

Chemical Process Design under Uncertainty –Models and Algorithms for Global Optimization

BY

JEONGHWA MOON

B. S., Pohang University of Science and Technology, Pohang, Korea, 2000

M. S., Pohang University of Science and Technology, Pohang, Korea, 2002

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Chemical Engineering
in the Graduate College of the
University of Illinois at Chicago, 2017
Chicago, Illinois

Defense Committee:

Dr. Andreas A. Linninger, Chair and Advisor

Dr. G. Ali. Mansoori

Dr. John R. Regalbuto

Dr. Urmila Diwekar, Bioengineering

Dr. Donald J. Chmielewski, Illinois Institute of Technology

Dr. Fouad Teymour, Illinois Institute of Technology

Acknowledgement

First of all, I would like to express my special thanks my advisor, Dr. Andreas A. Linninger. He inspired me to come into the world of this research.

I express my sincere thanks to Dr. Zhang, senior staff at Laboratory for Product and Process Design (LPPD) for offering support in applied mathematics and optimization techniques. His individual expertise helped me to trouble shoot typical research problems in a systematic fashion and hone my technical skills.

I also thank my colleagues Gerardo, Seon, Brian and Alex for providing valuable suggestions and critical reviews of my work especially during group meetings which led room for further improvement.

Last, but not least, I will like to thank my wife, Nayoung and my daughter, Yewon for their support and love in the last months of the thesis preparation making possible to finish this work in form and time.

Contribution of Authors

In Chapter I, Dr. Kulkarni performed calculations of applications (Application 1-5, Section 4), with MINLP method to compare results of calculations using my algorithm. Dr. Zhang developed case problems to solve for showing the effectiveness of the algorithm presented in this chapter.

In Chapter II and III, Dr. Kim carried out simulations and verified all calculations done in Chapter II and III, independently. He also contributed to literature reviews for these chapters.

Dr. Linninger, my PHD advisor encouraged me to develop the theories and perform the calculations for Chapter I, III and V. He supervised all researches and writings, and revised the final version of manuscripts.

TABLES OF CONTENTS

CHAPTER I. Parallel Hybrid Algorithm for Process Flexibility Analysis	5
1 Overview of flexibility analysis.....	5
2 Methodology	8
2.1 Mathematical formulation of flexibility index problem	8
2.2 Hybrid algorithm	10
3 Nearest constraint projection technique.....	13
3.1 Illustration of a nonlinear two dimensional flexibility problem.....	15
3.2 Nearest constraint projection with controls	18
3.3 Implementation of the nearest constraint projection technique	20
3.4 Multiple controls.....	26
3.5 Overcoming problems with disjoint or complicated flexibility regions	30
3.6 Implementation	31
4 Applications	33
4.1 Application 1: Non-convex, nonlinear two dimensional problem.....	33
4.2 Application 2: Heat exchanger network with varying heating duties.....	37
4.3 Application 3: Pump and pipe with uncertainty in operating pressure.....	39
4.4 Application 4: Reactor and cooler system with uncertain kinetics.....	41
4.5 Application 5: Polymerization Reactor with uncertain kinetics	43
4.6 Parallelization	46
5 Conclusions.....	50
CHAPTER II. Overview of Embedded Control Optimization	56
1 Previous works and difficulties in integration of design and control	56
2 Mathematical programs decomposition for design under uncertainty.....	57
2.1 Level-1: Modeling and structural decisions	58
2.2 Level-2: Problem decomposition and optimization.....	59
3 Embedded control optimization.....	62
CHAPTER III. Design and Control Integration for Isomerization Process.....	67
1 Isomerization Process description	67
2 Mathematical models for simulation	69
3 Operational constraints and control action strategy.....	70
4 Design variables.....	72
5 Uncertainty scenarios.....	73
6 Control Structures	74
7 Performance of a Nominal Process.....	75
8 Optimal design process	79

