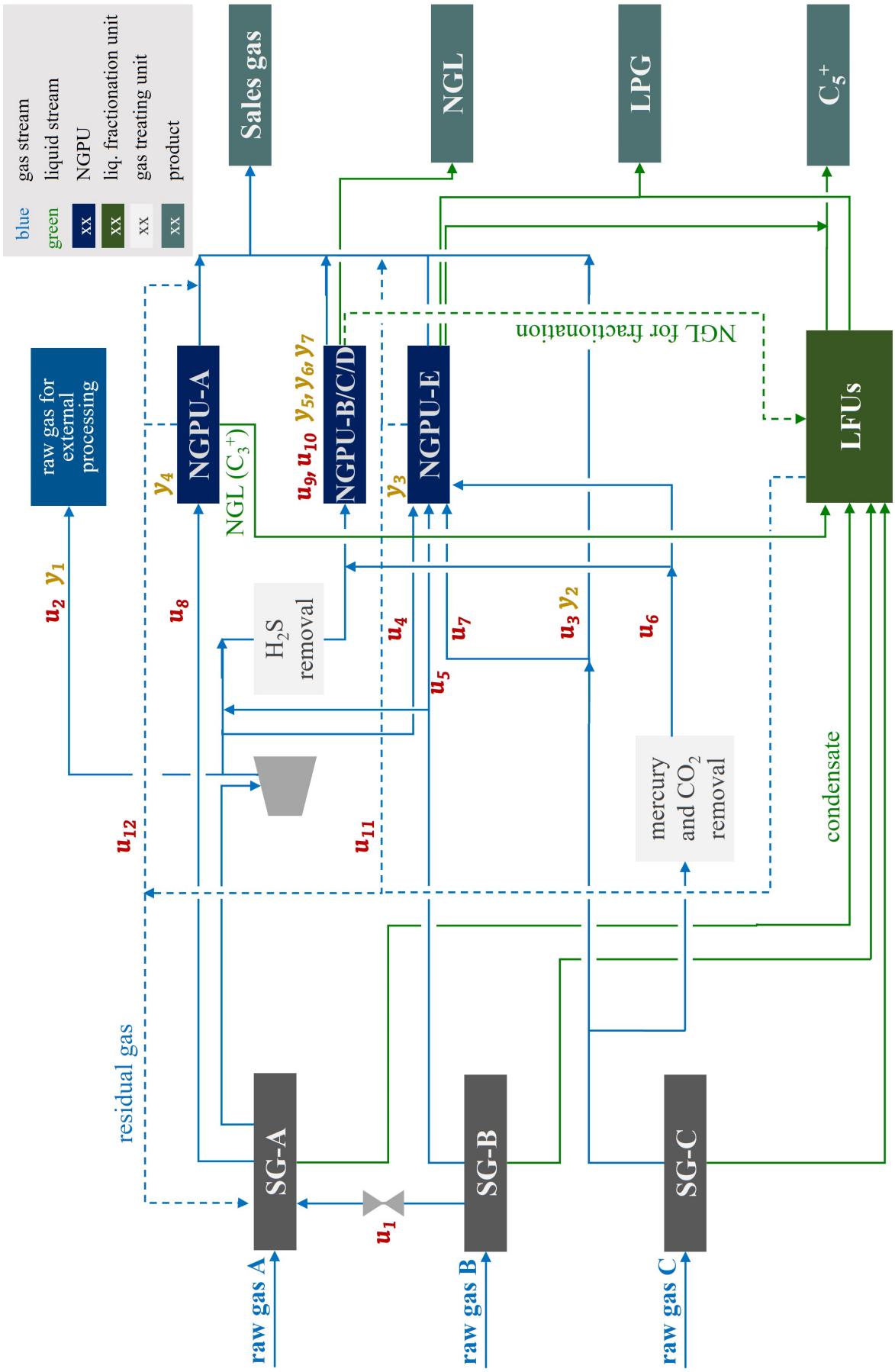

Figure 3.2: General schematic of the gas processing site in study. Created by the author.

SG-B has the flexibility of sending gas to SG-A (decision variable u_1). SG-A works at lower pressure and therefore needs compression when sending raw gas to NGPUs-B/C/D/E, which might increase costs. H_2S removal is a pre-treatment required exclusively by NGPUs-B/C/D. Molecular sieve water removal is present in NGPUs-B/C/D, upstream to gas cooling. Mercury and CO_2 removal units are dedicated to raw gas from SG-C. CO_2 removal units can be partially bypassed and the gas outlet from SG-C directly mixed into the sales gas stream (u_3). Residual gas from process units is sent back to SG-A for reprocessing; however, NGPU-A and NGPU-E also have the flexibility of mixing part or all of their residual gas into sales gas, which are additional decision variables (u_{12} and u_{11}).

NGPUs-B/C/D are the only ones that produce market-final product NGL (C_2^+) — part of this product might be sent, by operational and/or economical decision, to LFUs for fractionation, yielding LPG and C_5^+. However, a minimum NGL net production is required to fulfill an external sales contract. Table 3.1 brings a summary of products yielded by each process unit.

Table 3.1: List of products generated in each process unit studied in this work (NGPU-A/B/C/D/E and LFUs): sales gas, NGL (C_2^+), LPG and C_5^+ are final products destined for sale whereas residual gas and NGL (C_3^+) are intermediate streams.

Product	NGPU-A	NGPU-B/C/D	NGPU-E	LFUs
Sales gas	X	X	X	
NGL (C_2^+)		X		
LPG			X	X
C_5^+			X	X
Residual gas	X		X	X
NGL (C_3^+)	X			

Product profile and overall operating costs will vary according to feed route configuration and NGPUs operating modes/status. However, this is currently a manual decision of the Operations team, potentially leading to suboptimal operating conditions.

3.2 Process Modeling and Simulation

Process modeling of the integrated gas processing site is a necessary step to carry out optimization and achieve the objectives of this work. Commercial process simulator HYSYS was chosen as the modeling framework. Next subsections give details on approach, fluid package, workarounds and specificities of the model.

3.2.1 Digital Modeling Approach

A first principles approach was used to model the natural gas processing site. Commercial software HYSYS v11 was used to build up the process simulation, which is a rigorous static representation of the real site. However, since the focus of this work is production estimation via mass and energy balances, whereas keeping a short simulation time to make total optimization time feasible, it was convenient to implement simplified models for auxiliary process units. For this reason, treating units and other systems such as MEG dehydration and propane cycle were represented in a simplified approach.

In general, single-stream heaters and coolers were specified by inlet and outlet temperatures. Energy integration exchangers were defined by overall UA (product of overall heat transfer coefficient and heat transfer area), tuned by historical operating data of those equipment. *Simple end point* HYSYS model was used for single phase application, whereas *Simple weighted* was applied when phase change was present.

For compression equipment, outlet pressure was set and operational efficiencies were used. Distillation columns were solved with HYSIM Inside-Out method. Two liquid phase checking was disabled in all distillation columns for performance improvement, since it is not expected to have two liquid phases inside them.

Digital model representation included several equipment and unit operations, solved by HYSYS built-in methods. The main equipment with their respective solving methods and specifications are listed in Table 3.2.

3.2.2 Component List

The simulation considered a component list with a total of 17 components: normal and iso alkanes up to 10 carbons (C_1 to C_{10}), and non-hydrocarbons: H_2S, CO_2, N_2, H_2O and MEG. No other sulfur components were considered, due to their low content in the reaction system chosen for study. MEG is not considered in the gas feed stream, but is present in the NGPUs that use MEG injection for hydrate inhibition, as detailed in Section 3.2.3.

3.2.3 Plant Representation

Slug Catcher

The slug catcher was represented in a simplified approach as a three-phase vessel with gas, liquid hydrocarbon and aqueous outlet streams. No external (convection/radiation) heat

Table 3.2: List of equipment present in the digital simulation model and the corresponding solving methods and default specifications.

Equipment	Solving method	Default specification	Specification source
Distillation column	HYSIM inside out	Case by case evaluation	The component recoveries were estimated from plant history data (flowrates and compositions of the outlet streams) aided by measurements of column temperatures and pressures.
Demethanizer	HYSIM inside out	Component recovery	
Deethanizer	HYSIM inside out	Ethane over ethane, propane and butane ratio	
Debutanizer	HYSIM inside out	Butane fraction on C_5^+ and pentane fraction on LPG	
Single-phase heat exchanger	Simple end-point	UA	UA: estimated from temperature history data of both inlet andoutlet streams. The value estimated was kept constant for all simulations.
Multi-phase heat exchanger	Simple weighted	UA	
Flash vessel	Rachford-Rice	Temperature and pressure	T and P either calculated by HYSYS or specified from plant history.
Compressor	N/A	Adiabatic efficiency and discharge pressure	Discharge pressure: from plant history. Efficiency: collected from the plant Operations team.
Turbo expander	N/A	Isentropic efficiency and discharge pressure	Discharge pressure: from plant history. Efficiency: collected from the Operations team.
Pump	N/A	Adiabatic efficiency and discharge pressure	Discharge pressure: from plant history. Efficiency: collected from the Operations team.
Valve	N/A	Inlet and outlet pressures	Inlet pressure: calculated by the process simulator. Outlet pressure: estimated from plant history data.
Forced separator (contaminants)	N/A	Outlet contaminant fraction	Outlet contaminant fraction: defined from history plant data (composition of the outlet stream).

transfer was considered between rich gas and external ambient. Temperature and pressure are considered constant inside the slug catcher. Therefore, the simulation calculates a TP flash for the inlet feed stream. Those conditions are defined by typical plant measurements:

Temperature: might be a function of ambient conditions, flowrate and upstream pres-

sure. Ranges between 15 and 25 ^oC.

Pressure: at steady state conditions, is fixed and controlled by plant operation at default values. Ranges between 55 and 90 bar.

Gas Treating

UTGCAB has mercury, CO_2 and H_2S removal units. They are represented in simplified approaches:

Mercury: not represented explicitly, due to the low content of mercury in the inlet streams (parts per billion) and the lack of accurate representation of this process via HYSYS simulation. Mercury was not inserted in the component list, with no drawbacks for this work's purpose.

CO_2: represented as a set of simplified unit operations based on the general reactive absorption process (Figure 2.5). It includes the elements presented in Figure 3.3: a component splitter for forced CO_2 removal, water saturation to represent contact with aqueous amine solution and heat exchanger to reproduce increase in temperature due to exothermic reactive absorption. Amine streams were not represented nor alkanolamines were included in the component list.

H_2S: represented as a component splitter with forced H_2S removal and operational efficiency.

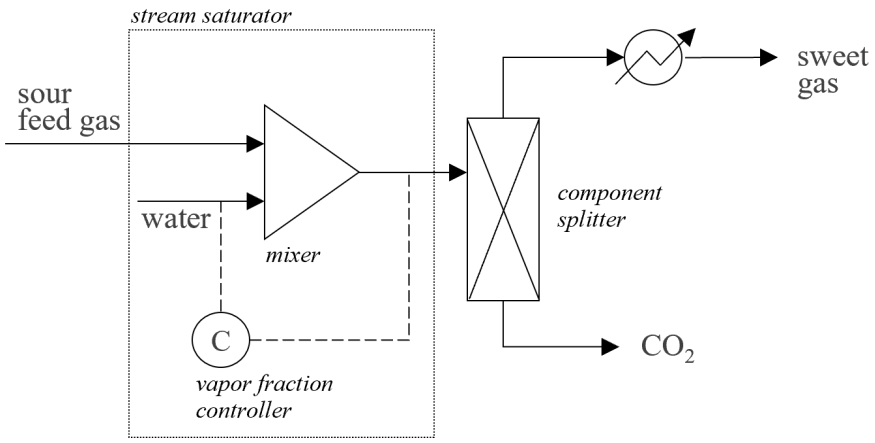

Figure 3.3: Schematic diagram showing how reactive absorption CO_2 removal treating unit was represented in the simulation model. Created by the author.

Gas Dehydration

The simulation represents two different processes for gas dehydration, according to the technology available in the respective NGPU — MEG injection and molecular sieve — as detailed in Table 3.3.

It is important to notice that MEG dehydration was represented in a simplified approach in the simulation. MEG injection in gas cooling heat exchangers is considered by mixing MEG and natural gas streams in the inlet of the equipment. MEG regeneration subflowsheet was disabled in order to improve simulation performance, since it is not relevant for this work's purpose and would consume execution time and require another thermodynamic model more suitable for this process (for instance, CPA EoS or another SAFT-type equation).

Cryogenic Natural Gas Cooling

Three distinct gas cooling processes are represented in the simulation, according to the respective NGPU, as detailed in Table 3.3. In this table, the term *enhanced mechanical refrigeration* refers to a modified process in which the NGPU has a gas cooling section with typical propane refrigeration cycle and a fractionation section using propane condenser in the deethanizer column for enhanced ethane recovery. Furthermore, all NGPUs with mechanical refrigeration listed use propane as cooling fluid.

Table 3.3: Gas dehydration/cooling processes used in each NGPU of this study and their products.

Process unit	Gas dehydration process	Gas cooling process	Products
NGPU-A	Temperature decrease with MEG injection for hydrate inhibition	Mechanical refrigeration	Sales gas, NGL (C_3^+) and residual gas
NGPU-B/C/D	Molecular sieve	Mechanical refrigeration, turboexpander and Joule-Thomson	Sales gas and NGL (C_2^+)
NGPU-E	Temperature decrease with MEG injection for hydrate inhibition	Enhanced mechanical refrigeration	Sales gas, LPG, C_5^+ and residual gas

NGPU-A is the least complex unit: it has only a gas cooling section with mechanical refrigeration, a layout similar to the simplified schematic from Figure 2.3. Due to its

temperature levels, liquid recovery is restricted to propane and heavier hydrocarbons. Recovered NGL (C_3^+) necessarily goes to LFU-A for fractionation into LPG and C_5^+. It is therefore the NGPU with the lowest liquid recovery fraction.

NGPU-B/C/D have the highest liquid recovery and the lowest temperatures, due to the combination of gas cooling technologies. Their layout is a blend of Figures 2.3 and 2.4 schematics. NGL from these NGPUs is formed by hydrocarbons heavier than ethane (C_2^+) and need no further fractionation. Optionally, part of this NGL can feed the LFUs, producing more LPG and C_5^+.

NGPU-E is more complex due to the presence of a liquid fractionation hot section that receives the NGL generated inside the unit's gas cooling area. Its layout is a combination of a propane refrigeration cold section (Figure 2.3) and a hot liquid fractionation area (adaptation of Figure 2.6). Because of that, this unit has no connection to the LFUs.

Both NGPU-A and NGPU-E produce residual gas streams that can be mixed with UTGCAB's overall sales gas stream and/or be sent back to SG-A for reprocessing. This is an individual operational flexibility for each of these NGPUs and can even be used simultaneously, with part of residual gas going back for processing and part mixed to sales gas.

It is important to notice that the mechanical refrigeration gas cooling process was not represented rigorously in the simulation model. Propane cycle subflowsheets were disabled for the sake of simulation performance, since their representation was not essential for this work's purpose and would lead to increased computational demand. Instead, temperatures were fixed in the outlet process streams of propane chillers, based on design and historical operational values, valid for the working operation range.

Liquid Fractionating Units (LFUs)

The three LFUs represented are considered to be identical, with the exception that LFU-A is the only one that can receive NGL (C_3^+) from NGPU-A. Their inlet streams come from the three slug catchers (SG-A/B/C), as well as from NGPU-A (exclusively to LFU-A). They can also receive processed NGL (C_2^+) from NGPUs-B/C/D, which is an operational flexibility that might or not be used according to economic, logistic or operational interest.

Their layout is analogous to the simplified schematic from Figure 2.6 and all three units produce LPG, C_5^+ and a residual gas stream, which is necessarily returned to SG-A for reprocessing.

3.2.4 Thermodynamic Model

In a process simulation, the thermodynamic model is responsible for calculating thermodynamic and most part of transport properties. In HYSYS, that is called Fluid Package. For gas processing applications, classical equations of state tend to have good performance, as discussed in Section 2.3.1. Actually, originally they were formulated as models to be fit for the Oil & Gas industry.

In agreement with literature studies and recommendations, Peng-Robinson EoS was chosen for this study. In the property package used, the Peng-Robinson EoS is considered for the calculation of all properties, except liquid molar volume, which was calculated by the COSTALD correlation due to its higher accuracy, as explained in Section 2.3.2.

HYSYS-implemented Peng-Robinson property package is recommended for temperatures above $-271\ ^oC$ and pressures below $100\,000\ kPa$, for VLE calculations of hydrocarbon systems with possible non-idealities. Interaction parameters are customized by Aspentech and default values were used. Aspentech also recommends the use of PR EoS for cryogenic gas processing, corroborating this choice.

Considerations on Thermodynamic Model Selection: The Cubic Plus Association EoS (CPA EoS)

The natural gas plant selected as this work case study contains streams with mixtures of natural gas, water and MEG. For such systems, some recent works in the literature such as KONTOGEORGIS and FOLAS (2009) suggest the use of CPA EoS, since it is a more appropriate model to represent systems with polar interactions, as discussed in Section 2.3.1. With a more rigorous theoretical basis than PR for modeling polar compounds and good fit compared to experimental data (for further detailes, see KONTOGEORGIS and FOLAS (2009)), CPA is presented as a promising model.

However, CPA higher complexity in comparison to cubic EoS might lead to numerical instability in flash calculation of streams containing high methane content in medium/high pressures. This issue has already been detected and discussed in previous works, such as DOS SANTOS *et al.* (2015), and creates a need of special attention for the arise of numerical issues. Furthermore, its implementation on commercial process simulation software is still not well-established when compared to cubic EoS. As an undergoing work, it still requires further testing and adjust to guarantee smooth utilization by the end-user in high dimension problems. This is especially true for cases in which the simulation model will still be coupled with an optimization layer, such as the one proposed

in this work.

For the sake of result comparison, this work developed the HYSYS simulation plant model utilizing both PR and CPA, considering the latter as benchmark for the system here in study, which contains a mixture of hydrocarbons and polar compounds such as water and MEG. The idea was to compare numeric results and execution time between both EoS. However, while the model ran smoothly with PR, the same was not true with CPA: when changing the fluid package from PR to CPA, convergence inconsistencies arose.

An alternative approach was then used: to compare PR and CPA results in the critical sections of the simulation, i.e. conditions of flowrate, composition, temperature and pressure in which the association effects could be relevant. It was therefore chosen the region around the three-phase vessel of the NGPUs that have MEG injection for dehydration (NGPU-A and NGPU-E). Also, this is the section of the NGPU with the lowest temperatures and highest MEG content.

For the vessel outlet gas-phase, water and hydrocarbon dew points were calculated for both EoS and compared. For the inlet stream of the three-phase vessel, comparison was carried out based on the phase envelope generated for both PR and CPA. All CPA calculations considered self-association for water and MEG, using 4C association scheme for both, as recommended by ABUNAHMAN *et al.* (2020); HUANG and RADOSZ (1991); KONTOGEORGIS and FOLAS (2009). No association scheme was considered for the CO_2 molecule, which is non-self-associating (KONTOGEORGIS and FOLAS, 2009). However, CO_2-water interaction might be affected by solvation effects. In those cases, treating CO_2 as a four-site associating molecule might yield to good results (KONTOGEORGIS and FOLAS, 2009). For this work, no CO_2 self-association was considered, in accordance to the results of ABUNAHMAN *et al.* (2020), who showed that there is an inversion point in the water content for mixtures of water plus natural gas as a function of pressure variation — from this point onward gas-phase water content increases with pressure, as a consequence of the association effects. Above this inversion point it would be necessary to account for the association effect of CO_2. ABUNAHMAN *et al.* (2020) showed that this inversion point happens in pressure levels from 80 bar (80 % CO_2) to 260 bar (10 % CO_2). This current work considers even lower CO_2 contents, which would lead to a higher inversion pressure, much higher than the NGPUs working pressure. Therefore CO_2 association effect could be disregarded with no prejudice for this work.

For the physical term of CPA EoS, conventional mixing rule was applied for parameter calculation, whereas CR-1 combining rule was used for the association terms — the

equations were presented in Section 2.3.1. Figure 3.4 shows a schematic of part of the gas cooling section of those NGPUs, highlighting the three-phase vessel in bold and indicating the main streams of interest in red along with their calculated properties for comparison.

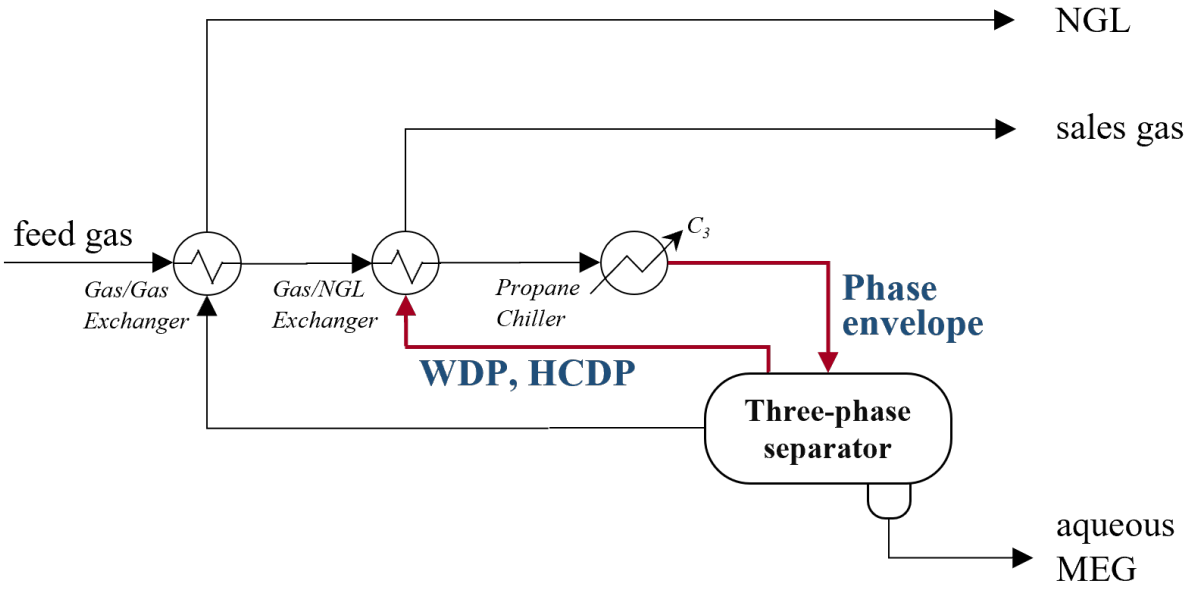

Figure 3.4: Schematic representation of the NGPU section (part of gas cooling) chosen for result comparison between PR and CPA EoS. Created by the author.

Tables 3.4 and 3.5 show, for NGPU-A and NGPU-E, respectively, a comparison between properties calculated for gas-phase outlet stream from three-phase vessel using PR and CPA EoS. Results show good agreement between the calculation of both EoS for all properties evaluated. NGPU-E comparison shows an even better agreement than NGPU-A, with values calculated by both EoS nearly the same. For NGPU-A, CPA-calculated values for WDP and HCDP are, respectively, 3.49 and and 9.92 % lower than the ones calculated with PR EoS. However, this is not an issue since it is a discrepancy lower than 10 % and PR values are more conservative, in the sense that they are closer to the upper limits of regulation (Table 2.1).

Finally, Figures 3.5 and 3.6 show the phase envelope for the inlet stream of three-phase NGPU vessel. Full lines correspond to values calculated with PR Eos while scatter points are CPA-calculated values, blue for bubble and red for dew points. Highlighted dashed-dotted areas correspond to NGPU operating range — and therefore the region of interest. Once again, results show good agreement between calculated values with both EoS.

Table 3.4: Comparison between PR and CPA EoS. Properties calculated for gas-phase outlet stream of NGPU-A three-phase vessel.

Property	PR EoS	CPA EoS
Temperature (^oC)	-30.20	-30.20
Water content (mol %)	0.000696	0.000532
Water dew point (^oC)	-67.90	-70.27
Hydrocarbon dew point (^oC)	-30.35	-33.36

Table 3.5: Comparison between PR and CPA EoS. Thermodynamic properties calculated for gas-phase outlet stream of NGPU-E three-phase vessel.

Property	PR EoS	CPA EoS
Temperature (^oC)	-26.82	-26.82
Water content (mol %)	0.000925	0.000938
Water dew point (^oC)	-65.47	-65.36
Hydrocarbon dew point (^oC)	-27.21	-26.98

In the light of above considerations, considering the laborious attempt of converging the model using CPA EoS using the same assumptions and simulation strategy from PR simulation model and comparative results that showed good agreement between PR- and CPA-calculated values, the Peng Robinson method was the EoS selected for this work.

3.2.5 Simulation Strategies

The execution of a simulation model coupled with an optimization algorithm brings additional challenges due to the elevated number of times that the simulation will need to be solved sequentially. In addition, there is the fact that the optimizer might and should take the simulation model to extreme conditions that differ from one iteration to the other.

This scenario arises the need for implementation of strategies to guarantee adequate simulation performance in terms of convergence, stability and execution time in the whole range of cases that might be covered by the optimization method. Main simulation strategies employed in this work will then be presented and discussed in the next topics for further comprehension.

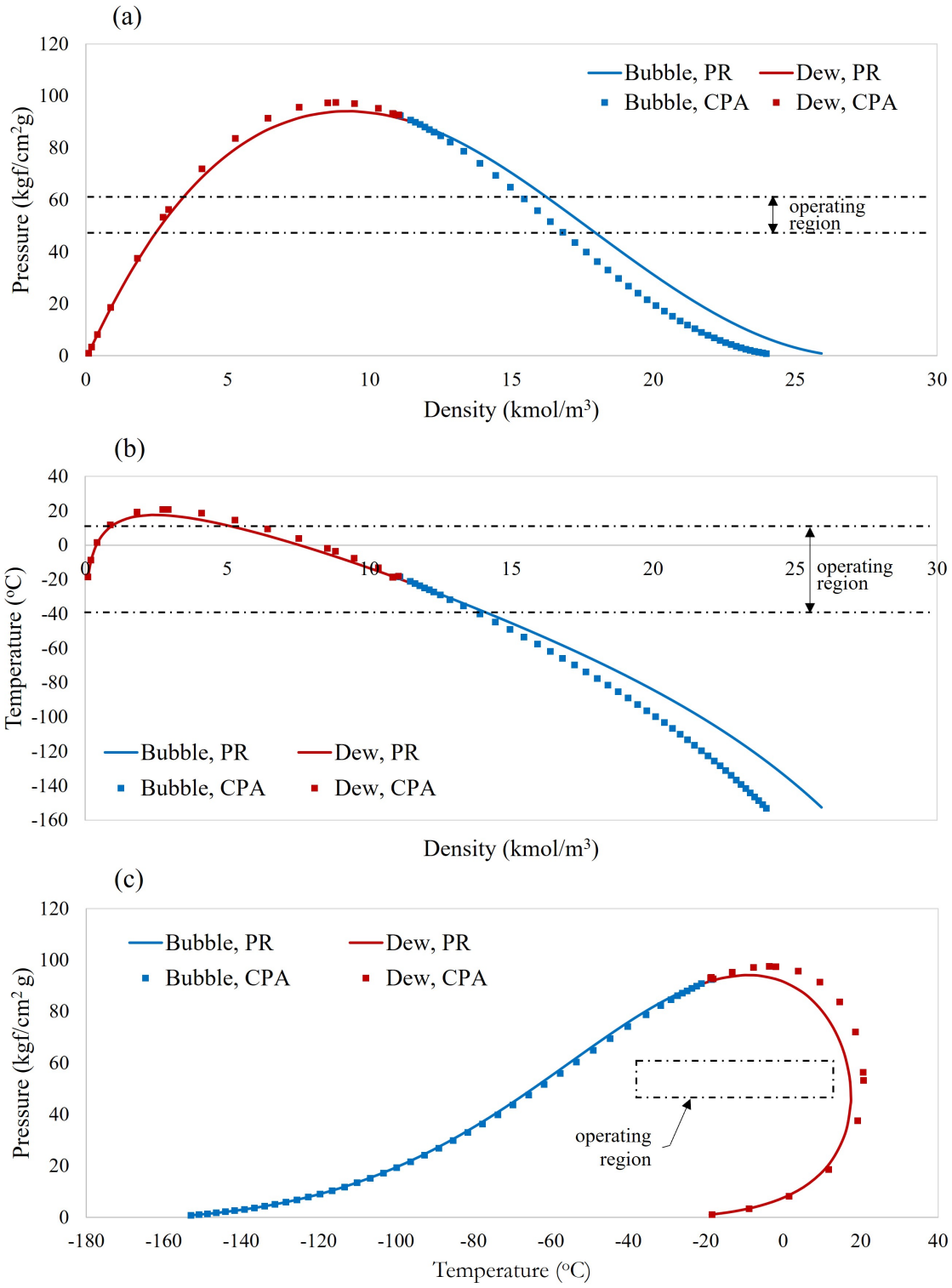

Figure 3.5: Phase envelope for NGPU-A inlet stream of three-phase vessel calculated with PR (red) and CPA (blue) EoS for comparison. Dashed-dotted region highlights NGPU operating range.

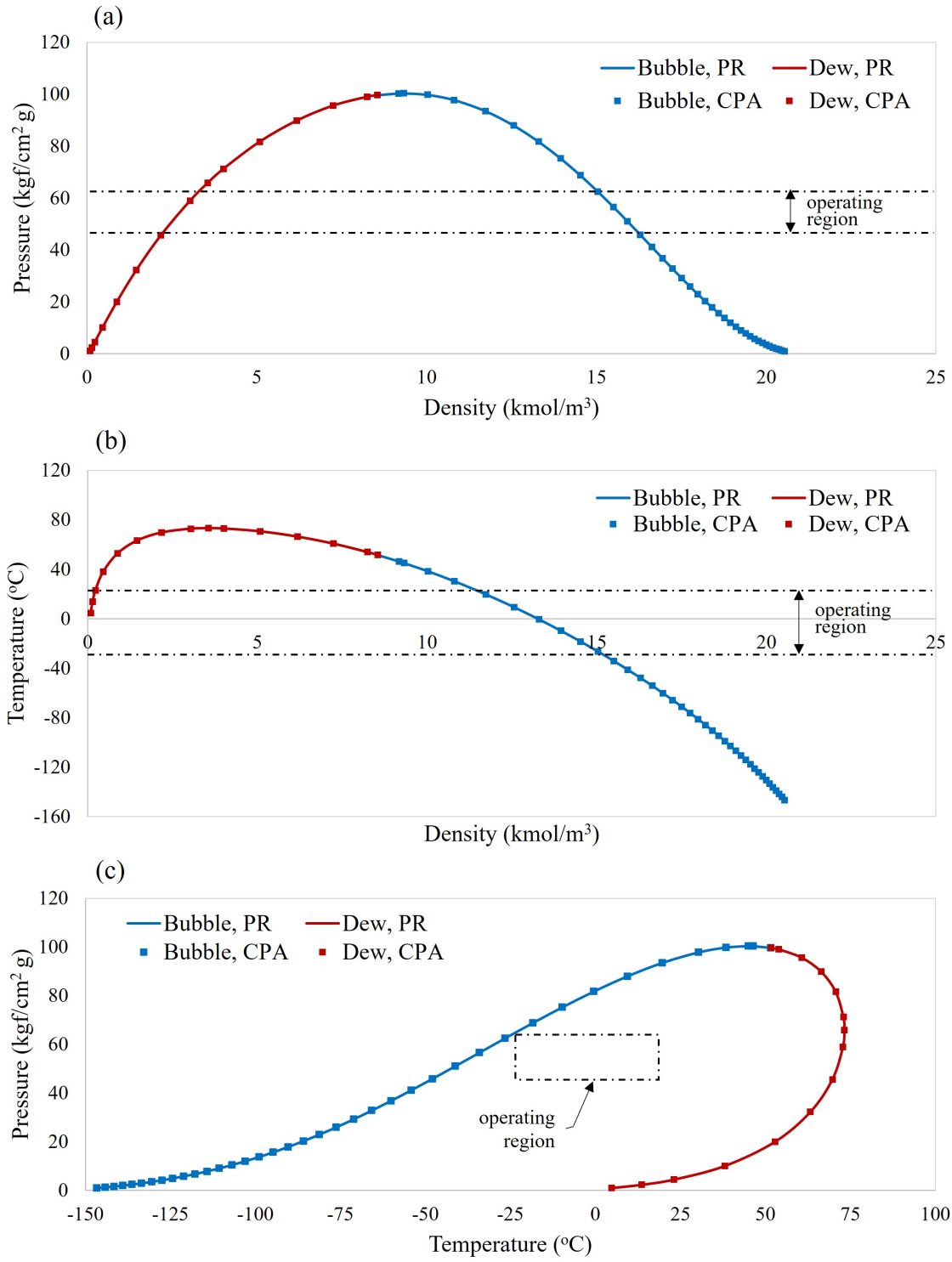

Figure 3.6: Phase envelope for NGPU-E inlet stream of three-phase vessel calculated with PR (red) and CPA (blue) EoS for comparison. Dashed-dotted region highlights NGPU operating range.

Recycle Blocks: Tear Streams

Recycle logical blocks were strategically added to the simulation. For instance, in order to improve convergence, a recycle stream was defined as the inlet of every energy integration

heat exchanger and return streams, such as residual gas for reprocessing.

Recycle Blocks: Residual Gas

There are five main residual gas streams in the simulation model, respectively from: NGPU-A, NGPU-E and the three LFUs. All those residual gas streams might return do SG-A for reprocessing.

In terms of modeling, they create complex recycles that are handled with HYSYS logical blocks. Those recycles are particularly important because impact simulation convergence and results, since they connect the outlet of several process units to the slug catcher, which is the first element of the simulation model. They also affect the total amount of raw gas that is available for processing, which is a key parameter for this work's optimization model but is only available after the model is executed.

For this reason, during the implementation it was assured that:

1. The main recycle streams were identified and analyzed properly (i.e. the location of recycle logical blocks was exhaustively tested and compared against simulation time and convergence robustness). Furthermore, suitable calculation priority was defined compared to the other elements of the model.

2. The numerical tolerance of HYSYS recycle blocks was evaluated in terms of impact in the simulation results and optimization performance. This topic will be further detailed in the following item.

Recycle Blocks: Convergence Tolerance

Each recycle block in HYSYS has an associated tolerance to indicate convergence when comparing recycle inlet and outlet streams. To increase this tolerance value means that less iterations will be need to reach final results and consequently simulation will improve performance, reducing execution time. However, increasing the tolerance above a certain limit can cause significant fluctuations in the simulation results: for the same inlet conditions, different outputs might be accepted as long as they are within the defined tolerance.

These fluctuations can cause problems in the local optimization algorithm, that might stay locally trapped if the simulation fluctuations are higher in magnitude than the tolerance specified for the optimization algorithm.

In order to adjust that, the software allows the user to specify a relative tolerance. Then the actual tolerance is calculated by multiplying this user-specified relative tolerance (*default* = 1) by an internal fixed one. The latter is defined by HYSYS and is an absolute value for all parameters except flowrates. For temperatures, for instance, the value is 0.1^oC, which will be multiplied by user-specified relative tolerance. For flowrate, the absolute tolerance is calculated by multiplying the actual stream flowrate in $kmol/s$ by the factor of 0.001. Therefore, the lower the stream flowrate, the lower its absolute tolerance.

Convergence

Simulation convergence is an important feature for the optimization algorithm. Non-converged scenarios must be considered unfeasible and discarded. In order to determine if a simulation has converged or not, a flag variable was created in Excel/VBA and its value is changed by analyzing the simulation overall product streams. If those streams are all fully calculated (all properties known), it indicates that each individual process unit (NGPUs and LFUs) has been calculated and therefore the simulation is considered converged. Otherwise, a non-convergence flag is set.

This was proven to be an efficient and fast approach to analyze convergence. It was implemented to eliminate the need of iterating through each and every simulation stream to check if all their properties were known.

Plants Operating Status

All five NGPUs have the flexibility of staying in standby when economically advantageous. This makes five integer decision variables.

In the simulation, when in standby, NGPU inlet and outlet flowrates were all set to zero. Furthermore, the NGPU subflowsheet was ignored in HYSYS, so that the process simulator would not spend time trying to converge this NGPU.

3.2.6 Validation Approach

For this step, the developed HYSYS model was used as a simulator (degree of freedom equal do zero) in a predictive manner. Model was validated against historical industrial plant data. Monthly average input data was used to compare actual production — mole

flowrate of each product — against simulation outputs.

Monthly data was chosen for comparison instead of instantaneous values because they are reconciled for more representative results. For each validation point, several plant process information was collected from data historian in order to adjust the simulation model to the actual operating conditions. These included temperature, pressure, flowrates, component recoveries, operating modes and feed route configuration.

3.3 The Simulation-Optimization Integration Tool

The simulation-optimization integration tool is the Excel spreadsheet responsible for data transfer between process simulation model and optimization algorithm. It guarantees automatized data transfer data between HYSYS and Python, and VBA was chosen as the programming language. It also aids user monitoring and debugging of the process, since Microsoft Excel is an extensively used visual software.

The structure of implementation consists of one Excel sheet for simulation data and another regarding optimization information. Model input communicates to HYSYS from cells in the simulation tab, which also receives model output from the simulation. Python communication is restricted to the optimization sheet, which receives continuous and binary decision variables from Python optimizer. Simulation and optimization sheets also communicate between each other, enabling calculations such as of the objective function.

3.4 The Optimization Framework

As mentioned previously, a gas processing site consists of several individual plants for gas treating, gas processing and liquid fractionation. Each of those involve several non-linear processes, such as distillation, VLE calculations, gas compression and dew-point control (ZHANG *et al.*, 2016). Those non-linearities are intrinsic from the process; to take them into account, a non-linear optimization approach was carried out.

Furthermore, a given NGPU has upper and lower feed boundaries — below the minimum flow, the plant needs to be shutdown because continuous operation is not feasible, whereas the maximum flow is limited by design specifications and public operating license. Those conditions generate integer binary decision variables that must be considered in the optimization problem, which becomes a Mixed Integer Nonlinear

Programming (MINLP). This is in agreement with the literature, which states that MINLP techniques are well-suited to such problems: nonlinear process phenomena plus desired selection of process alternatives, possibly with discrete variables (MENCARELLI *et al.*, 2020).

The overall model includes 19 decision variables: 12 continuous and 7 binary. There are also 63 inequality constraints, including restrictions of product quality, logistics and transportation and process units' design limitations. The strategy for solving the optimization problem, as well as a detailed description of the problem itself will be covered in the next subsections.

3.4.1 Optimization Strategy

The MINLP optimization problem was broken down into the combination of two others: NLP plus MIP:

NLP: solved by Nelder-Mead flexible polyhedron local optimization algorithm with a pre-screening using PSO global method.

MIP: solved by a branch-and-bound algorithm with breadth-first search and binary branching.

The branch-and-bound algorithm explores the binary space of variables via tree-search. At every iteration of the MIP algorithm, an NLP subproblem needs to be solved and then Nelder-Mead routine is called to solve for the local minimum of that node. As the NLP subproblem was solved using the feasible path approach, where the equality constraints are solved by the process simulator, the inequality constraints were included as penalty functions to the objective function to be able to use the Nelder-Mead algorithm. Furthermore, it is important to point out that HYSYS model provides no direct information of the objective function gradient, which makes direct-search methods such as Nelder-Mead flexible polyhedron particularly suitable for this work since they are derivative-free.

The system model integrates several NGPUs, LFUs and other auxiliary systems, consisting of a large number of nonlinear unit operations, besides the existence of integer decision variables. The high complexity of this system might generate non-convexities in the optimization problem. As a result, local optimization methods might get trapped in local minima and struggle in finding the global optimum.

For this reason, a pre-screening step was included in the optimization solving approach with global non-deterministic PSO method. This strategy aims to use one PSO execution with relaxed-integrality binary variables to find the vicinities of the global optimum. The PSO solution is then used as an initial estimate for the MINLP algorithm (branch-and-bound plus Nelder-Mead flexible polyhedron). It is assumed that the global MINLP solution is near PSO relaxed optimum. PSO global method was also used as a tool to study and gather knowledge over the space of variables of the problem.

The optimization problem was modeled in Python, which is a well-established open software with robust packages for several purposes, including mathematical optimization. For global pre-screening, *pyswarm* package was chosen, with routine *pso*, which was customized for this work.

For local NLP, *Scipy* package was used, in submodule *optimize* and routine *minimize* (*scipy.optimize.minimize*). This routine has many built-in optimization methods implemented, among them Nelder-Mead algorithm, chosen as the local NLP algorithm in this work. It is important to mention that, for the generation of initial estimates of the NLP subproblems, an LP problem was formulated and its results were utilized — further details will be given in Section 3.4.5. Lastly, for MIP, a Matlab implementation of branch-and-bound (SOARES, 2001) was adapted to Python and tailored for this work. This code is an extension of the implementation of spatial branch-and-bound for MINLP problems by PEREIRA (1999), incorporating equality constraints.

Global NLP Optimization: PSO

For the pre-screening and analysis of the space of variables, a swarm size of ten times the number of variables was chosen, to ensure sufficient exploration of the space of variables. The maximum number of iterations was set to 20 — a compromise between accuracy and execution time. For minimum step and minimum function, a very low value was set (10^{-20}), as a way of guaranteeing all iterations would be calculated without early algorithm stop for further exploration. The parameters w, c_1 and c_2 from Equation 2.31 were defined from *pyswarm's* default values, as listed in Table 3.6.

The default *pyswarm* implementation of PSO was found to be not suitable for this work for two reasons:

Lower and upper bounds: the default implementation considers that the lower and upper bounds of each variable are constant throughout all iterations. However, in this work those values might change at

Table 3.6: PSO default parameters (Equation 2.31) implemented in Python *pyswarm* module and used in this work.

PSO parameter	Value
w	0.5
c_1	0.5
c_2	0.5

each function evaluation and need to be updated constantly, otherwise the algorithm will repeatedly generate a set of variables outside the feasible region. Details on that are described in Section 3.4.3.

Particle generation: the default implementation generates the values of all decision variables at once. However, in this work there is a dependency between the upper bounds of continuous variables and the binary decision variables. Therefore, the latter (y) needs to be generated first so that the upper bounds of u are calculated and the continuous variables are generated within a feasible range.

For those two reasons, the *pyswarm* PSO code was modified to fit this work's specificities. Without those adjustments, no feasible points were found by the method during the execution attempt of relaxed-integrality problem. The modified code is presented in Appendix A.

Local NLP Optimization: Nelder-Mead Flexible Polyhedron

Default values consolidated in the literature (GAO and HAN, 2012) were used for reflection, expansion, contraction and reduction parameters, as presented in Table 3.7.

Table 3.7: Nelder-Mead method default parameters implemented in Python *scipy.optimize.minimize* module and used in this work.

Nelder-Mead Operation	Parameter value
Reflection	$\rho = 1$
Expansion	$\chi = 2$
Contraction	$\psi = 0.5$
Reduction	$\sigma = 0.5$

It is important to mention that it was necessary to study and define an initial polyhedron to improve the performance of the Nelder-Mead method. Further details will be presented in Section 3.4.5.

MIP Optimization: Branch-and-Bound

The branch-and-bound code utilized was customized for this work and uses a breadth-first implementation via tree-search. The modification was performed with the goal of allowing external calculation of the objective function (via HYSYS). Furthermore, it was necessary to couple the code to Nelder-Mead Scipy algorithm. In summary, branch-and-bound builds the tree of binary variables, defines the nodes and then calls Nelder-Mead flexible polyhedron every time an NLP subproblem needs to be solved.

Proposed Strategy

The methods described in this section will be used together in this work, in order to solve the formulated optimization problem. Figure 3.7 wraps up the methodology proposed: PSO global method is used in a pre-screning step to generate a set of initial estimates for the MINLP Problem; branch-and-bound and Nelder-Mead flexible polyhedron are employed in a loop in order to find an MINLP solution via tree-search; lastly, PSO and Nelder-Mead methods are utilized in series with tighter tolerances, aiming to refine the MINLP result to achieve a final solution.

The refinement step shown in Figure 3.7 was added to the optimization framework of this work in order to promote a final execution of PSO and Nelder-Mead to refine the results encountered by the MINLP algorithm. This was achieved by utilizing tighter tolerances for both Nelder-Mead method and in the simulation recycle convergence. In this post-refinement step, the binary variables were fixed at the same values defined by the MINLP solution and only u are set as decision variables. The MINLP solution is passed to the post-screening NLP PSO as one of particles that would be generated in the first iteration of the method — therefore, from the 120 total particles, 119 were randomly generated by the algorithm and 1 was taken from the MINLP solution. This guarantees that PSO considers the MINLP solution point in its execution. The result of NLP PSO is then used as initial estimate in a final Nelder-Mead execution. Further details on the tolerance values utilized will be given in Chapter 4.

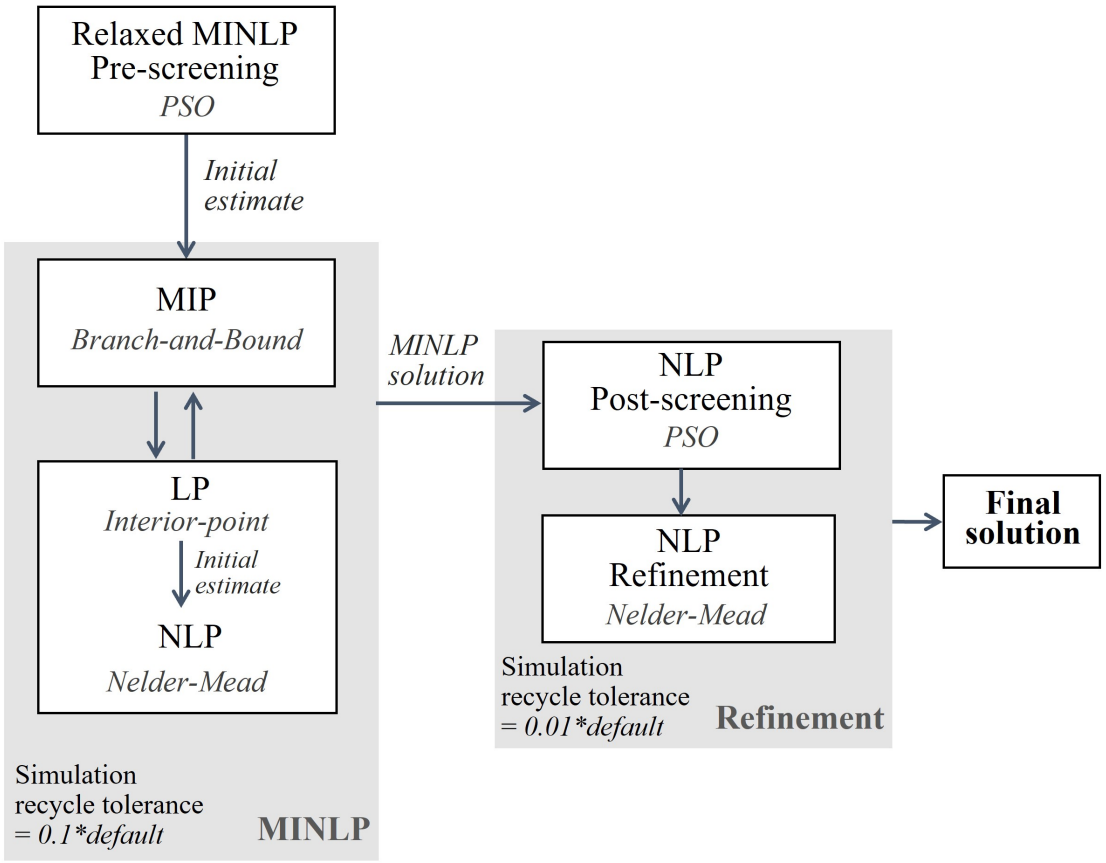

Figure 3.7: Overall schematic of the methodology proposed in this work. PSO, branch-and-bound and Nelder-Mead methods are employed in three main steps: pre-screening, MINLP solution and post-refinement.