9	Results.....	82
10	Discussion.....	89
	CHAPTER IV. Review of Global Optimization.....	95
1	Deterministic optimization methods.....	97
1.1	Interval method (Hansen, 1979)	97
1.2	Homotopy method	98
1.3	α BB method.....	98
1.4	Branch and reduced algorithm (BARON)	99
1.5	Global terrain method	99
2	Stochastic methods	100
2.1	Genetic Algorithms.....	101
2.2	Scatter search	102
2.3	Simulated annealing.....	103
2.4	Particle swarm optimization	104
3	Comparisons of global optimization methods	105
3.1	Deterministic vs stochastic methods.....	105
3.2	A comparison of complete global optimization solvers (Neumair et al, 2005) 106	
3.3	“No free lunch theorems for optimization” (Wolpert & Macready, 1997) 106	
	CHAPTER V. Hybrid Sequential Niche Algorithm for Problems with Solution Multiplicity	108
1	Introduction.....	108
1.1	Niche methods	109
1.2	Hybrid genetic algorithms	111
2	Methodology	111
2.1	Population size.....	114
2.2	Derating functions and modified fitness.....	114
2.3	Natural selection, mating, and mutation	115
2.4	Niche Detection and precise extremum location	116
2.5	Automatic niche radius adjustment	117
2.5.1	Variable Niche Radius.....	119
3	Benchmark test cases	125
3.1	Results	130
4	Realistic chemical engineering applications with solution multiplicity	133
4.1	The concentration and temperature profile of catalytic pellet.	133
4.2	Determination of reaction kinetics in catalytic reactors	136
5	Conclusions and future works.....	139
	Nomenclature (Chapter V).....	140
	APPENDICES	141

APPENDIX A- Mathematical Models of Isomerization Process	142
1 DIB column.....	142
2 Feed Effluent Heat exchanger.....	144
3 Purge Distillation column	145
4 Discretized plug flow reactor model.....	146
APPENDIX B- Introduction to Global Terrain Method.....	148
Summary	148
1 Global Terrain Method	148
2 Methodology	150
2.1 Governing Equations	150
2.2 Movement Between Stationary Points	151
2.2.1 Initial Movement	151
2.2.2 Downhill Movement.....	153
2.2.3 Uphill Movement.....	153
2.2.4 Predictor Step	155
2.2.5 Corrector Step.....	155
2.3 Saddle Detection.....	156
2.4 Termination.....	156
3 Whole procedure of Global Terrain method.....	157
4 Applications	158
4.1 Theoretical Applications.....	158
4.1.1 Example 1 – Nonlinear Equations	158
4.1.2 Example 2 – Polynomial Complex Roots.....	167
1.1 Chemical Engineering Applications	169
4.1.3 Example 3 – Nonlinear Continuous Stirred Tank Reactor	169
4.1.4 Example 4 – Catalytic Pellet Reactor	172
5 Conclusion	180
References.....	181
APPENDIX C- Permission to Reuse Published Materials	192

LIST OF TABLES

Table 1: Critical points and flexibility indices corresponding to different nominal points	17
Table 2: Critical points and flexibility indices corresponding to several nominal points for application 1. The super script * indicates the incorrect solution obtained with the MINLP method (TOMLAB).	34
Table 3: Heat exchanger variables, definitions and units.	37
Table 4: The variables of pump and pipe application.	39
Table 5: Nominal points and expected deviation of application 3	40
Table 6: Nominal points and expected deviation of application 4	42
Table 7: Nominal values and deviations of uncertain parameters of the polymerization reactor	45
Table 8: The observed flexibility indexes as a function of the increasing reactor volume	46
Table 9: Proposed decision hierarchy for integrated design and control (Malcolm et al, 2007)	58
Table 10: Variable types in integrated design and control	58
Table 11: Design, process, and manipulated variables.	71
Table 12: Dynamic disturbance scenarios	74
Table 13: Nominal values and expected deviation of uncertain parameters	74
Table 14: Best design with ten original samples and a critical point	85
Table 15: Parameter values in steady states.	85
Table 16: Test functions	126
Table 17: Results of Beasley's sequential niche technique	132
Table 18: Results of hybrid sequential niche algorithm	132

LIST OF FIGURES

Figure 1: Depiction of the discontinuity and non-differentiability of the flexibility index problem. This graph shows the feasible deviation index, δ , plotted for rays emanating from the nominal conditions with angle α . This non-differentiable δ space is problematic for gradient-based search methods.	7
Figure 2: Graphical representation of the feasible region boundary, $\psi(\mathbf{d}, \boldsymbol{\theta}) = 0$, and the critical point, $\boldsymbol{\theta}^*$. Geometrically, the flexibility index F , corresponds to the dimension of the hyper rectangle touching the feasible region boundary at the critical point. ..	9
Figure 3: Flowchart of the proposed hybrid algorithm.	12
Figure 4: A schematic depiction of crude samples and the one-dimensional projection to the feasible boundary, $\psi(\mathbf{d}, \boldsymbol{\theta}) = 0$. The crude samples generated by genetic inheritance do not lie of the flexibility boundary in general. One-dimensional projection brings the samples to the feasibility boundary.	14
Figure 5: Snapshot of for flexibility index problem. Samples coalesce to the critical point (5.7429, 2.2571).	16
Figure 6: Critical points and independent operational regions corresponding to different nominal points in the uncertain parameter space of example. The proposed method correctly identifies the size of hyper-rectangle in each situation.	17
Figure 7: Geometric representation of the feasible region, $\psi(\mathbf{d}, \boldsymbol{\theta}) \leq 0$, in the presence of control, (a) depicts constraint variation due to changing z values in the $\boldsymbol{\theta}$ space. (b) shows the projection of the z space onto the $\boldsymbol{\theta}$ space to obtain, $\psi(\mathbf{d}, \boldsymbol{\theta}) \leq 0$. The feasible region is the unified gray region made possible by permissible control adjustments z	19

Figure 8: Graphical representation of the <i>nearest constraint projection technique</i>	21
Figure 9: Three possible types for deciding correct direction of control moves. (a) For a type 1, $G^+ < 0$, $G^- > 0$. (b) For a type 2, $G^+ > 0$, $G^- < 0$. (c) For a type 3, $G^+ > 0$, $G^- > 0$	25
Figure 10: Flowchart of the nearest constraint projection technique with multiple controls. Projection with one control leads to the local extreme point (<i>local termination</i>). We continue the zig-zag search for other controls until we reach a point which cannot be moved in spite of changing controls.....	26
Figure 11: Graphical illustration of the example for the example in section 3.4. (a) This example has one parameter, θ , and two controls, z_1 and z_2 . (b) In step 1, projection in (θ, z_1) space yields the local termination point P_2 . (c) In step 2, projection in (θ, z_2) space yields the local termination point P_3 , which is also the global termination point P_4, P_5	29
Figure 12: Projection with multiple control level in the disjoint feasible region. To efficiently explore the feasible region boundary, multiple control level starting points are necessary to find candidate solutions on the feasible boundary in problems with complex constraints or disjoint feasibility spaces.	31
Figure 13: The results of application 1; (a) Seven nominal points and corresponding critical points found by our method. (b) The MINLP method failed when critical point did not lie on the vertex. Our method finds exact solutions even though the constraints are non-convex. The MINLP solver also required about 10-15 times the CPU times of our hybrid method.....	36
Figure 14: Heat exchanger network reproduced from Grossmann et al. (1986). F_{H1} is the uncertain parameter and Q_c is the control variable.	37

Figure 15: The feasible region of application 2 and the trajectory of the suggested projection technique starting from the point (1,15). Our method converged to the critical point $F_{H1}=1.118$, $Q_c=40.66$	38
Figure 16: The flowsheet for the centrifugal pump and pipe case study.....	39
Figure 17: The result for the pump and pipe application depicting the critical point ($m=11.233$, $P_2^*=923.408$) as a corner of the hyper rectangle touching the feasible region.	41
Figure 18: The flowsheet of first-order reactor system. It involves a heat exchanger and a reactor with first order exothermic reaction.	41
Figure 19: Normalized feasible region and critical point found using hybrid algorithm. We found critical point (85.19, 12.40) which lies on constraint g_1 . The control levels for this region is $T_1=389K$, $T_2=389K$, $T_{w2}=350K$	43
Figure 20: Polymerization reactor. (a) monomer: styrene (b) initiator: azo-bis-isobutyronitrile (AIBN), (c) solvent: benzene (d) the outlet stream: monomer, initiator, and the polymer (e) cooling jacket inlet (f) coolant efflux. The monomer, the initiator, the solvent are inlet streams and produce polymer.	44
Figure 21: A master-slave framework for parallel implementation of the hybrid algorithm: the master executes GA operations and distributes sample θ to slave nodes. The slave nodes project θ onto the region of boundary and evaluate the cost and send it back to master.	48
Figure 22: Computational time (left hand axis) vs number of processors in application 5 : 9 processor system is approximately one and half times faster than 6 process system, three times faster than 3 processor system, and nine times faster than a single	

processor system. The computational speed increases linearly with the number of processors. (right hand axis).....	49
Figure 23: Decomposition algorithm for integrated design and control under uncertainty.	
Main optimization problem (B) is separated from feasibility test (C). (Mohideen et al., 1996a)	60
Figure 24: Embedded optimization. A two-hierarchical loop renders more tractable optimization problem to determine the best economical design that has also dynamic controllability.	
64	
Figure 25: Moving horizon estimation-basic concept .The state is estimated from a horizon of the most recent $H+1$ output measurements that moves forward at each sampling time when a new measurement is available.	
66	
Figure 26: Isomerization process (Luyben et al., 1998). In this process, normal butane is converted to isobutane. The process is comprised of a reactor, a feed-effluent heat exchanger, and two distillation columns.	
68	
Figure 27: Design, process and manipulated variables in isomerization process case study.	
71	
Figure 28: Reactor size vs conversion ratio. Conversion ratio is almost linearly proportional to the reactor size. This means a larger reactor will increase the controllability.	
73	
Figure 29: Relationship between x and u . All variables are highly correlated.....	
75	
Figure 30: Classical control structure u_1 (reflux ratio of DIB) and u_2 (vapor ratio of DIB) are paired with x_1 (the mole fraction of nC_4 at the top of the DIB), u_1 (the reflux ratio of the DIB); u_3 (the vapor ratio of the purge column) is paired with x_2 (the mole	