3.4.2 The Optimization Problem Formulation

The proposed optimization problem has the goal of determining optimal feed allocation and NGPUs operating status in order to maximize total business profit for a given set of raw gas input.

The Objective Function

The objective function is defined as the total variable business profit. Profit is calculated as total revenue minus operating expenses (OpEx):

$$Profit\ [R\$/d] = Revenue - OpEx \tag{3.1}$$

The first term, $Revenue$, is the total income from products sale. For each product, a typical market price was considered, as shown in Table 3.8.

An economic assessment was performed in order to bring together a representative

Table 3.8: Typical market price for sales gas, raw gas for external processing, NGL, LPG and C_5^+. Values used to calculate business revenue. Market values from the week of 15 to 21 of May 2021.

Product	Market Price (R$/MMbtu)
Sales gas	25.40
Raw gas for external processing	21.35
NGL	42.15
LPG	57.94
C_5^+	65.91

expression for the *OpEx* term. Strictly speaking, *OpEx* is a function of all costs involved in the operation of the gas processing site: equipment, electricity, chemicals, labor, maintenance, utilities and so on. However, among those there are a few that stand out as major contributors. They are:

1. Electricity (EE)
2. Fuel gas (FG)
3. Chemical products (CP)

The other contributions were disregarded, either for having low overall importance or for being understood as roughly fixed costs, which values do not change regardless of the set of decision variables — labor costs, for instance.

Total *OpEx*, in $R\$/d$, can therefore be estimated as the sum of three contributions, as seen in Equation 3.2.

$$OpEx = Electricity + FuelGas + ChemicalProducts \tag{3.2}$$

The following subsections will describe each of these three terms.

Electricity Demand

Demands for both electricity (MW) and fuel gas (m^3/d) were calculated inside HYSYS, i.e., they are simulation model outputs, since they can be estimated from mass and energy streams. Table 3.9 lists all types of compressors, per process unit, considered for electricity consumption estimation.

The total electricity demand ($Power_{EE}$) is, therefore, calculated as the sum of each energy flowrate from a respective HYSYS energy stream associated to each of

Table 3.9: List of motocompressors, by process unit, accounted for in the estimation of total electricity demand.

Type of compressor	NGPU-A	NGPU-B/C/D	NGPU-E	LFUs	Raw gas bypass
Propane cycle	X	X	X		
Residual gas recirculation	X		X	X	
Sales gas booster		X			
Molecular sieve regeneration		X			
Raw gas bypass					X
NGPU feed gas from slug catcher		X	X		

the compressors listed in Table 3.9. Therefore, the energy required to operate each of those compressors varies with the manipulated variables values. Actual operational equipment efficiencies were added to the simulation model for more adherent results. Values were gathered from historical process data and a total of 22 pieces of equipment were considered.

Fuel Gas Consumption

Fuel gas was assumed to have the same properties from average sales gas, since typically part of the processed gas is diverted for this use. The major contributors are:

1. Furnaces
2. Sales gas turbo-compressors

Table 3.10 lists the 16 furnaces considered, per process unit. One single turbo-compressor represented the sales gas compression station. Actual operational efficiencies from historical process data were used for all furnaces and turbo-compressors for better fit to the industrial plant data.

Table 3.10: List of furnaces, by process unit, accounted for in the estimation of total fuel gas consumption.

Type of furnace	NGPU-A	NGPU-B/C/D	NGPU-E	LFUs	CO_2 removal unit
Demethanizer column		X			
Deethanizer column			X	X	
Debutanizer column			X	X	
Molecular sieve regeneration		X			
Amine regeneration					X

Fuel gas demand per consumer in energy base ($Power_{FG_i}$) was calculated from HYSYS energy streams and average operational equipment efficiencies ($\bar{\eta}_i$). Using the lower heating value of sales gas (LHV_{SG}) as a reference for fuel gas LHV, total fuel gas volumetric flowrate consumption (Q_{FG}) is calculated from the expression in Equation 3.3, by summing the contribution of all n_{FG} fuel gas consumers.

$$Q_{FG} = \sum_{i=1}^{n_{FG}} \frac{Power_{FG_i}}{\bar{\eta}_i \, LHV_{SG}} \tag{3.3}$$

Chemical Product Spending

For chemical consumption estimation, the four most relevant chemical products utilized by the process units of interest were selected. They are listed in Table 3.11, by process unit, along with their specific cost per feed flowrate, which are average historic values and were gathered from a Petrobras internal report.

Table 3.11: List of chemical products, by process unit, accounted for in the estimation of total chemical product spending. The average values reported utilize unit price data from a Petrobras internal report. Values are presented in $R\$$ per million cubic meters.

Chemical product	NGPU-A	NGPU-B/C/D	NGPU-E	LFUs	CO_2 removal unit
Alkanolamine solution	-	-	-	-	481.79
Propane	4613.39	-	1590.82	-	-
Molecular sieve	-	137.17	-	-	-
Activated carbon	-	-	-	-	92.20

The total cost associated to chemical product consumption per process unit was then estimated from the values in Table 3.11 times the simulated feed flowrate of the respective NGPU (Q_{NGPU}). This total cost is therefore a value that changes with the manipulated variables.

Other chemical products were disregarded either due to low overall contribution or for having non-variant costs regardless of the decision variables — H_2S removal bed, for instance, needs to be replaced periodically due to regulatory vessel inspection enforcement, even if it has not been used.

The Optimization Problem Equations

The optimization problem can then be introduced as follows:

$$\max \left\{ \bar{\bar{S}} \left[x(u,y), u, y \right] \right\} \tag{3.4}$$

with,

$$\bar{\bar{S}} \left[x(u,y), u, y \right] = \sum_{i=1}^{n_p} (m_i \, Qp_i) - \left[c_{EE} \, Power_{EE} + c_{FG} \, Q_{FG} + \sum_{j=1}^{n_{CP}} \left(c_{CP_j} \, \bar{Q}_{CP_j} \, Q_{NGPU_j} \right) \right] \tag{3.5}$$

where n_p is the number of plant products ($n_p = 5$: sales gas, raw gas for external processing, NGL, LPG and C_5^+), m_i is the market price vector for each product (values from Table 3.8), Q are flowrates in m^3/d (simulation outputs, function of u and y) and the subscripts p and FG refer to plant products and fuel gas, respectively. Vector c represents average costs of electricity (c_{EE}), fuel gas (c_{FG}) and chemical products (c_{CP}). For the first two, the typical values were considered for the Brazilian market are shown in Table 3.12. For chemical products, the unit price (c_{CP}) is multiplied by the average consumption of the respective chemical product per gas feed ($\bar{Q}_{CP}$). Table 3.11 presents the values of the product $c_{CP}\bar{Q}_{CP}$ per chemical product, in R$ per million cubic meters.

Table 3.12: Typical unit costs of electricity and fuel gas for Brazilian industry.

Utility	Typical Cost (c)
Electricity	406.73 $R\$/MWh$
Fuel Gas	0.60627 $R\$/m^3$

subject to:

$$\begin{aligned} x(u,y) \in \{ x \in \mathbb{X} \subseteq \mathbb{R}^{n_x} \mid h_j \left[x(u,y), u, y \right] = 0 \, , \; j = 1, 2... \, n_h \} \\ g_j[x(u,y), u, y] \leq 0 \; j = 1, 2... \, n_g \\ u \in \mathbb{U} \subseteq \mathbb{R}^{n_u} \\ y \in \mathbb{Y} \subseteq \mathbb{N}^q \end{aligned} \tag{3.6}$$

In this equation, u and y are decision variables, specified inputs for the simulation: u are continuous whereas y are binary. x are the model variables, output from the simulation, used to calculate the objective function. It is important to notice that the continuous decision variables u are normalized values between zero and one (more details

in Section 3.4.3). This normalization was a strategy to equalize the order of magnitude of all variables, aiming for better performance of the optimization method.

$h_j[x(u,y),u,y]$ are equality constraints, i.e., mass and energy balances, thermodynamic calculations and other correlations used to compute physical and chemical properties. $g_j[x(u,y),u,y]$ are inequality constraints, which are associated to upper and lower physical bounds of the NGPUs, product quality ranges and market and logistics restrictions.

However, considering the integrated simulation-optimization framework with the feasible path approach employed in this work, h_j are already computed by the process simulation and only g_j need to be accounted for by the optimization module. This approach solves the optimization problem in two levels: the process simulator ensures the equality constraints are satisfied, calculates the model variables $x(u,y)$ and sends them to Python optimizer via Excel. Python-implemented algorithms calculate the objective function and the optimization method determines the next set of decision variables (u and y). Those variables are then sent back to the simulator via Excel. Therefore, the reduced-order problem has u and y as optimization variables, opposed to x, u and y from the original problem.

Also, the objective function $\bar{\bar{S}}$ was divided by a constant of reference, S_{ref}, according to Equation 3.7, in order to work in the same order of magnitude from the decision variables.

$$\bar{S} = \frac{\bar{\bar{S}}}{S_{ref}} \tag{3.7}$$

In the light of above considerations, the optimization problem is then rewritten for a simulation-optimization framework with bounds and constraints incorporated via penalty function (Pen). The resulting augmented objective function (S) incorporates upper and lower bounds plus the inequality constraints and convergence issues as penalties.

$$S[x(u,y),u,y,r] = \bar{S}[x(u,y),u,y] - Pen[x(u,y),u,y,r] \tag{3.8}$$

with the penalty function given by Equation 3.9.

$$Pen[x(u,y),u,y,r] = \sum_{j=1}^{n_g} r_j \max\{0, g_j[x(u,y),u,y]\} + FlagPen_{conv} \tag{3.9}$$

$S[x(u,y),u,y,r]$ represents the augmented objective function, $Pen[x(u,y),u,y,r]$ is the penalty function and r is the scale factor vector, used to equalize the magnitude of each penalty contribution — a unit vector was utilized, since all the decision variables were previously normalized. The term Pen_{conv} is the penalty for non-convergence and accounts for the cases in which the simulation model does not converge. A fixed value was used in all cases, with $Pen_{conv} = 0.1786$. This value was defined after successive executions, in order to guarantee numerical stability. Lastly, $Flag$ is a flag that indicates whether the simulation has converged ($Flag = 0$) or not ($Flag = 1$).

The incorporation of inequality constraints into the objective function via penalties transformed the original constrained model into an unconstrained problem, which can be solved using unconstrained derivative-free optimization methods, such as Nelder-Mead flexible polyhedron, chosen in this work.

S (Equation 3.8) is the form of the objective function that will be utilized in the following steps, results and discussion of this work — hereafter referred only as *objective function*. The actual values are recovered when a quantitative financial analysis of optimization potential gain is performed in Section 4.3.4.

3.4.3 Decision Variables

u and y are the decision variables of the optimization problem. u represents all continuous variables, which, in physical terms, are the 12 gas processing site main flowrates. They are represented in Figure 3.2 and include:

1. NGPUs raw gas feed (u_1 to u_{10}).
2. Reprocessed fraction of the residual gas flowrate (u_{11} to u_{12}).

All of the continuous u variables are normalized in a range from 0 to 1 in order to improve algorithm performance. To perform this variable change, their lower (Q^{min}) and upper (Q^{max}) bounds were used, according to Equation 3.10.

$$u = \frac{Q - Q^{min}}{Q^{max} - Q^{min}} \tag{3.10}$$

Q^{min} is a constant for all continuous variables except Q_1. For this variable, it is a function of Q_2 and Q_8: $Q_1^{min} = f(Q_2, Q_8)$. For all NGPUs, the lower bound is 50 % of their design capacity. On the other hand, Q^{max} is only constant for Q_1, Q_{11} and Q_{12}. For all other variables the upper bound is a function of the corresponding binary decision variable y, according to Equation 3.11.

$$Q^{max} = K\,y \tag{3.11}$$

in which K is the design capacity of each NGPU. This dependency creates a challenging scenario for the optimization problem, especially when it comes to the correlation between Q^{max} and y. This means that each time y changes, the feasible range of the continuous variables also changes and Q^{max} need to be recalculated. For the optimization problem, this happens at every function evaluation. To overcome this issue, y needed to be specified first, then Q^{max} is calculated and u is determined from Equation 3.10.

y represents the 7 binary decision variables, which are the NGPUs operating status (y_1 to y_7). For the integer variables, binary 0-1 values are set according to the following meaning: 0 means that the respective NGPU is in standby and 1 refers to NGPU in operation.

For Nelder-Mead method and PSO executions of the relaxed problem, y should assume fraction values. As discussed before, those cases lead to Q^{max} varying at each function evaluation as a function of y. When assuming fraction values, y needs to be limited to guarantee feasible conditions for the simulation. This generates a discontinuity in $y = \frac{Q^{min}}{Q^{max}}$. For instance, when Q^{min} is 50 % of Q^{max} value, y is continuous from 0.5 to 1. For $y < 0.5$ the corresponding NGPU is in standby and $Q^{max} = Q^{min} = 0$, since the region of $0 \leq Q \leq Q^{max}$ is not feasible. This discontinuity is another great challenge to the optimization algorithm, since the method needs to surpass this considerable unfeasible region in order to find feasible points, corresponding to a given NGPU in standby ($y = 0$). The strategy employed in this work is detailed in Section 3.4.5.

3.4.4 Inequality Constraints

The optimization problem is subject to constraints that might come from physical limitations, market and legislation restrictions, logistics bottlenecks and so on. Those are incorporated in the model as inequality constraints, summed to the objective function in the form of penalties.

This model has a total of 63 inequality constraints (g_1 to g_{63}), as listed below:

1. **Upper and lower bounds: continuous decision variables** (g_1 to g_{24})
   The non-normalized continuous decision variables are limited by design specifications.

$$Q_i^{min} \leq Q_i \leq Q_i^{max}\,, i = i, ..., 12 \tag{3.12}$$

2. **Upper and lower bounds: integer decision variables** (g_{25} to g_{38})
   The integer decision variables range between 0 and 1 when relaxed.

$$0 \leq y_i \leq 1\,, i = 1, ..., 7 \tag{3.13}$$

3. **Upper bounds: dependent flow variables** (g_{39} to g_{42})

   NGPU-C (g_{39} and g_{40})

$$Q_{NGPU_C}^{min} \leq Q_{NGPU_C} \leq Q_{NGPU_C}^{max} \tag{3.14}$$

   NGPU-E (g_{41})

$$Q_4 + Q_5 + Q_6 + Q_7 \leq Q_7^{max} \tag{3.15}$$

   SG-B to SG-A flowrate (g_{42})

$$Q_1 + Q_5 \leq Q_1^{max} \tag{3.16}$$

4. **Market demand** (g_{43})
   NGL volume flowrate needs to guarantee minimum supply of external sales contract.

$$Q_{NGL} \geq Q_{NGL}^{min} \tag{3.17}$$

5. **Logistics bottleneck** (g_{44})
   LPG volume flowrate is limited by logistics capacity for product transfer from storage.

$$Q_{LPG} \leq Q_{LPG}^{max} \tag{3.18}$$

6. **Quality restrictions** (g_{45} to g_{63})
   Sales gas, LPG and NGL are limited by regulatory quality ranges. These restrictions

are extremely important, since product specifications must be met before it goes to the market.

Sales Gas (g_{45} to g_{59})

RANP 16/2008 specifications, listed in Table 2.1

LPG (g_{60} to g_{62})

LPG ethane content is limited by internal Petrobras technical restriction and pentane content and density upper limits are bounded by ANP regulation RANP 835/2020.

Internal plant limitation:

$$x_{C_2}^{LPG} \leq 11.0\ volume\ \% \tag{3.19}$$

RANP 825/2020 specifications (ANP, 2020):

$$x_{C_5}^{LPG} \leq\ 2.0\ volume\ \% \tag{3.20}$$

$$\rho^{LPG} \leq\ 575.0\ kg/m^3 \tag{3.21}$$

NGL (g_{63})

A minimum ethane content on NGL is required by the design of downstream fractionation facilities.

$$x_{C_2}^{NGL} \geq 40.0\ volume\ \% \tag{3.22}$$

Those constraints have different characteristics when it comes to their implementation in the optimization problem. Considering that, they were divided in three different categories, according to how they were dealt with during optimization. Table 3.13 describes the three types, which constraints fit in each of them and how they were handled.

Table 3.13: Description of the three types of inequality constraints categorized by their distinct treatment during implementation. Table also details how each type was handled and which constraints fit in each category.

Type	Number	Handling
Decision-variable-related bounds	g_1 to g_{24} g_{25} to g_{38} g_{41} g_{42}	Pre-analyzed. If not met, HYSYS is bypassed and is not called to execute the simulation model. A fixed value of zero is assigned to the objective function in those cases.
Other bounds	g_{39} to g_{40}	Post-analyzed. Used for constraints which are frequently active however need an output of the simulation to be verified. Check is done via Excel after the model is run. If not met, a penalization is subtracted from the already calculated objective function (Equation 3.9).
Market, logistics and quality restrictions	g_{43} g_{44} g_{45} to g_{63}	Post-checked. Used for constraints which are rarely active. Verification is done via Excel after the optimum is found. If not met, the result is discarded.

3.4.5 Implementation Strategies

Nelder-Mead Flexible Polyhedron: Initial Polyhedron

Nelder-Mead method uses an initial polyhedron (sim) as a starter shape for space exploration. For a problem of *N* dimension, the initial polyhedron has *N+1* vertices around initial estimate vertex (w_0). Therefore, sim is bi-dimensional with shape $sim(N+1, N)$, representing $N+1$ vertices, each with length N. The first vertex is equal to the initial estimate, $sim(0,:) = w_0$. The remaining vertices revolved around w_0 and the size of the initial polyhedron is set by a user-defined step-size ($\delta(j)$). Figure 3.8 shows a general flowchart with the steps for initial polyhedron calculation.

Python *Scipy.optimize* implementation of Nelder-Mead algorithm has defined fixed values for $\delta(j)$, as stated in Equation 3.23.

$$\delta(j) = 0.05 \; for \; w_0(j) \neq 0 \qquad \delta(j) = 0.00025 \; for \; w_0(j) = 0 \tag{3.23}$$

However, those default δ values were found out to be non-adequate for this work. Using default values, the initial polyhedron generated was too small and led the optimization algorithm to get trapped locally, with not enough exploration of the space of variables.

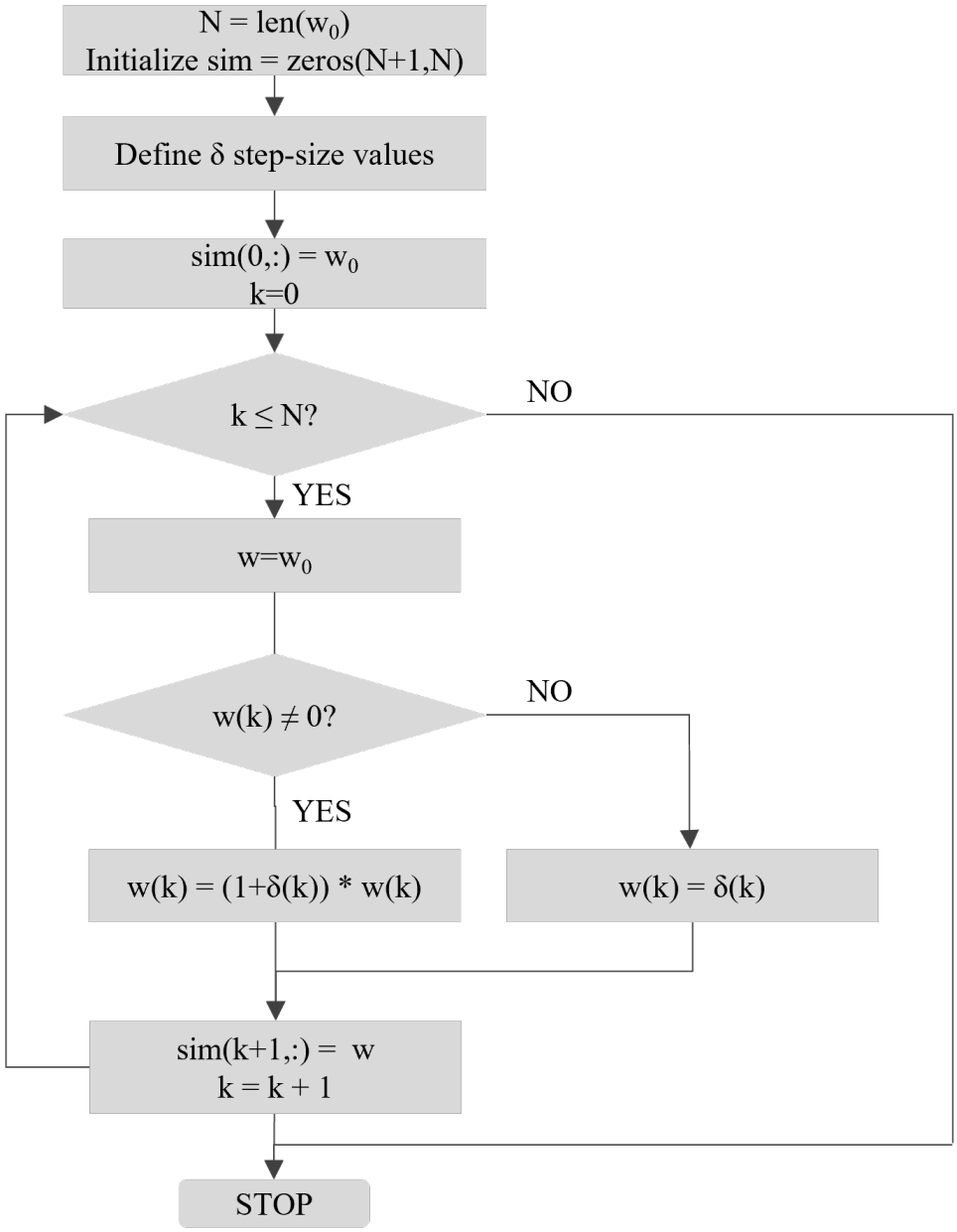

Figure 3.8: Flowchart showing the steps carried out for initial polyhedron determination.

Therefore, a modification was proposed in the δ parameter, according to Equation 3.24.

$$\delta(j) = 0.25 \; for \; w_0(j) \neq 0 \qquad \delta(j) = 0.00025 \; for \; w_0(j) = 0 \tag{3.24}$$

Generating Initial Estimates for Nelder-Mead Method: a Linear Programming (LP) Approach

As mentioned before, Nelder-Mead flexible polyhedron is a local optimization method and starts exploration from an initial polyhedron, generated from an initial estimate input.

Guaranteeing a feasible initial estimate is therefore an important step for proper use of the method.

In the MINLP problem, the feasible region of the continuous decision variables depend on the values of integer variables. This means that for each set of y there are specific feasible values for u. For a non-convex problem, this also means that when y changes, the problem might shift to an entirely different region of the space of variables. For instance, if one NGPU is put on standby ($y = 0$), then its raw gas feed must be distributed between one or more of the other NGPUs under operation, changing substantially the operating condition (u).

In order to overcome this complex scenario, an LP problem was formulated to generate an initial estimate for the Nelder-Mead method. The LP ensures that the initial estimate will be feasible, providing a starting point to the method by incorporating the inequality constraints regarding the upper and lower bounds of the continuous decision variables — g_1 to g_{24} and g_{39} to g_{42} (Table 3.13).

The LP implementation was also done in *Python*, using the *scipy.optimize.linprog* module that utilizes the interior-point method of ANDERSEN and ANDERSEN (1999) to solve the problem. The result is an initial estimate array (w_0). The decision variables vector corresponds to the continuous decision variables of the MINLP problem (u). The LP problem is solved for a fixed given set of y that is user-defined. Different baselines of y were defined — if no feasible LP solution was found, then y was gradually increased up to its upper limit, one.

Also, the objective function (f_{LP}) of the LP problem was defined with the addition of a slack variable in one of the main inequality constraints of the problem: the lower bound of NGPU-C (g_{39}), defined in Equation 3.14). Keeping in mind that NGPU-C feed rate is a dependent variable, it was written as a function of the gas feed of the other four NGPUs (Q_2 to Q_{10}), the LP objective function was expressed from Equation 3.25:

$$f_{LP} = -\sum_{i=2}^{10} Q_i \tag{3.25}$$

in which f_{LP} expresses the total feed flowrate of NGPUs-A/B/D/E and was minimized, considering as restrictions the upper and lower bounds of the NGPUs. Further details can be seen in the implemented coded presented in Appendix B.

This strategy ensured that, whenever possible, Nelder-Mead method would have a feasible initial estimate generated from an automated approach. Thus, this implementation presents itself as a contribution for this work, since it reduces the need of manual generation of initial estimates by knowledge of the plant.

Penalties: An Implementation Strategy to Avoid Simulation Convergence Issues

Since the constrained optimization model was reformulated into an unconstrained problem with the addition of penalty functions, it is possible that the optimization algorithm sets values for the decision variables that are outside a feasible range for the simulation. Penalty will make sure that this scenario is not attractive, by reducing the value of the objective function.

However, another issue might come up: unfeasible values of u might and will probably cause problems in modeling convergence. HYSYS might spend significant time trying to converge the simulation and even fail to do so, in which case might even compromise all subsequent simulation attempts. The workaround implemented is to check if decision variables are within bounds a priori and only run the simulation if this is true.

Still, there might be scenarios in which the simulation has trouble converging. For those cases, a specific convergence penalty term is added to the objective function. The VBA macro checks for simulation convergence. If non-convergence is detected, a penalty in the same magnitude order of the profit is used in the objective function S, in order to inform the optimization algorithm that this is not an interesting operating configuration.

Chapter 4

Results and Discussion

This chapter will present and discuss the modeling and optimization results obtained in this work. For modeling, this includes validation and tuning steps. On optimization, will be presented the NLP problem results, an analysis of space of variables utilizing PSO, the solution of the MINLP problem and discussion on the proposed methodology. Lastly, a case study is performed in order to evaluate the impact of changing product prices in a simulated scenario of high sales gas market value.

4.1 Process Modeling and Simulation

The process model of the natural gas processing site was built up in HYSYS v11 and was run in an Intel ® Core^{TM} i5 10210U processor with 1.60 GHz. The simulation comprises all three slug catchers (SG-A/B/C), five NGPUs (NGPU-A/B/C/D/E), three liquid fractionation units (LFUs), compression systems and auxiliary equipment, as well as a simplified representation of gas treating units (mercury, CO_2 and H_2S removal). It also includes MEG injection and molecular sieve dehydration and regeneration. Table 4.1 shows the values of molar composition, temperature and pressure of the three simulation inlet streams, which correspond to the entrance of slug catchers SG-A/B/C (details in Figure 3.2).

Table 4.1: Molar composition and process conditions (temperature and pressure of the three simulation inlet streams, which correspond to the entrance of slug catchers SG-A/B/C).

Property	Component	SG-A	SG-B	SG-C
Composition (mol %)	Methane	80.00	79.18	75.79
	Ethane	8.70	8.97	11.33
	Propane	5.29	5.70	6.58
	Isobutane	0.84	0.82	1.03
	N-butane	1.55	1.61	1.80
	Isopentane	0.37	0.38	0.33
	N-pentane	0.46	0.49	0.37
	N-hexane	0.34	0.35	0.19
	N-heptane	0.27	0.28	0.04
	N-octane	0.20	0.20	0.01
	N-nonane	0.13	0.13	0.00
	N-decane	0.05	0.05	0.00
	Nitrogen	0.73	0.71	0.79
	Carbon dioxide	1.07	1.12	1.73
	Water	0.00	0.00	0.00
	H_2S	0.00	0.00	0.00
	MEG	0.00	0.00	0.00
Temperature (^oC)		25	25	18
Pressure (bar)		57.87	77.49	88.27

The simulation model has a total of 1248 material streams, 167 energy streams, 107 heat exchangers, 65 recycles and 18 distillation columns. It is the rigorous digital representation of an industrial site able to process over one quarter of all natural gas used in Brazil. Building up and adjusting a process simulation of this size was challenging. Since the simulation model was constructed to be coupled with the optimization algorithm, it was necessary to assure not only model convergence, but also stability, fast execution time, proper communication to Microsoft Excel and previous treatment of extreme scenarios to avoid structural convergence issues and misleading of the optimization algorithm. All those points required several and repeated adjustments in the model until it was running smoothly enough to be coupled with the optimization algorithm utilized in this work.

In the literature, most previous works such as BULLIN and HALL (2000); FRANCO *et al.* (2020); MAZUMDER and XU (2020); MONDAL *et al.* (2013); MURALI *et al.* (2020) limited their rigorous modeling to only one NGPU, whereas the few who looked into multiple-unit sites opted for non-rigorous simplified models (ZHANG *et al.*, 2016), which shows the complexity of building a rigorous model of this size.

Even though challenging, this work managed to successfully build a rigorous simulation model of a multi-unit and multi-feedstock industrial site for natural gas processing and to use this simulation to provide accurate model results for the optimization algorithm at every function evaluation of the optimization loop implemented in this work. The model was proven to be stable enough to stay running non-stop for days, without numerical convergence issues.

Furthermore, it was noticed that previous technical knowledge of the industrial site was important to make adjustments in the simulation such as choosing the operating modes of the NGPUs — for instance, the NGPU temperature profile can be modified in order to recover more liquid fractions (lower temperatures) or to yield more gas (higher temperatures). For future works, it is possible to automate this step by incorporating those as decision variables in the optimization model, which would have more binary manipulated variables.

Lastly, by implementing and testing the model exhaustively, it was found out that execution time had a strong dependence on input conditions: if they were similar to the previous converged run, average time ranged from 40 to 60 seconds. For input conditions different from the previous converged run, running time took up to 2.5 minutes. This happens because HYSYS utilizes the previous converged solution as initial guess for the next run, which helps convergence if the sequential scenarios are similar. The results presented here were obtained using a recycle flowrate tolerance ten times tighter than HYSYS default value — a different approach would lead to different execution times. This value was defined after several analyses, detailed in Section 4.3.1, which verified that when this tolerance is above a certain limit, numerical oscillations arose in the simulation causing disturbances in the local optimization method.

4.1.1 Model Validation and Tuning

Once the challenges were overcome and the modeling was successfully completed, it was possible to validate the simulation results by comparing them against operation

process data. This was also a challenging step, due not only to the size and complexity of simulation but also to the need of acquiring reliable historical data.

The information chosen were mass and energy balances, both global and per process unit. The strategy employed was to use monthly-average historical plant data, since those were the best available information and are reconciled for better representation. These include data treatment for unavailable instruments by usage of measurement redundancies or calculation by inference and individual check of mass balances per process unit.

For this work's purpose, the most important values to be compared are the four overall product flowrates (sales gas, NGL, LPE and C_5^+), once they are used to calculate the objective function (S). Therefore, the model was validated by comparing flowrate values of the four final products: sales gas, NGL, LPG and C_5^+. Even though no detailed comparison by each process unit is presented here, this strategy achieved its high-level purpose, even if there were some compensation between units.

It is important to notice that even though the validation was performed on overall mass balance, the data from each process unit was also used during the process of model validation: this information was crucial to adjust the main process variables that define how the process unit is operating and, by consequence, their performance and product profile. Most of those variables were flowrates, including the continuous decision variables defined for this work (u_1 to u_{12}). The following variables were adjusted in the simulation file based on plant monthly data: feed flowrate per process unit; distribution of residual gas from NGPU-A and NGPU-E (reprocessing or sales gas); LPG ethane content; operating mode of the NGPUs; NGL return flowrate from storage to the LFUs for fractionation into LPG and C_5^+; fuel gas and flare loss assumptions. Lastly, occasional adjustments were also made in the overall temperature profiles and distillation column specifications. A base-case simulation file was used as starting point and specifically adjusted for each point of model validation.

The first quarter of 2020 was chosen for data acquisition, since it was a period in which the plant was relatively stable. Table 4.2 shows the relative error on product flowrate for the months of January, February and April. The month of March is not listed because its available data differed substantially from regular plant operation due to intermittent plant unavailability and was considered non-representative. In the three months used for validation, all NGPUs were operating, i.e., all the integer decision variables were known and constant ($y = 1$).

The comparison from Table 4.2 shows good agreement, with average discrepancies

Table 4.2: Model validation results: comparison between simulated and actual product flowrates for monthly input data.

	Relative error for product flowrate (%)			
Month of 2020	Sales Gas	NGL	LPG	C_5^+
January	0.30	1.08	-1.57	1.51
February	0.29	1.53	-0.68	3.97
April	0.27	-4.32	-7.17	-1.58
Average	0.29	-0.86	-3.14	3.90

under 4.0 %, measured in relation to product flowrates, the main variables of interest in this work. All the calculated errors were below 10 % in absolute values. The highest deviations were for C_5^+ and LPG, which are the streams produced in the lowest amount. Sales gas has systematically lower deviation percentages due to its high flow.

From the results presented, the simulation model was considered adherent to plant reality and validated for this work's purpose. The simulation base-file — starting point for the validation approach — was then chosen as a general representation for the subsequent optimization study. Thus, the model considers typical operating conditions for each process unit, which makes the model representative, robust and well-suited for coupling with the optimization model.

The good agreement between model and experimental data corroborates the choice of HYSYS as simulation tool and of Peng-Robinson EoS as the model fluid package with COSTALD correlation for liquid density calculation. It also encourages the next steps of this work.

4.2 The Simulation-Optimization Integration Tool

The integration tool was coded in VBA and also uses Microsoft Excel spreadsheet for further calculations and data visualization. It was built in a way that it has become a hub for visualization of the main parameters of the simulation and optimization.

The main simulation outlets — such as product flowrates, compositions and HHV, as well as estimated fuel gas and electric energy consumption — are presented in the simulation tab. It also contains several operational parameters that are input for the simulation model, however not considered as decision variables in this work. For those,

fixed values are used in all executions. It is this tab that makes all the communication between Excel and HYSYS.

All current values of the continuous and integer decision variables are presented in the optimization tab, as shown in the partial-tab screenshot of Figure 4.1. It is also where all the terms of the objective function are calculated: revenue and OpEx (electricity, fuel gas and chemical products cost). Lastly, it is also this tab that contains the analysis of simulation convergence and the inclusion of penalty terms, when necessary, as well as the post-check of quality, logistics and market constraints.

		Bounds		
Decision variable		Lower	Upper	Current value
Continuous	**u1**	0	1	0.37
	u2	0	1	0.50
	u3	0	1	0.52
	u4	0	1	0.52
	u5	0	1	0.52
	u6	0	1	0.50
	u7	0	1	0.68
	u8	0	1	0.00
	u9	0	1	0.00
	u10	0	1	0.43
	u11	0	1	0.63
	u12	0	1	0.50
Binary	**y1**	0	1	0
	y2	0	1	1
	y3	0	1	1
	y4	0	1	0
	y5	0	1	1
	y6	0	1	1
	y7	0	1	1

Figure 4.1: Example of information contained in the optimization tab of the simulation-optimization integration tool. Data contains lower/upper bounds and current value of all decision variables.

4.3 The Optimization Problem

The solution of the optimization problem was broken down into four main steps: 1) the NLP problem; 2) analysis of the space of variables with PSO; 3) the MINLP problem and 4) MINLP results refinement. Their results will be presented and discussed in the next subsection, in this same order.

The analysis of the space of variables was a previous step utilized to better understand the MINLP problem and the behavior of the variables along their feasible region. Then, the NLP was implemented and solved for the twelve continuous decision variables, which was also an opportunity to make fine adjustments in the simulation model and in its integration with the optimizer. Only then the integer decision variables were added to the problem and the entire MINLP problem was solved. Lastly, the MINLP results were refined by running a more accurate execution of PSO and Nelder-Mead flexible polyhedron. To wrap up the evaluations, a case study was performed, utilizing different product prices from the base-case, in order to compare and analyze the results obtained in a scenario with high sales gas price.

From here on, for the correct analysis of the results, it must be kept in mind that, as mentioned previously, the product prices established by the values of Table 3.12 reflect a scenario in which liquid products are more valuable than sales gas. Therefore, is it expected that any scenario that reduces liquid recovery should be disfavored by the optimizer. This will only change when the case study is presented (Section 4.4).

4.3.1 The NLP Problem

The NLP formulation of the problem was solved for the continuous decision variables (u_1 to u_{12}): raw gas feed (u_1 to u_{10}) and reprocessing fraction of residual gas flowrates (u_{11} and u_{12}). $c_{CP_j}\,\bar{Q}_{CP_j} = 0$ and all binary variables (y) were set to 1 in this phase of the study, meaning that all process units represented in the simulation were under operation in this scenario.

This was the first step of the optimization framework solution and was used to evaluate potential financial gain from the NLP approach. It was also where implementation feasibility was verified and general tests were made in terms of: software communication, simulation robustness in extreme input conditions, fine adjustments in the simulation structure to avoid non-convergence and software crash.

Nelder-Mead Method and Initial Estimates

As mentioned before, the Python algorithm was run with *Scipy.optimize.minimize* package and Nelder-Mead flexible polyhedron method with custom initial polyhedron. It was employed an absolute tolerance of 0.005 for both normalized decision variables and objective function. Average running time was 3 hours. For tighter optimization tolerance values, running time got excessively long (over 12 hours), which would be unfeasible for short-term planning. Besides, this level of refinement led to no apparent expressive gain for this planning problem. Since natural gas processing complexes deal with very dynamic scenarios that change in a daily basis, excessive running time looking for further result refinement might mean that when solution is available, plant reality has already changed and input conditions are different from the ones considered in the optimization problem, which is not desirable. Therefore, a tolerance of 0.005 was considered suitable for the purposes of this work, representing a fine balance between accuracy and execution time.

Five different sets of initial estimate values were used to study the space of variables (named I01, I02, I03, I04, I05). They are all feasible points and were defined based on actual operating conditions that are typical for this industrial plant and that, at the same time, were distinguished between one another. Therefore, they are all feasible conditions and were selected to evaluate the response of a local optimization method to this system. Table 4.3 shows the values of the 12 continuous decision variables in each initial estimate set considered.

When trying to utilize those initial estimates in Nelder-Mead method, it was observed that the objective function was oscillating and the method was getting trapped in local peaks near the given initial estimate. Therefore, further analysis was performed on the NLP space of variables and on the influence of simulation recycle tolerance in the optimizer performance. Those topics will be discussed in the following section.

Convergence Tolerance of the Simulation Recycles

Since the main variables of interest here are flowrates, for those, an investigation step was carried out in order to analyze the effect of changes in HYSYS user-specified flowrate recycle tolerance in the optimization results.

The methodology employed was to sweep the whole space of continuous decision variables using a bi-dimensional 20x20 mesh for each pair of continuous decision variables, varying them while keeping all other variables fixed. This resulted in 400

Table 4.3: Initial estimate values of continuous decision variable u_1 to u_{12} for five distinguished sets (I01 to I05).

Decision variable	Initial estimate set				
	I01	I02	I03	I04	I05
u_1	0.51	0.5	0.16	0.16	0.16
u_2	0	0.5	0	0	0.5
u_3	0	0.5	1	0.5	0.75
u_4	0	0.5	0.5	0.025	0.5
u_5	0.50	0.5	0	0.025	0.5
u_6	0	0.5	0	0	0.5
u_7	0.89	0.5	0.17	0.31	0.5
u_8	0.33	0.5	0	0	0.5
u_9	0.57	0.5	0.13	0.26	0.5
u_{10}	0.57	0.5	0.57	1	0.5
u_{11}	0.50	0.5	1	0	0.5
u_{12}	0.50	0.5	0	1	0.5

simulation points per mesh with the values of both variables and the calculated objective function. This procedure was repeated for three levels of user-specified recycle flowrate tolerance: *default*, *0.1*default* and *0.01*default* — this value is utilized to calculate the actual tolerance as described in Section 3.2.5. Figures 4.2 and 4.3 illustrate the results for two chosen pair of variables, u_1 x u_4 and u_{11} x u_{12}.

Figure 4.2 shows more oscillation in the objective function for the same level of recycle tolerance than Figure 4.3 and also a higher total simulation time. For both, there is a clear reduction in the oscillations when tightening the tolerance. u_1 (flowrate from SG-B to SG-A) and u_4 (flowrate from SG-A to NGPU-E) are the most influential variables of the simulation model, therefore by changing them it is expected that the simulation takes longer to run, when compared to changing u_{11} and u_{12}, which are associated to smaller residual gas flows. It is also interesting to notice the high increase in total simulation time from Figures 4.2 and 4.3 (b) to (c) — for u_1 x u_4, the time more than doubles when

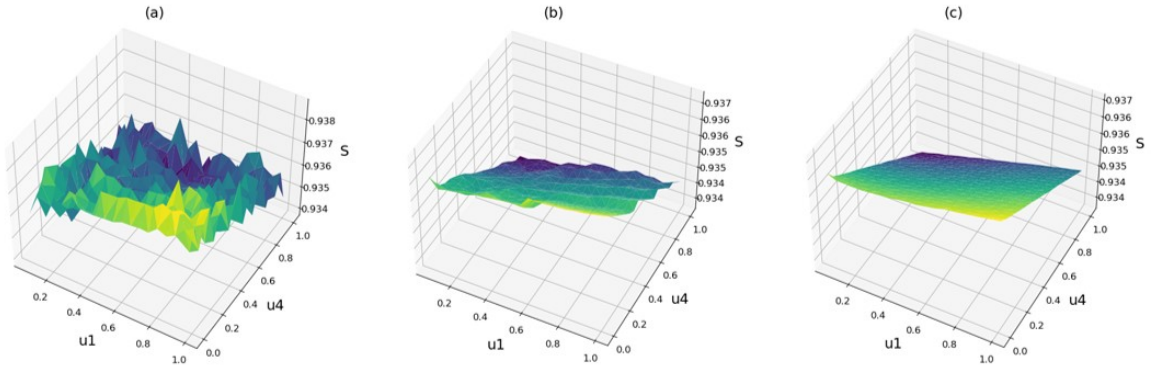

Figure 4.2: Space of variables investigation for three different levels of flowrate recycle tolerance in HYSYS, 400 simulation points. Objective function S presented as a function of u_1 and u_4. (a): default recycle tolerance, 2h 46min; (b): one-tenth of default recycle tolerance, 2h 41min; (c): one-hundredth of default recycle tolerance, 5h 35min.

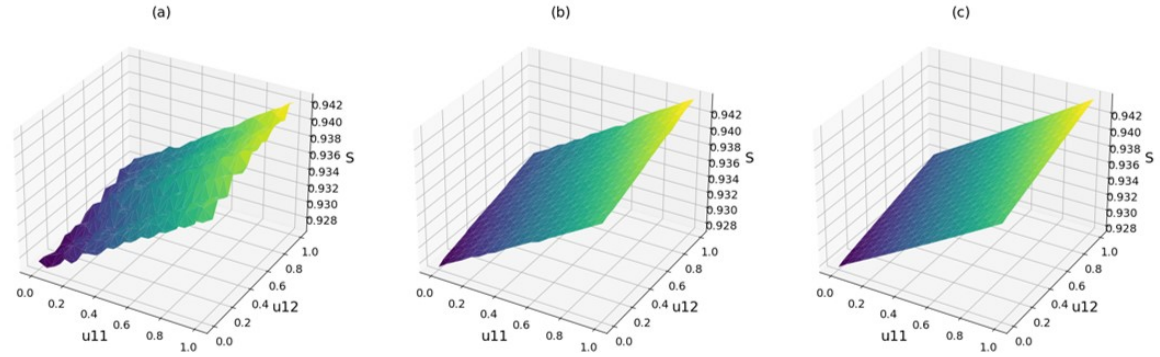

Figure 4.3: Space of variables investigation for three different levels of flowrate recycle tolerance in HYSYS, 400 simulation points. Objective function presented S as a function of u_{11} and u_{12}. (a): default recycle tolerance, 1h 15min; (b) one-tenth of default recycle tolerance, 2h 05min; (c): one-hundredth of default recycle tolerance, 3h 02min.

moving from the intermediate (2h 41min) to the tightest (5h 35min) recycle tolerance.

For both figures, results show that HYSYS default flowrate tolerance is not an appropriate choice for this work's purpose. When using default tolerance, the objective function values fluctuate significantly when the decision variables are changed, with no distinguished pattern (Figures 4.2(a) and 4.3(a)). For tighter tolerances (10 and 100 times smaller), fluctuations decrease until a smooth surface is formed – a shape that is suitable for coupling the model with a local-search optimization algorithm without compromising the performance of the method. When the outer optimizer tolerance is tighter than the inner simulation tolerance, the local-search method gets trapped in the vicinities of the given initial estimate.

However, tightening recycle tolerance has the drawback of increasing simulation execution time due to the higher number of iterations required to converge each simulation.

This can be seen in Figure 4.4, which shows the total optimization time results for each of the five initial estimate sets from Table 4.3, in the three investigated levels of simulation flowrate recycle tolerance. The red dots in the figure are the average optimization times for each level of simulation recycle tolerance, showing the increase in time demand as this parameter gets tighter. This is specially noticed when moving from *0.1*default* to *0.01*default* tolerance, where average optimization time goes up over 170 %, from 2h 58min to 8h 04min.