fraction of nC_4 at the top of the DIB), and u_4 (the input temperature of the reactor) is paired with x_3 (the reactor pressure).	76
Figure 31: Result of the classical control structure. The classical control structure could not keep x_1 and x_2 under the set points until $t = 2500$	77
Figure 32: Result of the integrated control structure. It succeeded in keeping all process variables under the set points. Unlike the classical control structure, It used u_4 to control x_1 and x_2	78
Figure 33: Probability of two uncertain parameters. We chose 10 samples using Latin Hypercube method.....	79
Figure 34: Simulation result of the final optimal design (Case 1): (a) trajectories of process variables x_1 - x_3 and (b) trajectories of manipulated variables u_1 - u_4 . In the identification phase (phase I), the controller is not activated, because sufficient data have not been collected for the moving horizon estimator. Phase II describes startup behavior. The controller kept all process variables under the upper limits. When a dynamic disturbance occurred, all values of process variables are increased. However, they did not exceed the upper limits but reached a new steady state.	86
Figure 35: Simulation result of the final optimal design (Case 2). A feed composition change occurred at $t = 1500$, but the response of the embedded controller was satisfactory.....	87
Figure 36: Simulation result of the final optimal design (Case 3). In this case, two dynamic disturbances occurred at $t=1500$, which caused increases in x_1 and x_3 increased. However, the embedded controller adjusted the manipulated variables to keep all process values below the upper limits.	88

Figure 37: Example of a global maximum near local maximum locations.....	96
Figure 38: Multiple niches in two-dimensional search space. Each peak conforms a local extrema and niche area of attraction.....	110
Figure 39: Flowchart of the two-stage hybrid sequential niche algorithm. Genetic search of stage1 runs until termination condition is satisfied. When the majority of the individuals coalesce around the best-in-cluster, stage 2 activates a local optimizer. Taking the <i>BIC</i> solution as the initial guess, the local gradient-based method converges to the precise local minimum S . The niche radius is adapted according to the distance between <i>BIC</i> and the solution S	113
Figure 40: Example demonstrating difficulty with fixed radius strategies in hybrid genetic algorithms. If fixed radius is used with hybrid genetic algorithm, the local solution S ₂ can not be found, because genetic search repeatedly clusters close to niche around B ₃ or B ₂ . This problem is known as trapping.	119
Figure 41: Two different niche types. a) In hyper-spherical type, the niche radius size is $r = \ \mathbf{B} - \mathbf{S}\ + \alpha$. B ₁ is best-in-cluster (<i>BIC</i>) found by the genetic algorithm, and S is a real solution found by a local optimizer starting from B ₁ . b) In hyper-rectangular type, only one length side of rectangular is decided from <i>BIC</i> c) If same solution S is found from different initial point B ₂ , niche radius r is augmented to r_{new} . d) In the hyper rectangle strategy, the niche size is updated by taking each maximum length side.	122
Figure 42: The example of using variable radius size in hybrid genetic algorithm. a) The first solution, S ₁ is found from B ₁ b, c) Niches are updated from N_1 to N_2 , from N_2 to N_3 . d) Finally, S ₂ can be found from B ₄	124