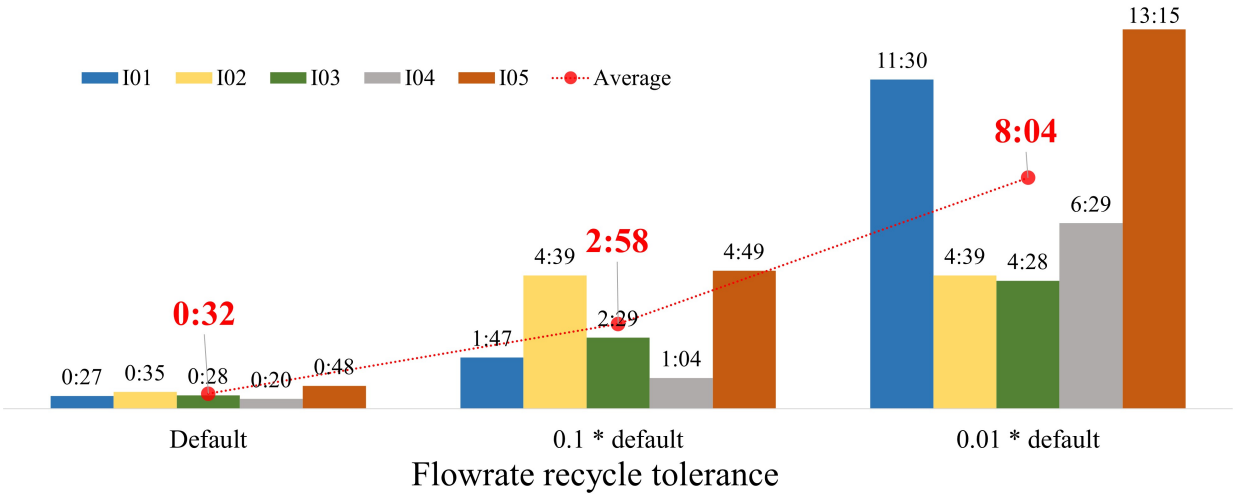

Figure 4.4: Graphical results of the NLP problem. Values of total optimization time for each initial estimate set at three levels of simulation recycle tolerance.

In the light of the study and results presented, from here on a simulation flowrate recycle tolerance of one-tenth of HYSYS default value will be used, considering that this value is a compromise between result accuracy and total optimization time, since the latter is a major parameter to determine optimization engineering applicability.

Nelder-Mead Results

Five executions of the NLP optimization with Nelder-Mead method were performed using the same inlet conditions, raw gas flowrates and compositions. Thus the only difference between those calculations were the initial estimates for the continuous decision variables, where the five distinguished sets from Table 4.3 were used.

Figure 4.5 shows the behavior of the objective function S throughout the optimization run, with each line representing one optimization execution for one given initial estimate. For all five runs, Nelder-Mead method found one optimal solution, however, they are different among each other. For each execution, the method found one optimum point in the vicinity of the given initial estimate, what can be seen in Figure 4.5 by the lines not

crossing one another. Those results therefore indicate that the NLP problem has local minima and is non-convex. Since Nelder-Mead flexible polyhedron is a local optimization algorithm, it does not guarantee global optimality and instead finds different optimum results depending on the starting point.

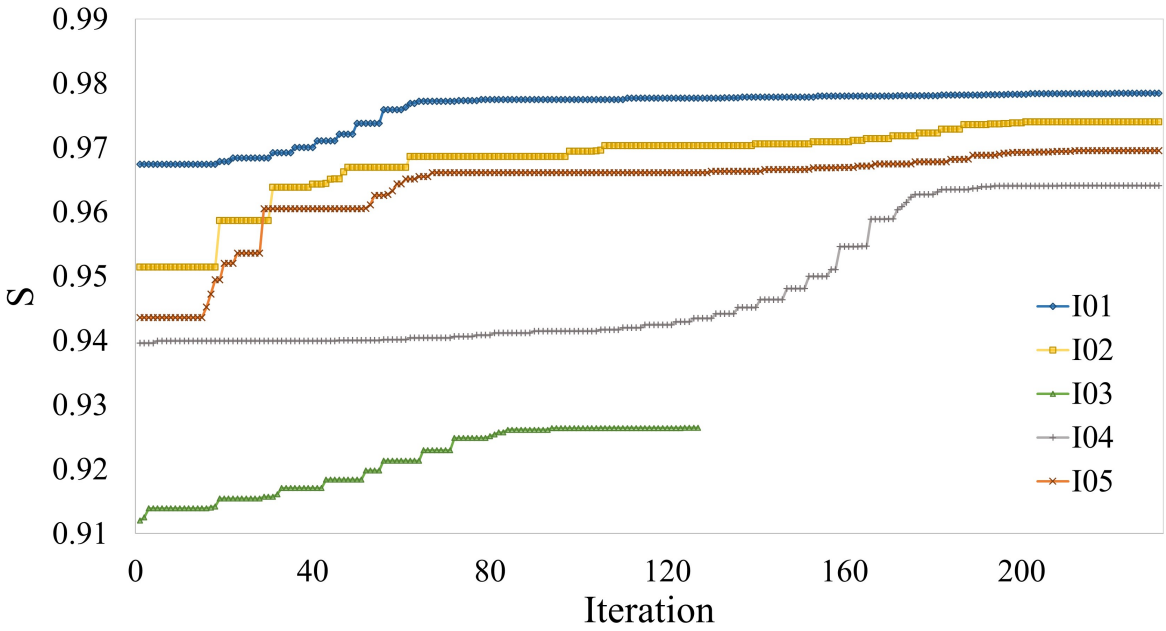

Figure 4.5: Graphical results of the NLP problem with Nelder-Mead method, objective function S per iteration. Each line represents the NLP optimization for one given initial estimate from Table 4.3.

Those results indicate the need to define a strategy to generate adequate initial estimates for Nelder-Mead method. To handle that, a pre-screening step was added to the NLP problem, in which global PSO algorithm is executed and generates an initial estimate for the local method. This strategy aims to find an initial estimate located in the vicinity of global optimum, such that the final optimization result is independent of the user-defined initial estimates. This strategy will be discussed in the next section.

Also, further investigation on the NLP problem indicated that, depending on the given initial estimate, the intermediate simulation recycle flowrate tolerance (*0.1*default*) might not be enough to guarantee optimality. This can be seen in Figure 4.6, which shows the optimum values of the objective function found by Nelder-Mead algorithm for each initial estimate set on the three levels of flowrate tolerance. Results show that for I02, I03 and I05 there was no gain in tightening the tolerance from *0.1*default* to *0.01*default*, since the same optimum was found and execution time increased. On the other hand, for I01 and I04 a higher optimum profit was found when the tightest flowrate tolerance was used.

Since using the tightest recycle tolerance for the whole steps of optimization is

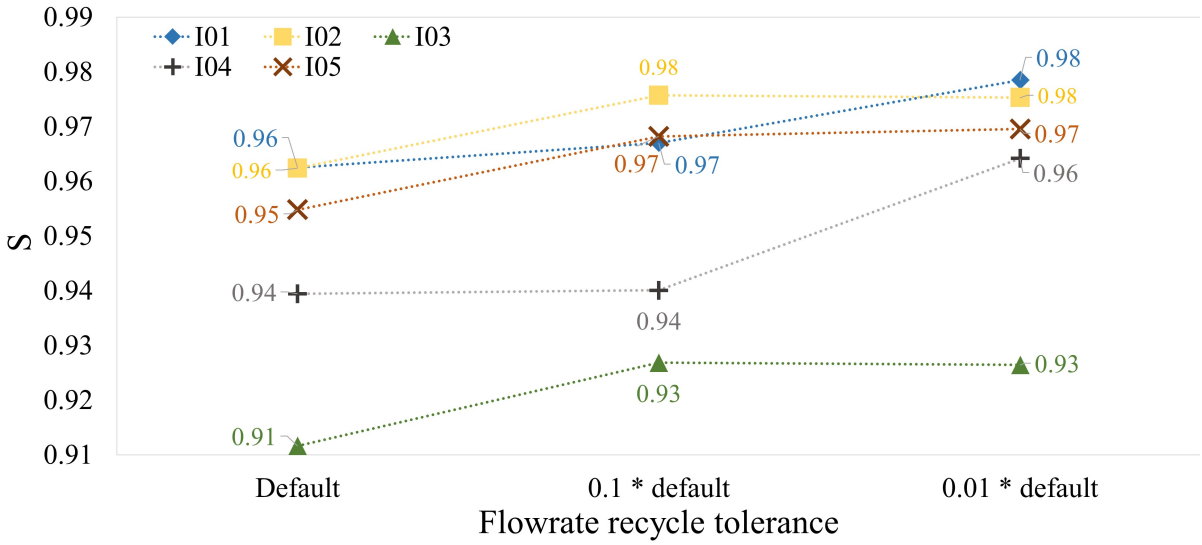

Figure 4.6: Graphical results of the NLP problem for values of the objective function for each initial estimate set for three simulation recycle tolerances.

unfeasible due to execution time, to handle this a refinement step is included after solving the MINLP problem, in which the *0.01*default* tolerance is used to verify if there are further gains in tightening flowrate recycle tolerance.

PSO Pre-Screening

As the evaluation of the NLP problem showed that local minima exist in the space of variables, PSO was used as a pre-screening step in order to guarantee that the NLP optimization search is global and not local in this phase of the study. The strategy implemented was to use the global optimization algorithm as a first execution for the NLP problem to find a region near global optimum and then use those results as input to Nelder-Mead method.

The results from Figure 4.7 show that the algorithm converged to a value of 0.978 for the objective function after 2100 function evaluations, from a starting value of 0.948. 35 of the 2100 evaluations were non-feasible points, therefore Figure 4.7 shows 2065 calculated points. The objective function value found by the PSO was better than the corresponding ones from all the five user-specified initial estimate values (objective function values for iteration zero in Figure 4.5). To carry out this procedure a total of 22 hours was required, an average of 37.7 seconds per rigorous function evaluation.

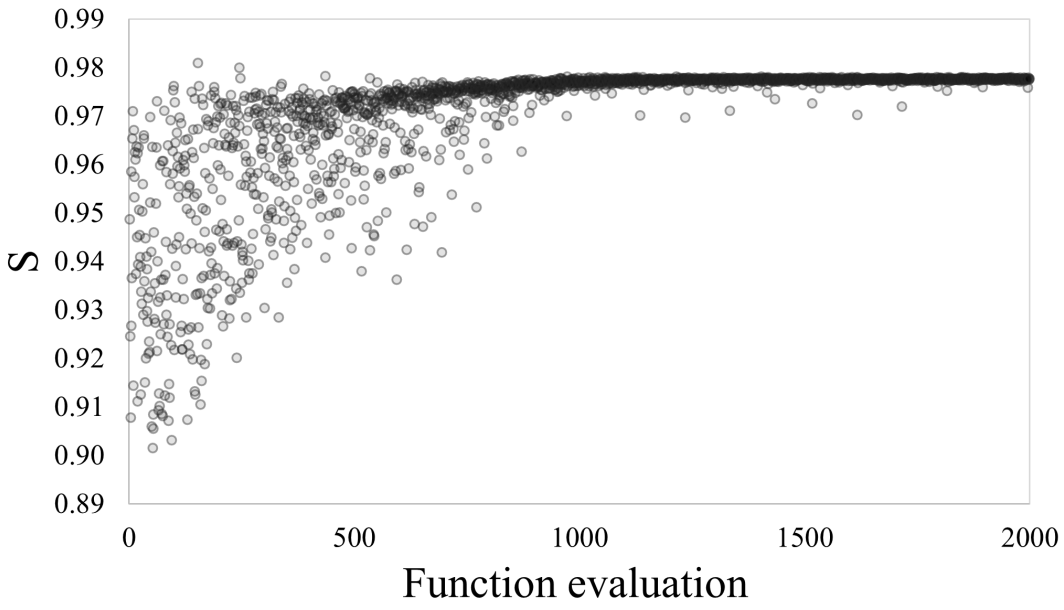

Figure 4.7: Results of PSO pre-screening for the NLP problem. Chart shows the values of the objective function S in each of the 2100 function evaluations during the optimization procedure.

Final NLP Approach: PSO Pre-Screening plus Nelder-Mead Refinement

PSO and Nelder-Mead were successfully coupled in series: PSO identified the vicinity of global optimum and its result (objective function of 0.978) was used as an initial estimate for the Nelder-Mead method. This method refined the PSO result locally, converging to a value of 0.979, as seen in Figure 4.8. By coupling the two methods it was possible to find the best result from all NLP executions, when compared to the results found by Nelder-Mead for the five user-specified initial estimates (Figure 4.5). Those results proved the effectiveness of the proposed methodology. In quantitative terms, a gain of 7.3 % in the variable profit was identified when comparing the objective function value of initial estimate I03 (0.912) to the final value encountered by Nelder-Mead method when coupled with PSO pre-screening (0.979). I03 was defined as the base-case scenario since it is a possible configuration already observed in the industrial plant. This base-case was also the chosen scenario to further detail the mass balance of the natural gas processing site, per NGPU. This data is presented in C and have the intention of providing more information on the process simulation and NGPU performance.

If the PSO method had been used with the objective of finding accurate final results instead of approximate initial estimates, it would be recommended to repeat its execution a few more times in order to create a histogram of the optimum values encountered to verify their consistency, since the method has stochastic characteristics.

Even though PSO pre-screening took 22 hours to be completed, this time is not limiting for the proposed engineering problem. Since the pre-screening finds the vicinity of global optimum, this step would only need to be redone in case of significant disturbances in the plant or when a new plant setup was defined.

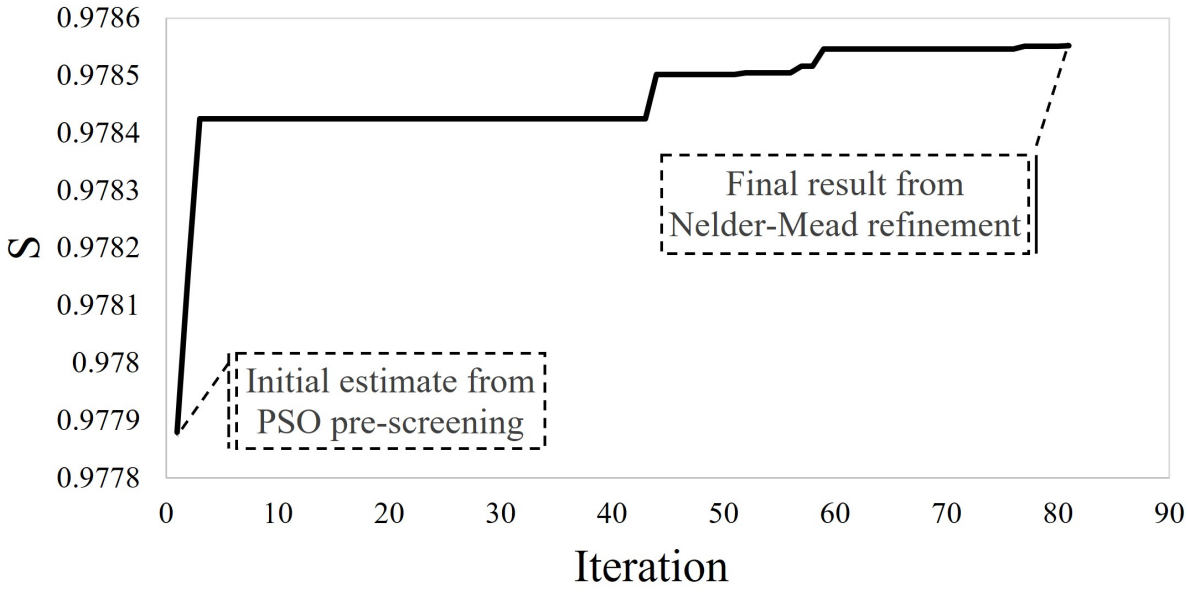

Figure 4.8: Results of PSO pre-screening coupled with Nelder-Mead local refinement for the NLP problem. Chart shows the values of the objective function S in each iteration of the optimization procedure.

4.3.2 PSO: MINLP Analysis of the Variables Search Space

Besides its usage for pre-screening, the PSO method was used to explore, analyze and better understand the space of variables (u and y). This step was crucial for the further development of the work, due to the size and complexity of the problem. It was this analysis that indicated the general behavior of the variables and that gave insight to the next steps of the problem solution.

In this analysis, several PSO executions were performed separately. The results were gathered and filtered in order to select only the results that do not violate any constraints, hereafter called *feasible points*. There was a total of 6592 feasible points that were plotted in 2-dimensional scatter plots, each showing a pair of decision variables colored by the corresponding value of the objective function. Those results are shown in Figures 4.9, 4.10,4.12 and 4.13. Each of those figures present eleven charts, always showing the behavior of one chosen continuous decision variable u in the x-axis with the variation of the other eleven variables (y-axis). The colors indicate the value of the objective function

for each point — an agglomeration of dark-red points indicate a local maximum region.

For a more detailed analysis of the results obtained, it is important to keep in mind that the product sales price established in Table 3.8 represent a scenario in which liquid products are more valuable than gas streams. Therefore, it is expected that this statement is translated by the optimization model and, consequently, observed in the study of the space of variables. For instance, variables such as u_2 (raw gas flowrate for external processing) and u_3 (raw gas bypass flowrate) lead to a proportional increase in gas products and should be disfavored.

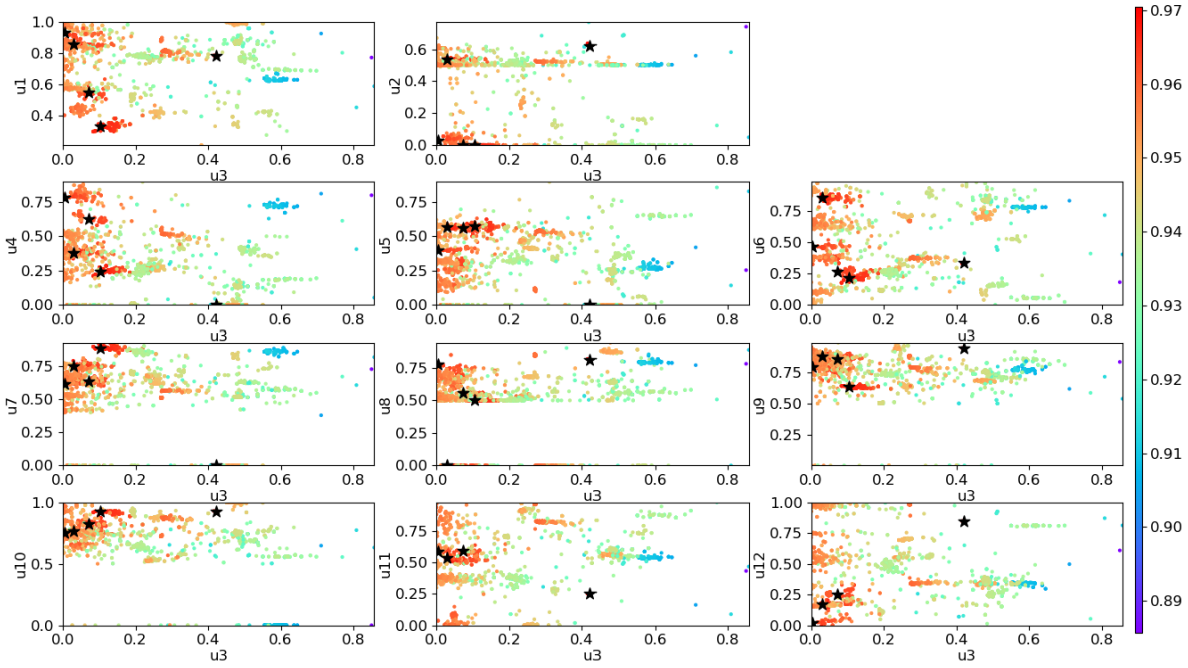

Figure 4.9: Eleven 2D plots of u_3 (raw gas bypass flowrate from the outlet of SG-A to sales gas) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions.

Figure 4.9 shows the behavior of u_3 against the other eleven continuous variables. u_3 represents the flowrate of a raw gas bypass that mixes non-processed gas from the outlet of SG-A directly into the sales gas stream, without any liquid recovery. As such, in a scenario of low natural gas sales price, it is expected that this bypass is disfavored and that the optimum region is located at low values of u_3. This is corroborated in all charts of Figure 4.9, where, in general, there is a continuous increase in the value of the objective function from the right to the left-side of the plots, in which the optimum region is located. These results then indicate that the optimum tends to be located at $u_3 = 0$. However, by observing the star-shaped symbols locations it is possible to see

high objective function values outside the concentration of dark-red point, therefore corroborating the non-convex characteristic of the problem.

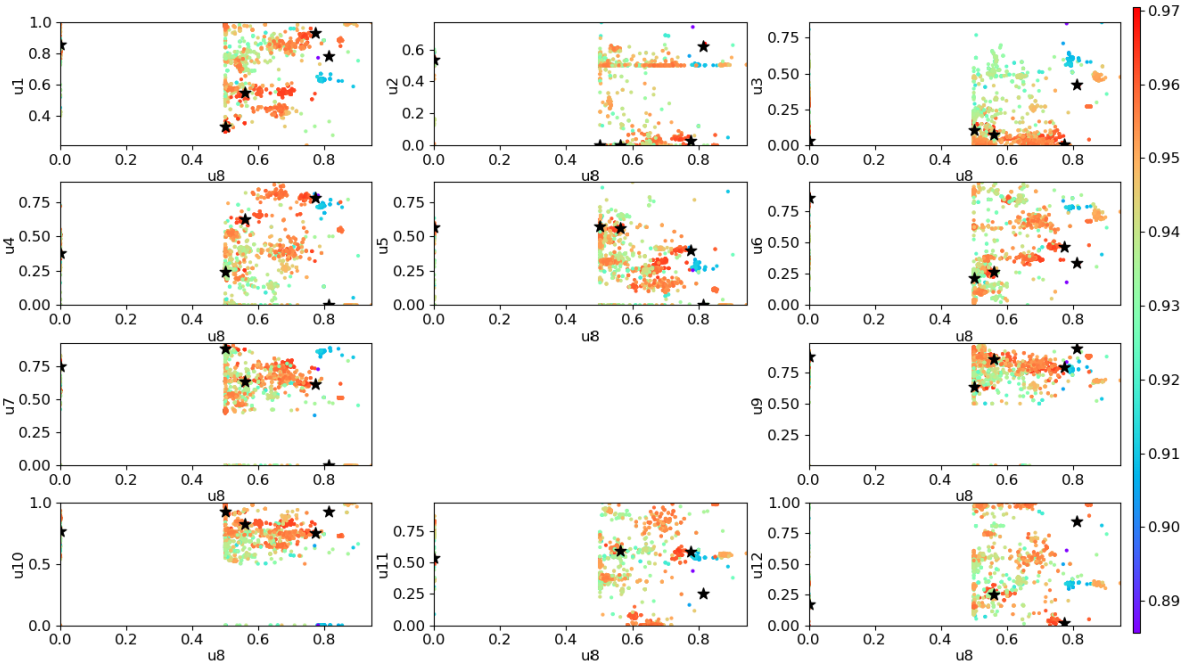

Figure 4.10: Eleven 2D plots of u_8 (NGPU-A feed gas flowrate) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions.

The behavior of NGPU-A feed flowrate (u_8) against the other eleven continuous variables is shown in Figure 4.10. First, it can be seen that a wall-shaped region is formed on $u_8 = 0.5$, which corresponds to the lowest value of u_8 if NGPU-A is under operation ($y_4 = 1$). Moreover, some of the best points found are in this regions, what indicates a tendency to increase profit by reducing NGPU-A feed flowrate. This would correspond to low values of u_8. However, this variable is limited by its lower bound, 0.5. To go under that, NGPU-A would need to be put in standby ($y_4 = 0$).

On the contrary of Figure 4.9, the plots from Figure 4.10 show a discontinuity between $0 \leq u_8 \leq 0.5$. This comes from the lower bound of u_8, which is 50 % of the upper bound, meaning that NGPU-A cannot operate with a feed rate lower than 50 % of its maximum flow. Under theses conditions, the process unit is put in standby. As a consequence of this discontinuity, the optimization method has difficulty in finding a feasible region with $u_8 = 0$ — this was found to be a general behavior for all variables that show discontinuity.

However, the region with NGPU-A in standby exists and was previously known to be feasible due to operational knowledge of the industrial plant. Therefore, the region with

$u_8 = 0$ was encountered by setting NGPU-A in standby ($y_4 = 0$) on the initial generation of PSO particles. This strategy aimed to show PSO that those regions exist and then let the method walk freely from there, exploring this area. As a result, the scatter points on the left side of the plots from Figure 4.10 were found. They are in lesser quantity than the region that considers NGPU-A under operation due to the fact that the method needed guidance based on plant knowledge to find them. However, once found, the results prove that this region is not only feasible, but present a concentrated area of economically interesting points.

To further analyze the behavior of u_3 and u_8, a 3-dimensional plot of those variables against the objective function was built and is shown in Figure 4.11. This plot gives a clearer view of the wall-shaped region at $u_8 = 0.5$, in which there is a high concentration of points, including one of the five highest optima found. It is evident that the most explored area is near this region, indicating that the method walks towards reducing NGPU-A feed flowrate up to its lower bound. Another region of interest lays at $u_8 = 0$: even though there are less points, they are enough to notice a pathway and to verify that high values of the objective function might be found when NGPU-A is in standby. At this optimum point, it is also noticeable that u_3 is near zero.

The star-shaped icons in Figure 4.10 show the best points found and one of them is in $u_8 = 0$. This indicates that it is possible to leave this NGPU in standby and redistribute its feed gas to the other NGPUs. The high values of the objective function in this region indicate that this scenario in economically interesting due to the reduction of overall operating expenses when a NGPU is out of operation. In physical terms, NGPU-A is the simplest process unit and the one with the lowest liquid recovery fraction — as a process unit that produces proportionally more gas streams than the other, it is indeed expected to be disfavored in this scenario of low gas sales price.

Similarly, Figures 4.12 and 4.13 plot u_7 (main feed flowrate for NGPU-E) and u_{10} (total feed flowrate of NGPU-D), respectively, against the other eleven continuous decision variables and also show a region of discontinuity. However, the wall-shaped region at the lower bound is not observed in either figures, differently from Figure 4.10. By guiding PSO initial generation, it was found the region to the left of the plots, in which $u_7 = 0$ and $u_{10} = 0$, respectively. Even though there are few calculated points in this region, for the same reasons discussed regarding Figure 4.10, the star-shaped symbols assist the reader. In terms of optimization, the region with $u_7 = 0$ shows higher values for the objective function than for $u_{10} = 0$, which is in accordance to the behavior of the plots, discussed as following.

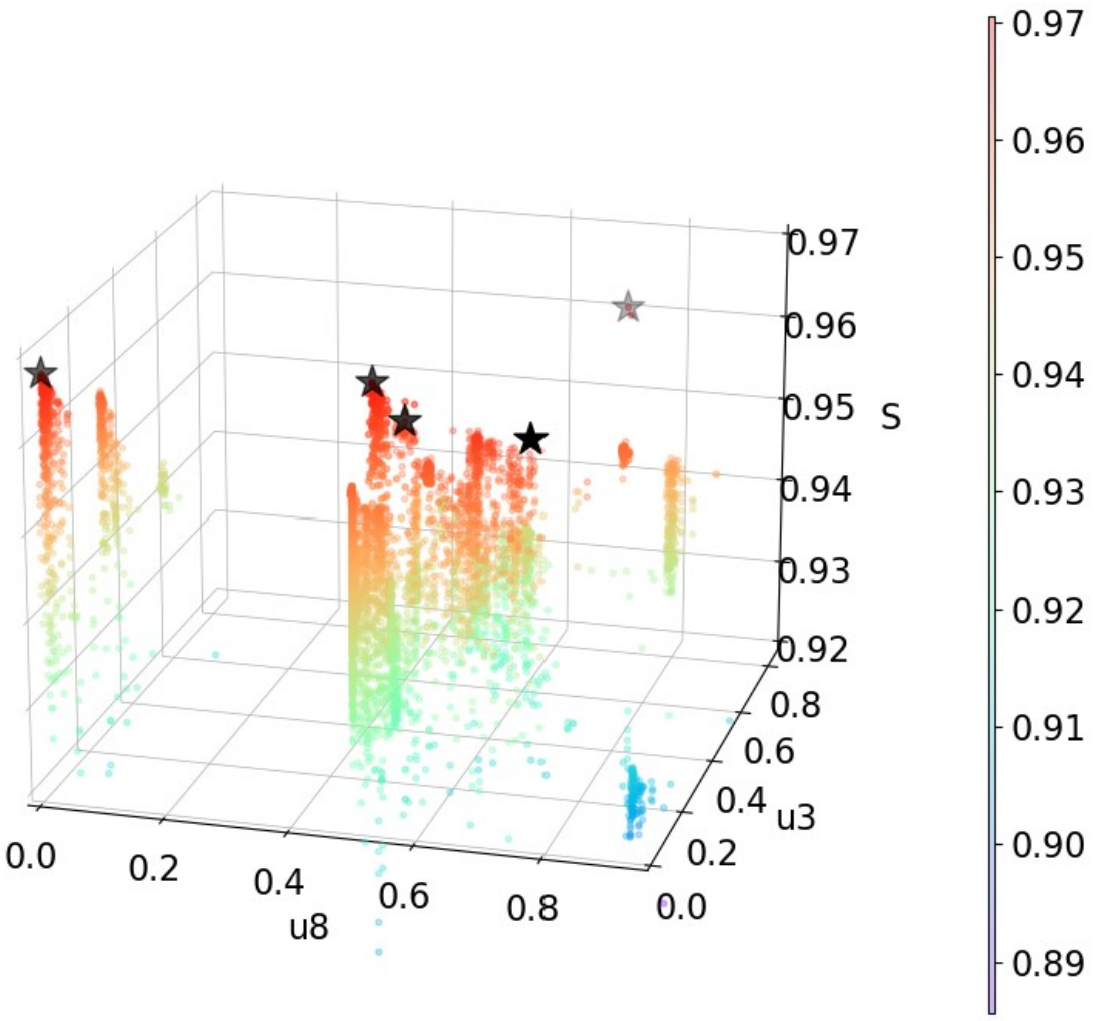

Figure 4.11: 3D plot of u_3 and u_8 against the objective function S. The star symbols represent the five highest objective function values encountered in distinguished PSO executions.

In Figure 4.12, it is interesting to notice that there are several distinguished regions of local maxima, each indicated by a concentration of dark-red points. This illustrates well the complexity of the problem and corroborates the importance of incorporating in the methodology a global pre-screening step to walk through theses local peaks. No general tendency is observed in the plots and the star-shaped icons show that local optima were found both for low and high values of u_7.

On the other hand, Figure 4.13 shows that there is a tendency for the optimum to be located at high values of u_{10}, i.e., NGPU-D operating at high feed rates. This is a consistent result: as described in Table 3.3, NGPUs-B/C/D use a combination of propane refrigeration and turbo-expansion to cool down the natural gas, which results in the lowest temperatures from all NGPUs and, by consequence, the highest liquid recovery. In this scenario of high sales price for liquid products, process units such as NGPU-D are then

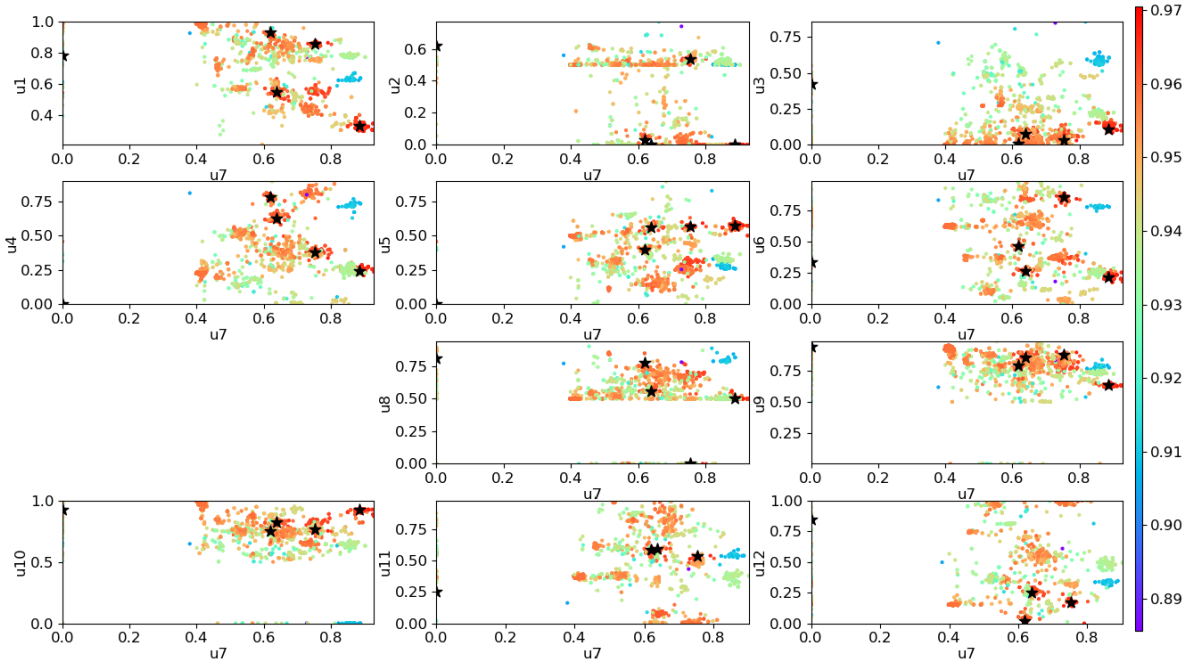

Figure 4.12: Eleven 2D plots of u_7 (NGPU-E main feed gas flowrate) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions.

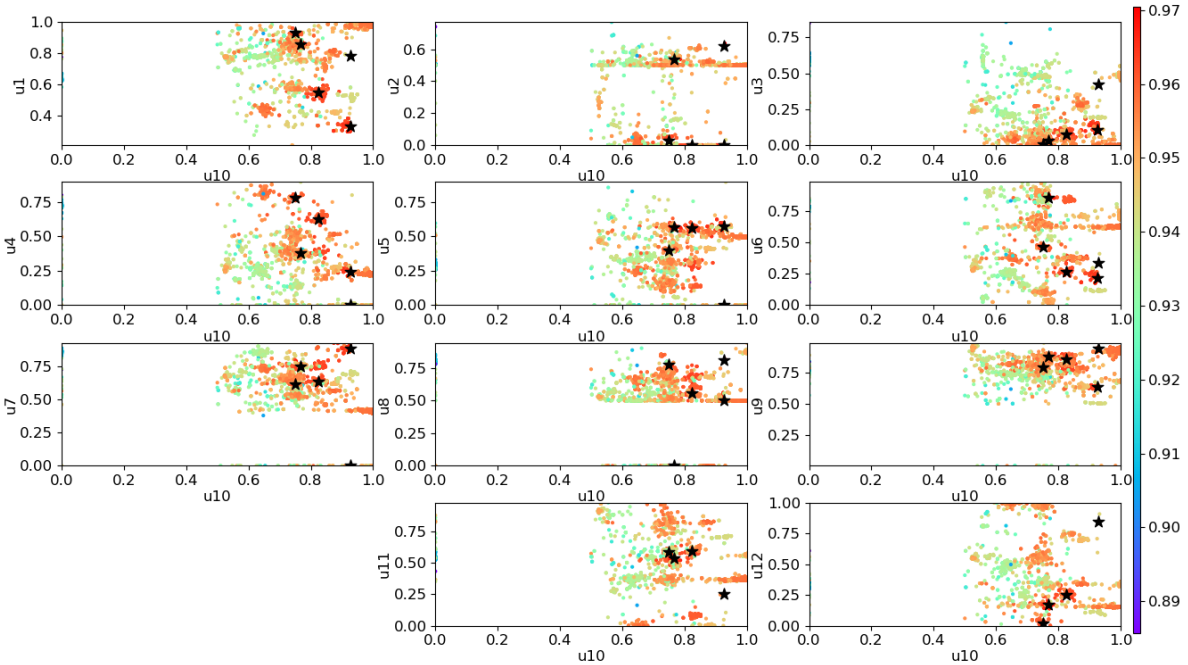

Figure 4.13: Eleven 2D plots of u_{10} (NGPU-D feed gas flowrate) against the other eleven continuous variables, in sequence (u_1 to u_{12}). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions.

expected to be favored.

In summary, the plots discussed in this section show several localized concentration of feasible points, indicating the presence of a number of local maxima. Once more, this corroborates the complexity of the problem and poses the challenge in establishing a methodology to find global optimum.

Furthermore, the discontinuity observed in most of the so called continuous variables is an additional challenge for the optimization method. This behavior is present due to physical and design plant bounds that define a minimum feed gas flowrate for each NGPU. It was found out that this discontinuity interferes with the method's search for the optimal point. Even though PSO is a global method from a probability perspective, it has limitations, such as any other method, and a way to overcome this was found by plant user experience guidance to discover certain feasible regions of variables search space. It is important to mention that including this artificial intelligence in the PSO method was important to ensure that this work's objective was achieved.

By incorporating these slight modifications (i.e., distinct initial particle generations guided by plant experience), PSO method was found to be walking towards the expected path. While in Figure 4.10 the wall-shaped area at $u_8 = 0.5$ might leave unclear if the method is able to proceed to the left of this area, Figure 4.14 certifies that, by plotting the binary variable that represent the operating status of this same process unit, NGPU-A. In this chart, the binary variables corresponding to the operating status of raw gas bypass (y_2) and NGPU-A (y_4) are plotted against each other and the colors indicate the value of the objective function for each scatter point, according to the presented color-bar. On the contrary of Figure 4.10, the plot of the binary variables has no discontinuity. There are feasible points in the region between $0 \leq y_4 \leq 0.5$, indicating that the method is able to explore this region and proceed to putting NGPU-A in standby ($y_4 = 0$).

4.3.3 The MINLP Problem

After solving the NLP problem and analyzing the space of variables with graphic results from PSO executions, the seven integer decision variables (y_1 to y_7) were inserted and the MINLP problem was implemented in Python. In order to facilitate the discussion addressed in this section, Figure 4.15 shows a simplified branch-and-bound algorithm including the main steps implemented and evaluated in this study. Furthermore, the list of parameters used for each of the phases executed is shown in Figure 4.16. For PSO, a number of 20 iterations was fixed and the number of particles was set to 10 times

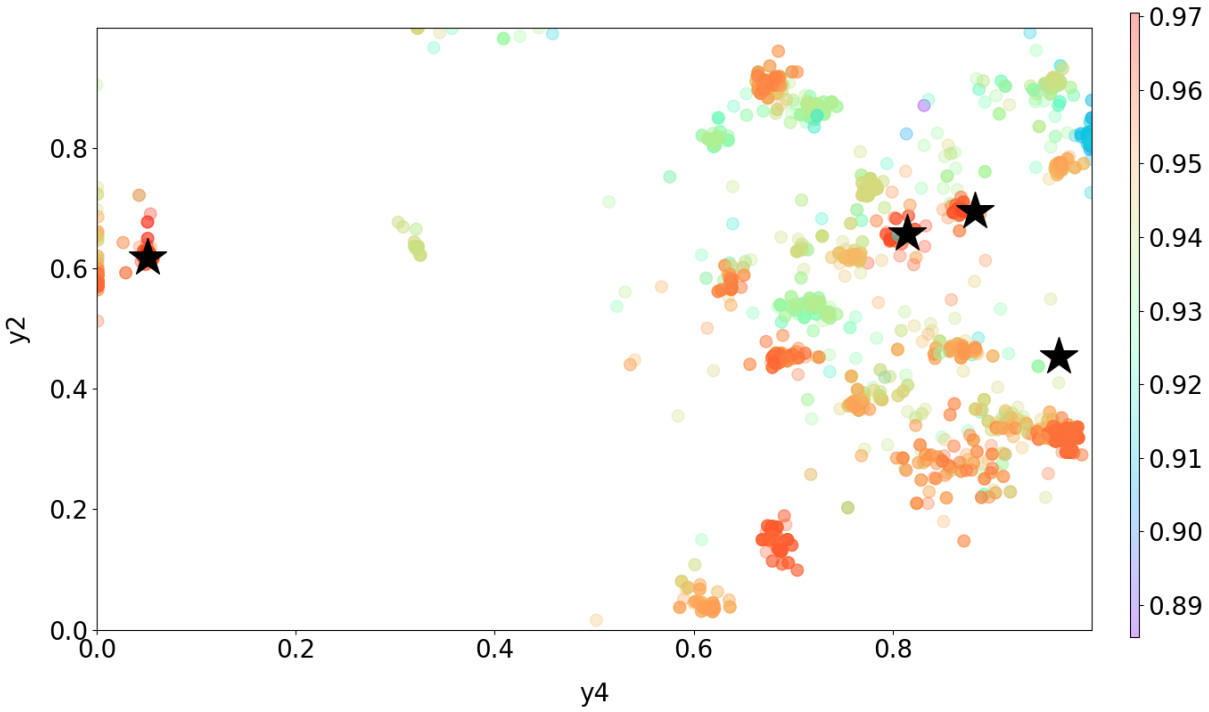

Figure 4.14: 2D plot of y_2 (raw gas bypass operating status) against y_4 (NGPU-A operating status). The colors refer to the value of the objective function for each point, according to the color-bar in the right. The star symbols represent the five highest objective function values encountered in distinguished PSO executions.

the number of decision variables. In the proposed methodology, Nelder-Mead flexible polyhedron used different levels of tolerance depending on where was located: for the MINLP step, a looser tolerance was allowed and then, for post-refinement, the tolerance was tightened in order to obtain the final results.

The schematic from Figure 4.16 couples, for the MINLP problem, a PSO pre-screening, NLP local optimization with Nelder-Mead flexible polyhedron and MIP tree-search via branch-and-bound. The execution time for each phase is presented in Table 4.4, accounting for a total of 23 hours and 22 minutes to solve the MINLP problem, including PSO pre-screening. Those times are associated to the tolerances set for the methods, according to Figure 4.16.

The results show that the MINLP problem was solved successfully and that the total time, less than 24 hours, was suitable for the objective of this work considering its dimension and complexity. Table 4.5 details the results of each NLP execution during tree-search, with information on the values of binary variables, objective function value, number of iterations and function evaluations. The MINLP algorithm executed the NLP problem via Nelder-Mead method 13 times, each corresponding to a node in the tree-search. After the twelfth execution of the NLP, the result found was within the

1. Define the initial estimate of the MINLP problem from **PSO relaxed-integrality pre-screening**.
2. Solve a relaxed-integrality NLP. If the solution is integer, then MINLP solution found and **END**. Otherwise, y is rounded and a new NLP problem is solved with fixed y.
3. If the solution of (2) is feasible, then store it for latter comparison.
   Set upper limit, $z_{upper} = \infty$.
4. Start **tree-search.** Initialize the list of NLP subproblems with the solution of the relaxed-integrality NLP from step 2.
5. If the list is empty, then the best current result is the MINLP solution. **END**. Otherwise, branch tree, create two new child subproblems from the least integer *y* and add them to the list of candidates.
6. Select current candidate from the list.
7. Solve the relaxed-integrality formulation of the current problem (RP). The initial estimate is generated from the solution of the corresponding LP problem. The RP solution z_{RP} is obtained.
8. Apply the investigation criteria:

   8.1 If RP is unfeasible, then the current candidate has no feasible solution.

   8.2 If $z_{RP} \geq z_{upper}$, then the current candidate has no feasible solution better than current best.

   8.3 If the optimal solution of RP is feasible, integer and $z_{RP} < z_{upper}$, then $z_{upper} \leftarrow z_{RP}$. Go to 10.
9. If there is another subproblem from the same parent, select it as the current problem and go to 7. Otherwise, go to 5.
10. Eliminate the NLP subproblems that have $z_{RP} \geq z_{upper}$. Go to 5.

Figure 4.15: Simplified branch-and-bound algorithm implemented for the MINLP problem.

Table 4.4: Execution time for each phase of the MINLP problem.

MINLP phase	Time per execution	Number of executions	Objective function value
PSO: pre-screening	8h 02min	1	0.9610
Nelder-Mead: relaxed-integrality	14min 49s	1	0.9629
Nelder-Mead: rounded integer variables	38min 54s	1	0.9689
Partially-integer Nelder-Mead	1h 06min 41s	13	0.9852
Total MINLP execution	23h 22min		

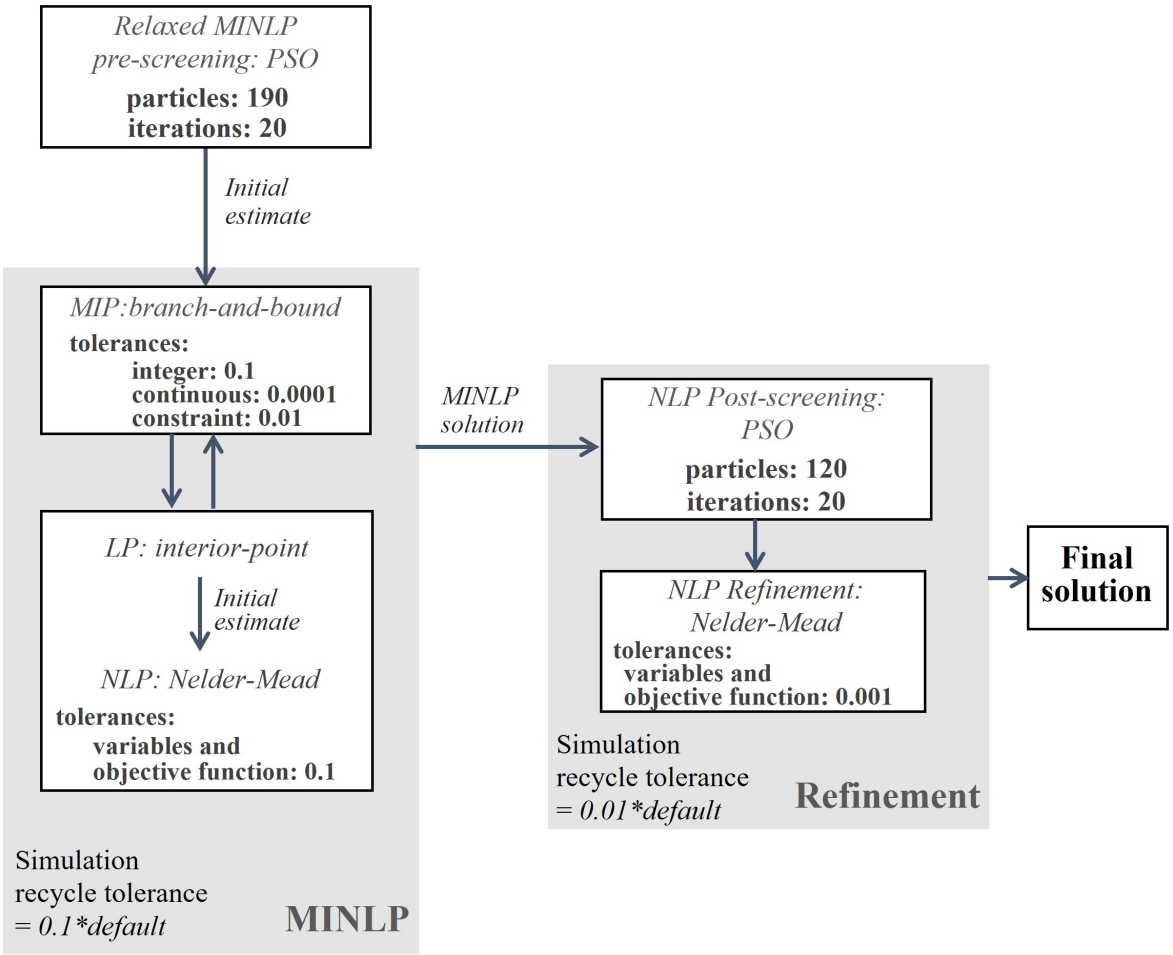

Figure 4.16: Schematic diagram of the MINLP calculation phases, listing the values of the main parameters utilized for every method in each phase of problem solution.

established tolerance for the integer variables (0.1) and, therefore, it was followed by a strictly-integer NLP execution, in which the values of the binary variables from the previous execution were rounded. This is run number 13 from Table 4.5 — the result was feasible and was then taken as the MINLP solution, corresponding to a objective function of 0.9852, the best value found so far.

The MINLP algorithm found the solution presented in Table 4.6. Looking at the binary variables, the optimizer found $y_1 = y_4 = 0$. In physical terms, this means that there is no raw gas being sent for external processing (y_1) and that NGPU-A is in standby (y_4), while all other process units are under operation. Furthermore, the continuous variables show that the optimum considered no raw gas bypass flowrate (u_3). Those results are in total agreement with the behaviors thoroughly discussed in Section 4.3.2. In summary, the three process units mentioned are the ones, among all, that favor sales gas production the most. Since the market price scenario is of increased liquid products price, raw gas for external processing, NGPU-A and raw gas bypass are expected to be

Table 4.5: Tree-search results of the MINLP problem solution. Table shows the number of executions, respective tree node, values of the binary decision variables (y_1 to y_7, number of iterations and function evaluation of each NLP Nelder-Mead run.)

Number	Tree node	y_1	y_2	y_3	y_4	y_5	y_6	y_7	S	Iterations	Fun. Eval.
1	1	1	0.5088	0.9704	0.6276	0.9548	0.9018	0.9650	0.9550	202	342
2	1	0	0.3950	0.9962	0.9973	0.8914	0.9964	0.8265	0.9686	209	337
3	2	0	1	0.9285	0.5891	0.9679	0.9302	0.9671	0.9729	217	361
4	2	0	0	0.9001	0.5260	0.9090	0.9690	0.9413	0.9721	100	198
5	3	0	1	0.8213	0	0.9983	0.9860	0.9211	0.9673	158	257
6	3	0	1	0.9678	1	0.9344	0.9741	0.8327	0.9725	130	251
7	4	0	1	0.9716	1	0.9928	0.9933	0	0.9337	187	301
8	4	0	1	0.6924	1	0.8855	0.9452	1	0.9731	119	212
9	4	0	1	0	0	0.98	0.98	0.98	0	4	67
10	4	0	1	1	0	0.8292	0.9953	0.9995	0.9849	108	186
11	5	0	1	1	0	0	0.85	0.85	0	4	63
12	5	0	1	1	0	1	0.9732	0.9720	0.9827	175	308
13	Final	0	1	1	0	1	1	1	0.9852	19	62

disfavored in comparison to the other NGPUs.

It is interesting to notice, from these results, that the optimizer was able to put three process units in standby simultaneously, by manipulating both continuous and integer variables. Also, the other variables were changed such as all the raw gas input flowrate was able to be received by the process units under operation, therefore maximizing their utilization and reducing OpEx. The optimum operating scenario showed a high utilization of all NGPUs that were operating, indicating that no other process units could be put in standby. Also, raw gas was preferably sent to NGPUs-B/C/D, since those are the NGPUs with the highest liquid recovery fraction. Lastly, the result of u_{11} indicates that in the optimum scenario 53.79 % of the residual gas from NGPU-E returns for reprocessing. This is an interesting and not intuitive result, since it shows that there is a compromise between two effects: 1) revenue increase due to higher liquid recovery when u_{11} is high and 2) OpEx reduction due to lower NGPU utilization when u_{11} decreases. As a result, the optimum value of u_{11} is not near any of its bounds. u_{12}, on the other hand, has no physical meaning in this scenario, since it represents the fraction of residual gas flowrate for reprocessing from NGPU-A, which is in standby.

There are a few more interesting considerations to be made and discussed on the results and the methods utilized to achieve them. Those will be detailed in the following subsections.

Table 4.6: Optimization results obtained by the MINLP, PSO post-screening and Nelder-Mead refinement steps. Values of continuous and binary variables.

Type	Variable	MINLP algorithm	PSO post-screening	Nelder-Mead refinement
Continuous	u_1	0.3177	0.5805	0.7630
	u_2	0.6556	0.0963	0.0971
	u_3	0.0008	0	0
	u_4	0.4837	0	0
	u_5	0.9670	0.8139	0.6946
	u_6	0.8809	0.5310	0.5849
	u_7	0.4996	0.7330	0.5734
	u_8	0.6907	0.0140	0.0103
	u_9	0.9654	1	0.9898
	u_{10}	0.8602	0.7954	0.9468
	u_{11}	0.5379	1	1
	u_{12}	0.2390	0.8263	0.1801
Binary	y_1	0	0	0
	y_2	1	1	1
	y_3	1	1	1
	y_4	0	0	0
	y_5	1	1	1
	y_6	1	1	1
	y_7	1	1	1
Execution time		23h 22min	9h 17min	1h 22min

PSO: Pre-Screening with Relaxed Integrality

In this approach, PSO pre-screening was only executed once and accounts for over 34 % of total MINLP execution time, the largest time demand for a single step. These numbers confirm that it would not be feasible to consider using PSO inside every tree-node of the MINLP code. Even if this modification was possible, there would be no guarantee that better results would be achieved, since it was shown that the method needed guidance to navigate through discontinuities. Therefore, considering the proposed methodology, PSO pre-screening was successfully utilized as an attempt to find the vicinity of an optimal region.

It is important to notice that, in order to achieve the presented results, it was crucial to modify the default method code, as explained in Section 3.4.1. Only when the use of the decision variables was partitioned (first the binary and then the continuous) the method was able to find feasible points and to explore the space of variables.

The strategy implemented was to use a global optimization algorithm as a first run to determine initial estimates for the MINLP problem with relaxed integer variables. In this approach, PSO results are used to indicate a region near global optimum. Then, Nelder-Mead and branch-and-bound algorithms are employed in order to refine PSO initial results and determine actual global optimum, corresponding to maximum profit.

Branch-and-Bound: MIP Optimization

During the MINLP executions, it was noticed that the large number of local maxima was interfering in the default implementation of the branch-and-bound. An example of that can be seen in the results of Table 4.5, when comparing runs number 12 and 13. The latter is the NLP execution using the results of the former as initial estimate, but rounding the values of the binary decision variables. If this were a convex problem or at least it were in a region of convexity, the result of execution 13 should be worse than of 12. However, the numbers show the contrary: when rounding the binary variables from run 12 (objective function value of 0.9827), an even higher value was achieved — 0.9852 for run 13. This result is very interesting, as it illustrates the MINLP method encountering the nonconvexities and local peaks of the space of variables.

For this reason, this work adapted the branch-and-bound method. The default implementation utilizes the feasible results of the rounded execution of the relaxed-integrality Nelder-Mead as the initial lower limit for the objective function. In this case, if the first execution inside the search tree yielded lower values than this pre-defined cut, the tree would be pruned and this first result would have been mistakenly defined as the MINLP solution. This happened when attempting to use the default code and would have happened in the MINLP execution presented here (Tables 4.4 and 4.5), where the objective value from the rounded execution of the relaxed-integrality Nelder-Mead (0.9629, from Table 4.4) is higher than the results of the two executions from the first node (0.9550 and 0.9686, Table 4.5). To handle that, the code was modified such that the value of 0.9629 was found and stored, however used for comparison only after the MINLP execution.

LP Interior-Point: Generating Initial Estimates

The LP problem was solved every time Nelder-Mead was called, providing a feasible initial estimate for the NLP method. This strategy was crucial to ensure automation of the process and improved significantly the performance of Nelder-Mead method. Before this modification, most of the NLP executions were yielding $S = 0$ and finishing without any HYSYS call, as result of an unfeasible initial polyhedron.

Nelder-Mead: Local NLP Optimization

Nelder-Mead flexible polyhedron successfully converged for all NLP executions in which at least one feasible point was found. For runs number 9 and 11 (Table 4.5), the value of $S = 0$ indicates that no feasible point was found and no HYSYS calls were made, because for all attempts there was one or more variables outside their constraint limits. This is expected, since those executions correspond to attempts of putting at least three process units in standby, which is an extreme condition for the plant and in practical terms might not be feasible if the raw gas income is superior to the total capacity of the operating units.

Final Results: MINLP + Post-Refinement

PSO post-screening and Nelder-Mead refinement were successfully implemented as the last steps of problem solution. By using tighter tolerances both in the simulation and in the optimization method, as indicated in Figure 4.16, this step refined the approximate results from the previous step. The objective function reached a final value of 0.9948, higher than the 0.9852 found by the MINLP algorithm.

Besides, analyzing the results obtained in the MINLP, before and after the post-refinement, it can be observed the importance of this step. For instance, the final results of u_{10} (NGPU-D feed flowrate) and u_{11} (NGPU-E residual gas reprocessing fraction) direct the plant operation towards liquid product maximization.

4.3.4 NLP and MINLP Results: Profit Gain

Both NLP and MINLP solutions shows potential financial gain in terms of variable industrial plant profit. Table 4.7 shows the quantitative results: objective function and potential profit gain estimated in comparison to a base case, initial estimate I03 from Table 4.3, presented in both percentage and actual plant potential profit gain in R$ per

thousand cubic meters of raw gas inlet.

The results show that, for the NLP problem, the usage of PSO and Nelder-Mead in series resulted in better performance, with a potential gain of 7.3 %. However, by adding the operating status of the process units as integer decision variables, even better results were encountered — result from the flexibility of being able to put one or more process units in standby, reducing OpEx. Coupling the MINLP solution with a post-refinement showed the best overall results for the system in study, bringing a potential gain of over 9 % in the variable plant profit.

Table 4.7: Potential profit gain resulting from the solution of NLP and MINLP problem formulations in comparison to a base case, defined as initial estimate I03 from Table 4.3.

Problem formulation	Objective function	Potential profit gain (%)	Potential profit gain (R$/1000 m^3)
NLP (Nelder-Mead)	0.9704	6.4 %	83.73
NLP (PSO + Nelder-Mead)	0.9790	7.3 %	96.11
MINLP	0.9852	8.0 %	104.94
MINLP + post-refinement	0.9948	9.1 %	118.78
Base case (I03)	0.9120	–	–

4.4 Case Study: Changing Product Price Values

To wrap up this work, a case study was performed to analyze the effect of product prices in the optimization results. This analysis was motivated by the recent worldwide increase in sales gas revenue price due to growth in demand and constrained supply (more details in DUBREUIL (2022)).

To accomplish that, the sales price of gas streams (sales gas and raw gas for external processing) was multiplied by four, whereas the prices of liquid products were unchanged from Table 3.8. This is a simulated scenario that represents a circumstance in which gas streams are more valuable than liquid products.

The MINLP problem was then solved in order to evaluate the difference in the optimizer search path between base case and this case study. The final results from MINLP, PSO NLP post-screening and Nelder-Mead refinement phases are presented in

Table 4.8 and can be analyzed by comparison to Table 4.5. By comparing both tables it is noticeable that the optimizer moved towards opposite directions, as expected.

Table 4.8: Case study optimization results obtained by the MINLP, PSO post-screening and Nelder-Mead refinement phase. Values of continuous and binary variables considering a scenario of gas streams sales price four times higher than base case.

Type	Variable	MINLP algorithm	PSO post-screening	Nelder-Mead refinement
Continuous	u_1	0.7071	0.6818	0.6815
	u_2	0.9870	1	1.0012
	u_3	0.9988	1	0.9984
	u_4	0.2596	0.1250	0.1400
	u_5	0.6786	0.7217	0.8070
	u_6	0.6167	0.1282	0.1285
	u_7	0.8846	0.8958	0.8967
	u_8	0.9970	0.9989	0.9968
	u_9	0.8802	0.0074	0.0074
	u_{10}	0.1604	0.0458	0.0459
	u_{11}	0.0109	0	0
	u_{12}	0.0203	0.0005	0.0006
Binary	y_1	1	1	1
	y_2	1	1	1
	y_3	1	1	1
	y_4	1	1	1
	y_5	0	0	0
	y_6	1	1	1
	y_7	0	0	0

In this case study results (Table 4.8), process units such as raw gas bypass (u_3), NGPU-A (u_8) and raw gas for external processing (u_2) were prioritized due to their high sales gas production. On the other hand, NGPUs-B/D (y_5/u_9 and y_7/u_{10}) were put in standby, since they have the highest liquid recovery fractions from all process units, as previously discussed (Section 4.3.2), which in this case study would yield a lower profit. Another interesting results are observed looking at u_{11} and u_{12}: in this scenario (high sales gas price), both variables were set to zero, meaning that all that residual gas from NGPU-E and NGPU-A is being directly mixed with the sales gas stream.

It is also interesting to notice that in this analysis there was very little difference between the results of the three phases presented in Table 4.8. This is a result of the combination of total raw gas input and available processing capacity: when NGPUs-B/D are put in standby, the other process units needed to operate near their maximum capacity in order to admit all the raw gas total flowrate. As a consequence, once the binary variables y were defined by the MINLP algorithm, the refinement phase did not have much flexibility to change the continuous variables u.

Those results are consistent considering the economic scenario defined in this analysis and the characteristics of the industrial plant process units. It shows that the simulation and optimization framework built in this work is robust and flexible enough to: adapt to different market scenarios, evaluate the whole space of variables (see, for instance, u_2, u_3, u_8, u_{11} and u_{12} results from Table 4.8) and successfully deal with more than 60 inequality constraints, searching and finding an optimum operating point. Therefore it corroborates the methodology proposed and implemented.

Chapter 5

Conclusions and Suggestions for Future Works

This work had the objective of the simulation and optimization of an industrial site for natural gas processing with multiple process units and feedstock, utilizing an actual industrial site as study case. The mathematical modeling was successfully built up in HYSYS commercial process simulator and integrated to a data transfer interface in Excel/VBA. The simulation was validated against industrial real data and showed good agreement at an average discrepancy below 4.0 %. These results corroborate the choice of HYSYS as process simulator, Peng Robinson as thermodynamic model and COSTALD as liquid density correlation and show that even though there are no other works like this in the literature, it is possible to carry out optimization procedures considering a complete integrated natural gas processing industrial site keeping the use of rigorous simulations.

An optimization framework was formulated considering a high-level integrated perspective of the natural gas processing facility. This resulted in an MINLP optimization problem, aiming at the optimum operating point for maximum business profit. The twelve NGPUs raw gas feed rates and residual gas reprocessing flowrates were defined as the continuous decision variables and the seven operating status of the NGPUs were included as binary variables. The MINLP model was thrivingly broken down into an NLP combined with MIP, both solved as a simulation-optimization integrated framework, using HYSYS for simulation, *Python* for optimization and Excel as data transfer interface.

An augmented objective function was used in order to incorporate inequality constraints of the problem, as a strategy to guarantee that the physical limits of the plant and quality restrictions were met. The NLP problem was successfully solved using two derivative-free methods coupled in series: global PSO followed by local Nelder-Mead algorithm, indicating a potential gain of 7.3 % in variable profit. Five sets of initial estimates were tested using Nelder-Mead and led to five different optimal points, what

indicate the presence of local minima. Due to that, a pre-screening step was included in the optimization, using PSO global method to determine a promising initial estimate set for the problem.

The MINLP problem was successfully solved using branch-and-bound coupled with Nelder-Mead, preceded by a PSO pre-screening, in 23h 22min in an Intel ® Core^{TM} i5 10210U processor with 1.60 GHz. An LP problem was formulated and solved with interior-point method in order to generate feasible initial estimates for the Nelder-Mead. The MINLP results were then refined in a posterior step that used PSO in series with Nelder-Mead method, considering tighter tolerances for both the NLP problem and the process simulation. The results were in agreement with the graphical analysis conducted via PSO executions. Both resulted in disfavoring the three process units that have the lowest liquid recovery fractions, a consequence of the products sales prices defining higher values for liquid streams. An objective function value of 0.9948 was obtained, in comparison to 0.9120 of base-case, indicating a potential increase of 9.1 % or 118.78 R$/1000 m^3 in the industrial plant variable profit.

Lastly, a case study was performed in order to evaluate the MINLP optimization results in a simulated scenario of gas streams price four times higher than the base-case. The results were opposite from the ones of the base-case, as expected: the process units having the lowest liquid recovery fractions were prioritized, while NGPUs-B/D were disfavored, since they have the highest liquid recovery. Those results corroborated the robustness and flexibility of the methodology proposed and the simulation/optimization framework developed and built in this work, illustrating its ability to adapt to different market scenarios.

In the light of the methodology proposed and results presented, this work provided a contribution to the literature by successfully proposing an MINLP optimization framework and methodology for the business economic optimization of a natural gas processing site. It also makes way for new works and developments in the natural gas processing research field, such as:

- Include the so called *operating modes* of the NGPUs — gas or liquid preferential production, defined by temperature profiles — as binary decision variables.
- Build Pareto analysis based on product market price variation.
- Integrate the simulation and optimization models to plant data historian in order to acquire real-time process data, building a prototype of a plant Digital Twin.
- Test other global optimization methods instead of PSO.

Appendix A

PSO Python Code

```
import numpy as np
import xlwings as xw


def pso(func, lb, ub, ieqcons=[], f_ieqcons=None, args=(),
    kwargs={},
        swarmsize=100, omega=0.5, phip=0.5, phig=0.5,
            maxiter=100,
        minstep=1e-8, minfunc=1e-8, debug=False):

    """
    Perform a particle swarm optimization (PSO)

    Parameters
    ==========
    func : function
        The function to be minimized
    lb : array
        The lower bounds of the design variable(s)
    ub : array
        The upper bounds of the design variable(s)

    Optional
    ========
    ieqcons : list
        A list of functions of length n such that ieqcons[j
            ](x,*args) >= 0.0 in
```

```
    a successfully optimized problem (Default: [])
f_ieqcons : function
    Returns a 1-D array in which each element must be
        greater or equal
    to 0.0 in a successfully optimized problem. If
        f_ieqcons is specified,
    ieqcons is ignored (Default: None)
args : tuple
    Additional arguments passed to objective and
        constraint functions
    (Default: empty tuple)
kwargs : dict
    Additional keyword arguments passed to objective
        and constraint
    functions (Default: empty dict)
swarmsize : int
    The number of particles in the swarm (Default: 100)
omega : scalar
    Particle velocity scaling factor (Default: 0.5)
phip : scalar
    Scaling factor to search away from the particle's
        best known position
    (Default: 0.5)
phig : scalar
    Scaling factor to search away from the swarm's best
         known position
    (Default: 0.5)
maxiter : int
    The maximum number of iterations for the swarm to
        search (Default: 100)
minstep : scalar
    The minimum stepsize of swarm's best position
        before the search
    terminates (Default: 1e-8)
minfunc : scalar
    The minimum change of swarm's best objective value
        before the search
    terminates (Default: 1e-8)
debug : boolean
```

```
        If True, progress statements will be displayed
            every iteration
        (Default: False)

    Returns
    =======
    g : array
        The swarm's best known position (optimal design)
    f : scalar
        The objective value at ``g``

    """

#MODIFICATION ########################################
    #Excel Connection
    wb = xw.Book('UTGCAB_Interface_v1.7.4_OT.xlsm') #
        Simulation-Optimization Integration Tool file

    #Specify the number of decision variables for
        partitioned generation
        ######################################
         n  = 12     #number of continuous decision variables
    m  = 7      #number of binary decision variables
    nm = n + m #number of total decision variables

#####################################################


    assert len(lb)==len(ub), 'Lower- and upper-bounds must
        be the same length'
    assert hasattr(func, '__call__'), 'Invalid function
        handle'
    lb = np.array(lb)
    ub = np.array(ub)
    assert np.all(ub>lb), 'All upper-bound values must be
        greater than lower-bound values'

    vhigh = np.abs(ub - lb)
    vlow  = -vhigh
```

```
# Check for constraint function(s)
    #########################################
obj = lambda x: func(x, *args, **kwargs)
if f_ieqcons is None:
    if not len(ieqcons):
        if debug:
            print('No constraints given.')
        cons = lambda x: np.array([0])
    else:
        if debug:
            print('Converting ieqcons to a single
                constraint function')
        cons = lambda x: np.array([y(x, *args, **kwargs
            ) for y in ieqcons])
else:
    if debug:
        print('Single constraint function given in
            f_ieqcons')
    cons = lambda x: np.array(f_ieqcons(x, *args, **
        kwargs))

def is_feasible(x):
    check = np.all(cons(x)>=0)
    return check

# Initialize the particle swarm
    ############################################
S = swarmsize
D = len(lb)  # the number of dimensions each particle
    has
x = np.random.rand(S, D)  # particle positions
v = np.zeros_like(x)  # particle velocities
p = np.zeros_like(x)  # best particle positions
fp = np.zeros(S)  # best particle function values
g = []  # best swarm position
fg = 1e100  # artificial best swarm position starting
    value
```

```
for i in range(S):
    # Initialize the particle's position
    #x[i, :] = lb + x[i, :]*(ub - lb)  ---> original
       Python code

  #MODIFICATION:
    ########################################
     #IMPLEMENTING PARTITIONED PARTICLE GENERATION
    x[i, n:nm] = lb[n:nm] + x[i, n:nm]*(ub[n:nm] - lb[n
       :nm]) #generate integer variables

    #OPTIONAL: put one or more NGPUs in standby
    #x[i,14] = 0 #example: set NGPU-A in standby

       #Data communication with the Simulation-
          Optimization Integration Tool
    for j in range(m):
        xw.Range('StatusOP_OT')[j].value = x[i,n+j]

    Qmin =  np.array(xw.Range('Qmin_OT').value)
    Qmax =  np.array(xw.Range('Qmax_OT').value)

        #update lower and upper bounds
    lb[0:n] = Qmin
    ub[0:n] = Qmax

    x[i, 0:n] = lb[0:n] + x[i, 0:n]*(ub[0:n] - lb[0:n])
        #generate continuous variables

       #
          ####################################################




    # Initialize the particle's best known position
    p[i, :] = x[i, :]

    # Calculate the objective's value at the current
       particle's
```

```
        fp[i] = obj(p[i, :])

        # At the start, there may not be any feasible
            starting point, so just
        # give it a temporary "best" point since it's
            likely to change
        if i==0:
            g = p[0, :].copy()

        # If the current particle's position is better than
            the swarm's,
        # update the best swarm position
        if fp[i]<fg and is_feasible(p[i, :]):
            fg = fp[i]
            g = p[i, :].copy()


    #MODIFICATION
        ########################################
        # Insert vhigh and vlow calculation inside the loop
        vhigh = np.abs(ub - lb)
        vlow = -vhigh

    ######################################################

        # Initialize the particle's velocity
        v[i, :] = vlow + np.random.rand(D)*(vhigh - vlow)


# Iterate until termination criterion met
    ##################################
it = 1
while it<=maxiter:
    rp = np.random.uniform(size=(S, D))
    rg = np.random.uniform(size=(S, D))
    for i in range(S):

        # Update the particle's velocity
        v[i, :] = omega*v[i, :] + phip*rp[i, :]*(p[i,
```

```
    :] - x[i, :]) + \
            phig*rg[i, :]*(g - x[i, :])

 # Update the particle's position, correcting
    lower and upper bound
 # violations, then update the objective
    function value
 x[i, :] = x[i, :] + v[i, :]

#MODIFICATION
   ########################################
         #Update lower and upper bounds
 Qmin =  np.array(xw.Range('Qmin_OT').value)
 Qmax =  np.array(xw.Range('Qmax_OT').value)

 lb[0:n] = Qmin
 ub[0:n] = Qmax

#
   ######################################################



 mark1 = x[i, :]<lb
 mark2 = x[i, :]>ub
 x[i, mark1] = lb[mark1]
 x[i, mark2] = ub[mark2]
 fx = obj(x[i, :])

 # Compare particle's best position (if
    constraints are satisfied)
 if fx<fp[i] and is_feasible(x[i, :]):
     p[i, :] = x[i, :].copy()
     fp[i] = fx

     # Compare swarm's best position to current
        particle's position
     # (Can only get here if constraints are
        satisfied)
```

```
                if fx<fg:
                    if debug:
                        print('New best for swarm at
                            iteration {:}: {:} {:}'.format(
                            it, x[i, :], fx))

                    tmp = x[i, :].copy()
                    stepsize = np.sqrt(np.sum((g-tmp)**2))
                    if np.abs(fg - fx)<=minfunc:
                        print('Stopping search: Swarm best
                            objective change less than {:}'.
                            format(minfunc))
                        return tmp, fx
                    elif stepsize<=minstep:
                        print('Stopping search: Swarm best
                            position change less than {:}'.
                            format(minstep))
                        return tmp, fx
                    else:
                        g = tmp.copy()
                        fg = fx

        if debug:
            print('Best after iteration {:}: {:} {:}'.
                format(it, g, fg))
        it += 1

    print('Stopping search: maximum iterations reached -->
        {:}'.format(maxiter))

    if not is_feasible(g):
        print("However, the optimization couldn't find a
            feasible design. Sorry")
    return g, fg
```

Appendix B

LP Python Code

```
"""
Created on Tue Jul 12 08:52:55 2022

author: Tayna Embirucu Gonsalves de Souza

Objective: generate a feasible initial estimate for the NLP
    optimization problem to be solved

Methodology:
    The code solves an LP problem utilizing interior-point
       method.
    The integer variables are set to constant values, which
        feasible ranges are known.
    Then, the continuous variables are found by the LP
       method solution.

Equations:
    min c^T * x
    min f = - sum(Q[1] to Q[9]) --> slack variable
       associated to NGPU-C lower bound was used as
       objective function

    s.t.
    Ax <= b
    [Q,y] <= [Qmax, Ymax] --> upper bounds of the decision
       variables
    [-Q,-y] <= [-Qmin, -Ymin] --> lower bounds of the
```

```
    decision variables
Qinput - Qcarga <= Qmax_NGPU-C --> NGPU-C upper bound
    Qinput - Qcarga >= Qmin_NGPU-C --> NGPU-C lower
        bound
Q[3] + Q[4] + Q[5] + Q[6] <= Qmax[6] --> NGPU-E upper
    bound
Q[0] + Q[4] <= Qmax[0] --> Q1 upper bound

in which,
Qinput = parameter (total raw gas feed flowrate)

OBS: The integer variables were set as constant a priori,
    because otherwise it would become an NLP prolem. An
    external loop ensures that, in case a feasible solution
    is not found for the set of y defined, the LP problem
    will be executed again, for higher y values, until a
    feasible solution is found or y upper bound is reached,
    indicating that there is no feasible solution..

"""
import numpy as np
import xlwings as xw
from scipy.optimize import linprog

def EstInicialLP(n,mvar,ytot,ypos,Qinput):

        # n:      number of continuous decision variables
        # mvar:   number of relaxed integer variables
        # ytot:   vector containing all binary variables
        # ypos:   vector containing the position of the
            integer values of y
        # Qinput: value of total raw gas input flowrate

    # Excel connection
    wb = xw.Book('UTGCAB_Interface_v1.7.4_SIP_OT.xlsm')

    m = 0
    yfix_len = len(ypos)
    mtot = mvar + yfix_len
```

```
nm = n + mtot

Abound = np.identity(n+m)

#y bounds
Ymin = np.zeros(m)
Ymax = np.ones(m)

success = False # flag indicating if the LP problem
   found a feasible solution
nIter = 0       # number of LP solution attempts
feasible = 1
ncount = 0
nFix = 0
while success == False:

    #defining the values of y
    if nIter == 0:
        y0 = np.ones(mvar) * 0.65

    else:
        y0 = y0 * 1.05

        if max(y0) > 1 and ncount == 0:
            y0 = np.ones(mvar)
            ncount = 1

    if yfix_len == mtot:
        yExcel = ytot
    else:
        i = 0
        j = 0
        k = 0
        yExcel = np.zeros(mtot)
        while k < len(ypos) :
            if i == ypos[k]:
                yExcel[i] = ytot[ypos[k]]
                k = k+1
            else:
```

```
                yExcel[i] = y0[j]
                j = j + 1
            i = i + 1
        yExcel[i:] = y0[j:]

    if yfix_len < mtot:
        if max(y0) > 1:
            print('No feasible solution found for the
                LP problem.')
            feasible = 0
            return np.concatenate([np.zeros(n),y0]),
                feasible
    else:
        if nFix == 0:
            nFix = nFix + 1
        else:
            print('No feasible solution found for the
                LP problem.')
            feasible = 0
            return np.concatenate([np.zeros(n)]),
                feasible

    #Simulation-Optimization Integration Tool
       communication
    for i in range(mtot):
        xw.Range('StatusOP_OT')[i].value = yExcel[i]

    #bounds of the continuous variables
    Qmin = xw.Range('Qmin_OT').value
    Qmax = xw.Range('Qmax_OT').value

    #A matrix
    A = np.vstack([+Abound, \
                   -Abound, \
                   [0,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,0,*
                      np.zeros(m)], \
                                        [0,+1,+1,+1,+1,+1,+1,+1,+1,+
                                           np.zeros(m)],
                                            \
```

```
                    [0,+0,+0,+1,+1,+1,+1,+0,+0,+0,0,0,*
                        np.zeros(m)], \
                    [1,+0,+0,+1,+0,+0,+0,+0,+0,+0,0,0,*
                        np.zeros(m)]])

    #Lower and upper bounds
    Wmin = np.concatenate([Qmin,Ymin])
    Wmax = np.concatenate([Qmax,Ymax])

    #b matrix
    b = np.hstack([[*Wmax], \
                    [*-Wmin], \
                                        [Qinput - 1e6],
                                                \
                    [6e6-Qinput],           \
                    [Qmax[6]],              \
                    [Qmax[0]]])

    c = -np.array([0,1,1,1,1,1,1,1,1,1,0,0,*np.zeros(m)
        ])

    #Call linprog for solution (interior-point method
        default)
    res = linprog(c,A,b)

    success = res.success
    nIter = nIter + 1

wtot = np.concatenate([res.x,y0])

print('Number␣of␣attempts␣(y)␣=␣' + str(nIter))
print('fObj␣=␣' + str(res.fun))
print('Solution␣found:␣' + str(wtot))

return wtot, feasible
```

Appendix C

Simulation Mass Balance: Base-Case

This Chapter presents further detail on the process simulation, with information on the mass balance per NGPU and global, utilizing the base-case scenario (initial estimate I03, Table 4.3) as reference.

Since all the flowrate values presented in this work are normalized between zero and one, the mass balances, presented in Tables C.1 and C.2, are expressed in terms of product recovery, a percentage of the respective process unit feed gas flowrate.

Table C.1: Mass balance of the natural gas processing site per NGPU. Values expressed in product recovery, calculated as a percentage of the NGPU feed gas flowrate.

Product	NGPU-A (u_8)	NGPU-B (u_9)	NGPU-C	NGPU-D (u_{10})	NGPU-E ($u_4 + u_5 + u_6 + u_7$)
Sales gas	71.60 %	79.90 %	79.95 %	79.85 %	78.76 %
Residual gas	14.96 %	-	-	-	14.06 %
NGL (C_2^+)	-	20.02 %	19.97 %	20.03 %	-
NGL (C_3^+)	13.38 %	-	-	-	-
LPG	-	-	-	-	5.82 %
C_5^+	-	-	-	-	0.84 %

Table C.2: Overall mass balance of the natural gas processing site. Values expressed in product recovery, calculated as a percentage of the overall raw gas input flowrate.

Product	Overall site
Sales gas	85.45 %
Residual gas	-
NGL (C_2^+)	10.90 %
NGL (C_3^+)	-
LPG	1.87 %
C_5^+	0.50 %
Raw gas for external processing	0.00 %
Fuel gas	1.81 %
Flare gas	0.25 %

Bibliography

ABUNAHMAN, S. S., DOS SANTOS, L. C., TAVARES, F. W., et al., 2020, "A computational tool for parameter estimation in EoS: New methodologies and natural gas phase equilibria calculations", *Chemical Engineering Science*, v. 215, pp. 115437. ISSN: 00092509. Availability: <https://doi.org/10.1016/j.ces.2019.115437>.

AGENCY, I. E., 2020, *Implementing Gas Market Reform in Brazil*. Relatório técnico. Availability: <https://iea.blob.core.windows.net/assets/e552c7ce-d35d-4e09-8cf9-a7f2a38ff50b/ImplementingGasMarketReformsinBrazil-InsightsfromEuropeanexperience.pdf>.

AL-SOBHI, S. A., ELKAMEL, A., 2015, "Simulation and optimization of natural gas processing and production network consisting of LNG, GTL, and methanol facilities", *Journal of Natural Gas Science and Engineering*, v. 23, n. 2015, pp. 500–508. ISSN: 18755100. Availability: <http://dx.doi.org/10.1016/j.jngse.2015.02.023>.

ANDERSEN, E. D., ANDERSEN, K. D., 1999, "The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm", *High Performance Optimization*, v. 54, n. 33, pp. 197–232.

ANP, 2008. "Resolução ANP no 16, de 17.6.2008, DOU 18 de junho de 2008". .

ANP, 2020. "Resolução ANP no 825, de 17.6.2020, DOU 28 de agosto de 2020". .

BANSAL, S., MANJARE, S. D., 2016, "Theoretical Investigation on the Performance of Multicomponent Distillation Column for the Separation of Hydrocarbon Mixture Using Inside out Approach", *International Journal of Chemical Engineering and Applications*, v. 7, n. 4, pp. 282–288. ISSN: 20100221.

BARKER, W. J., CONWAY, P. A., 2007, "Convergent simplex searches and the 'gloveless DataGlove'", *Inverse Problems*, v. 23, n. 1, pp. 405–419. ISSN: 02665611.

BIDART, A. M. F., DE MELO, D. C., PASSARELLI, F. M., et al., 2015, "Challenges in Offshore Natural Gas Processing", *Offshore Technology Conference*.

BIEGLER, L. T., 2010, *Nonlinear Programming: Concepts, Algorithms, and Applications do Chemical Process*. First ed. United States of America, Society for Industrial ad Applied Mathematics and Mathematical Optimization Spciety. ISBN: 9780898717020.

BIGDOLI, A. A., 2018. "Simulation and Optimization of Primary Oil and Gas Processing Plat of FPSO Operating in Pre-Salt Oil Field". .

BULLIN, K. A., CHIPPS, J., 2005, "Optimization of natural gas gathering systems and gas plants", *GPA Annual Convention Proceedings*. ISSN: 00968870.

BULLIN, K. A., HALL, K. R., 2000, "Optimization of natural gas processing plants including business aspects", *GPA Annual Convention Proceedings*, pp. 1–12. ISSN: 00968870.

CAMPBELL, J. M., 2013, *Gas Conditioning and Processing*, v. 1. ISBN: 9788578110796.

CAMPOS, M., GOMES, M., SOUZA, A., et al., 2012, "Optimisation of natural gas plant - Gains in profitability, stability and energy efficiency", *International Gas Union World Gas Conference Papers*, v. 3, pp. 2089–2120.

DELOU, P. A., RIBEIRO, L. D., PAIVA, C. R., et al., 2021, "A Real-Time Optimization Strategy for Small-Scale Facilities and Implementation in a Gas Processing Unit", *Processes*, v. 9, pp. 1179.

DOS SANTOS, L. C., ABUNAHMAN, S. S., TAVARES, F. W., et al., 2015, "Cubic Plus Association Equation of State for Flow Assurance Projects", *Industrial and Engineering Chemistry Research*, v. 54, n. 26, pp. 6812–6824. ISSN: 15205045.

DOS SANTOS, L. C., TAVARES, F. W., AHÓN, V. R. R., et al., 2015a, "Modeling MEA with the CPA equation of state: A parameter estimation study adding local search to PSO algorithm", *Fluid Phase Equilibria*, v. 400, pp. 76–86. ISSN: 03783812. Availability: <http://dx.doi.org/10.1016/j.fluid.2015.05.004>.

DOS SANTOS, L. C., ABUNAHMAN, S. S., TAVARES, F. W., et al., 2015b, "Modeling Water Saturation Points in Natural Gas Streams Containing CO2 and H2S - Comparisons with Different Equations of State", *Industrial Engineering Chemistry Research*, v. 54, pp. 743–757.

DUBREUIL, J.-B., 2022, “Gas Market Report”, *Gas, Coal and Power Markets Division (GCP)*. ISSN: 08029474.

ECONOMOU, I. G., DONOHUE, M. D., 1996, “Equations of state for hydrogen bonding systems”, *Fluid Phase Equilibria*, v. 116, n. 1-2, pp. 518–529. ISSN: 03783812.

FRANCO, S. V. D. A., DA CUNHA RIBEIRO, D., MENEGUELO, A. P., 2020, “A comprehensive approach to evaluate feed stream composition effect on natural gas processing unit energy consumption”, *Journal of Natural Gas Science and Engineering*, v. 83, n. September, pp. 103607. ISSN: 18755100. Availability: <https://doi.org/10.1016/j.jngse.2020.103607>.

GAO, F., HAN, L., 2012, “Implementing the Nelder-Mead simplex algorithm with adaptive parameters”, *Computational Optimization and Applications*, v. 51, n. 1, pp. 259–277. ISSN: 09266003.

GONG, J., YOU, F., 2017, “A New Superstructure Optimization Paradigm for Process Synthesis with Product Distribution Optimization: Application to an Integrated Shale Gas Processing and Chemical Manufacturing Process”, *AIChE Journal*, v. 64.

HANKINSON, R. W., THOMSON, G. H., 1979, “Thermodynamics of Multicomponent Liquid Mixtures Con-taining Subcritical and Supercritical Components A New Correlation for Saturated Densities of Liquids and Their Mixtures”, *Ind. Eng. Chem. Fundamentals*, v. 25, n. 4, pp. 653–663.

HANKINSON, R. W., COKER, T. A., THOMSON, G. H., 1982, “Get accurate LNG densities with COSTALD”, *Hydrocarbon Process*, v. 62.

HUANG, S. H., RADOSZ, M., 1991, “Equation of State for Small, Large, Polydisperse, and Associating Molecules: Extension to Fluid Mixtures”, *Industrial and Engineering Chemistry Research*, v. 30, n. 8, pp. 1994–2005. ISSN: 15205045.

ISO, 2004, *ISO 18453: Natural gas - Correlation between water content and water dew point*. First ed. International Organization for Standardization.

KENNEDY, J., EBERHART, R., 1995, “Particle Swarm Optimization”, *IEEE*, pp. 1942–1948.

KIANFAR, K., 2011, “Branch-and-Bound Algorithms”. In: *Wiley Encyclopedia of Operations Research and Management Science*, n. January 2011. ISBN: 9780470400531.

KONTOGEORGIS, G. M., FOLAS, G. K., 2009, *Thermodynamic Models for Industrial Applications: From Classical and Advanced Mixing Rules to Association Theories*. ISBN: 9780470747537.

KONTOGEORGIS, G. M., VOUTSAS, E. C., YAKOUMIS, I. V., et al., 1996, "An equation of state for associating fluids", *Industrial and Engineering Chemistry Research*, v. 35, n. 11, pp. 4310–4318. ISSN: 08885885.

LAND, A. H., DOIG, A. G., 1960, "An Automatic Method of Solving Discrete Programming Problems", *Econometrica*, v. 28, n. 3, pp. 497–520.

LEE, K. Y., EL-SHARKAWI, M. A., 2008, "Fundamentals of Particle Swarm Optimization Techniques". In: *Modern Heuristic Optimization Techniques*, Wiley, cap. Chapter 5, pp. 71–88.

LOPEZ-ECHEVERRY, J. S., REIF-ACHERMAN, S., ARAUJO-LOPEZ, E., 2017, "Peng-Robinson equation of state: 40 years through cubics", *Fluid Phase Equilibria*, v. 447, pp. 39–71. ISSN: 03783812. Availability: <http://dx.doi.org/10.1016/j.fluid.2017.05.007>.

MAZUMDER, M., XU, Q., 2020, "Modeling and Optimization for a Comprehensive Gas Processing Plant with Sensitivity Analysis and Economic Evaluation", *Chemical Engineering and Technology*, v. 43, n. 11, pp. 2198–2207. ISSN: 15214125.

MENCARELLI, L., CHEN, Q., PAGOT, A., et al., 2020, "A review on superstructure optimization approaches in process system engineering", *Computers and Chemical Engineering*, v. 136. ISSN: 00981354.

MME, ANP, PRÉ-SAL PETRÓLEO, et al., 2020, *Estudo sobre o Aproveitamento do Gás Natural do Pré-Sal*. Relatório técnico. Availability: <http://www.anp.gov.br/arquivos/estudos/aproveitamento-gn-pre-sal.pdf>.

MOKHATAB, S., POE, W. A., MAK, J. Y., 2012, *Handbook of Natural Gas Transmission and Processing*. Elsevier Inc.

MOLNAR, G., BERKENWALD, M., 2021. "Novo Mercado de Gás – The Brazilian gas market enters a new era". Availability: <https://www.iea.org/commentaries/novo-mercado-de-gas-the-brazilian-gas-market-enters-a-new-era>.

MONDAL, S. K., UDDIN, M. R., AZAD, A. K., 2013, "Simulation and Optimization of Natural Gas Processing Plant", *International Conference on Mechanical, Industrial and Materials Engineering*.

MORAES, A. O., MITRE, J. F., LAGE, P. L., et al., 2014, "A robust parallel algorithm of the particle swarm optimization method for large dimensional engineering problems", *Applied Mathematical Modelling*, v. 39, n. 14, pp. 4223–4241. ISSN: 0307904X.

MORRISON, D. R., JACOBSON, S. H., SAUPPE, J. J., et al., 2016, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning", *Discrete Optimization*, v. 19, pp. 79–102. ISSN: 15725286. Availability: <http://dx.doi.org/10.1016/j.disopt.2016.01.005>.

MURALI, A., BERROUK, A. S., DARA, S., et al., 2020, "Efficiency enhancement of a commercial natural gas liquid recovery plant: A MINLP optimization analysis", *Separation Science and Technology (Philadelphia)*, v. 55, n. 5, pp. 955–966. ISSN: 15205754. Availability: <https://doi.org/10.1080/01496395.2019.1574825>.

NELDER, J. A., MEAD, R., 1965, "A Simplex Method for Function Minimization", *The Computer Journal*, v. 7, n. 4, pp. 308–313. ISSN: 0010-4620.

NUCHITPRASITTICHAI, A., CREMASCHI, S., 2011, "Optimization of CO2 capture process with aqueous amines using response surface methodology", *Computers and Chemical Engineering*, v. 35, n. 8, pp. 1521–1531. ISSN: 00981354. Availability: <http://dx.doi.org/10.1016/j.compchemeng.2011.03.016>.

PENG, D. Y., ROBINSON, D. B., 1976, "A New Two-Constant Equation of State", *Industrial and Engineering Chemistry Fundamentals*, v. 15, n. 1, pp. 59–64. ISSN: 01964313.

PEREIRA, E. C., 1999, "Algoritmo Numerico para a Solucao da Programacao Mista Linear e Inteira", *Master Thesis. Universidade Federal do Rio Grande do Sul*, p. 110.

PESCHEL, W., WENZEL, H., 1984, "Equation-of-State Predictions of Phase Equilibria At Elevated Pressures in Mixtures Containing Methanol." *Berichte der Bunsengesellschaft/Physical Chemistry Chemical Physics*, v. 88, pp. 807–812.

POE, W. A., MOKHATAB, S., 2017, *Modeling, Control, and Optimization of Natual Gas Processing Plants*. Elsevier Inc. ISBN: 978-0-12-802961-9.

RIBEIRO, L. D., DELOU, P. A., SANTOS, L. L. D., et al., 2020, "SORAIA: A PETROBRAS System of Revenue Optimization and Artificial Intelligence", *Proceedings of the Rio Oil & Gas Expo and Conference*, pp. 1–11.

SEADER, J. D., HENLEY, E. J., ROPPER, D. K., 2011, *Separation process principles: chemical and biochemical operations*. John Wiley & Sons Inc. ISBN: 9780470481837.

SELOT, A., KUOK, L. K., ROBINSON, M., et al., 2008, "A Short-Term Operational Planning Model for Natural Gas Production Systems", *AIChE Journal*, v. 54, n. 2, pp. 495–515. ISSN: 12350621.

SELVAN, K. K., PANDA, R. C., 2018, "Mathematical Modeling, Parametric Estimation, and Operational Control for Natural Gas Sweetening Processes", *ChemBioEng Reviews*, v. 5, n. 1, pp. 57–74. ISSN: 21969744.

SILVA, M. S., S., B. L., PONTES, K. V., et al., 2021, "Avaliação Técnica de Membranas Poliméricas para Captura de CO2 do Gás Natural do Pré-Sal Brasileiro", v. 23.

SOARES, R. D. P., 2001. "Modificação e Implementação de Algoritmos para Problemas MINLP". .

SULEMAN, H., MAULUD, A. S., MAN, Z., 2015, "Review and selection criteria of classical thermodynamic models for acid gas absorption in aqueous alkanolamines", *Reviews in Chemical Engineering*, v. 31, n. 6, pp. 599–639. ISSN: 01678299.

TECHNOLOGY, A., 2019. "Aspen HYSYS Help". .

THOMSON, G. H., BROBST, K. R., HANKINSON, I. . W., 1982, "An Improved Correlation Compressed Liquids and Liquid Mixtures", *AIChE Journal*, v. 28, n. 4, pp. 671–676.

ZHANG, B. J., CHEN, Q. L., LI, J., et al., 2016, "Operational Strategy and Planning for Raw Natural Gas Refining Complexes: Process Modeling and Global Optimization", *AIChE Journal*, v. 00, n. 00, pp. 1–17. ISSN: 12350621.

ZHENG, Q. P., REBENNACK, S., ILIADIS, N. A., et al., 2010, "Optimization Models in the Natural Gas Industry", , n. June 2010, pp. 121–148.