Figure 43: Himmelblau's function (H). It has four maximum- \mathbf{S}_1 (3.58,-1.86), \mathbf{S}_2 (3.0,2.0), \mathbf{S}_3 (-2.815,3.125), and \mathbf{S}_4 (-3.78,-3.28).....	126
Figure 44: Two-dimensional Griewank's function (G). It has 17 equal minima in range of $[-10,10]^n$	127
Figure 45: One-dimensional Bohachevsky's function (B). It has 5 minima range of $-10 < x < 10$ and search space is asymmetric.	127
Figure 46: Two-dimensional modified Vincent's function (V). It has one global minimum and 35 local minima in range of $[0.25,10]^n$	128
Figure 47: Ackley's function (A), a) $n=1$ b) $n=2$, This function has one global minimum and 21 local minima.	129
Figure 48: Pellet concentration profiles evaluated by OCFE with hybrid sequential niche algorithms, where x is the dimensionless pellet radius and y is the dimensionless concentration in eq (60). For parameters sets in frame a) and b), our algorithm verified only one solution exists. The algorithm found three solutions for the parameters and conditions of the examples depicted in frame c-f. More details about these cases are described elsewhere (Kulkarni et al., 2008).....	135
Figure 49: Snapshots of individual distributions in pellet reactor inversion problem. a) Iteration 1: ten individuals are generated randomly. b) Iteration 6: every individual coalesces to point (0.36, 0.72) c) from point (0.36, 0.72), the solution \mathbf{S}_1 (0.40, 0.73) was obtained using a local optimizer. Then niche was set. d) Iteration 7; population is reinitialized to find the next solution, e) Iteration 15: every individual coalesces to point (0.76, 0.75). f) Finally, another solution \mathbf{S}_2 (0.80, 0.73) was obtained.	138

Summary

Uncertainty is a fundamental factor we should consider in process design because it is inherent characteristics of any process systems. However, classical process design uses only nominal information to find optimal design variables, and in reality, this is not suitable because these parameters have unavoidable variations, leading to uncertainty. Thus design is usually oversized to minimize risk of operating outside specifications to accommodate uncertainties. But this arbitrary overdesign does not always guarantee feasibility and optimality of the process. For this reason, novel consideration of uncertainty is necessary and important for the optimality and feasibility of operation of the chemical plant.

The first aim of this thesis is to develop novel methodologies to tackle problems of classical approach for design under uncertainty. Two main topics in design under uncertainty—flexibility analysis and integrations of design and control dealt with this thesis.

Part A addresses flexibility analysis of process. A new hybrid algorithm for flexibility analysis problem is suggested. Flexibility analysis is to quantify flexibility of a given process design to handle uncertainty in process parameters as well as variations in operating conditions. It is one of important problem in “design under uncertainty”. It is formulated as a multistage global optimization problem, whose search space is discontinuous and non-differentiable. Traditional local deterministic approaches cannot solve this problem properly, so I used a new approach based stochastic method and project technique to tackle this problem. This approach can be easily parallelized, so it reduces computational time when we solve large size problems.

In part B, the problem of integrating design and control is addressed. Integration of design and control is finding an optimal design considering dynamic controllability. It aims

at pursuing the synergetic power of a simultaneous approach to guarantee the economical and robust operation of the process in spite of any disturbance and uncertainty. However integration of design and control renders a complex combinatorial optimization problem which cannot be solved directly with existing mathematical methods. Thus we suggested a decomposition technique which eases the problems of this integration called embedded control optimization. In this thesis, I will extend embedded control optimization for integration of design and control. A new identification method is adopted to produce a better performance, and this methodology will be applied to large-scale processes successfully.

The second area this thesis considers is global optimization. Global optimization applications are widespread in all disciplines. Despite there are many challenging and important problems that require global solutions, relatively little effort has been made in this area compared to the area of local optimization. Specially, the problem of finding all solutions in nonconvex search area remains as still challenging and difficult area in applied mathematics, engineering, and sciences.

Part C addresses global optimization for multimodal objective functions. A novel hybrid sequential algorithm is suggested in this part. It aims to find multiple global solutions as well as local solutions. To locate multiple optimal points, it uses niche concept. It also adopts a local deterministic method to accelerate finding solutions. This algorithm was applied to tackle multiplicity problems in engineering problems such as finding multiple optimal parameters of distributed systems in problem inversion.

PART A: Flexibility Analysis under Uncertainty

Flexibility analysis is an important task for the optimal design and synthesis of chemical processes with uncertainty. It is a challenging problem because of the discontinuity and nonconvexity of rigorous flexibility programming formulations. In this article, we propose a new parallel hybrid algorithm based on stochastic search in conjunction with a nearest constraint projection technique to numerically solve the flexibility index problem. This method can be useful regardless of the convexity of the design constraints. The stochastic method robustly identifies the global solution without the need for derivative information. The new nearest constraint projection technique is used to handle the constraints of the flexibility index problem in reduced state space. In contrast to existing methods, this technique does not require the addition of artificial variables for active constraints, does not need to have access to explicit analytical forms of the problem formulation or its derivatives, and does not solve for additional artificial variables. Its implementation is well-suited for parallel computing so that computational time can be dramatically reduced. Five applications illustrate the efficacy of the proposed method.

CHAPTER I. PARALLEL HYBRID ALGORITHM FOR PROCESS FLEXIBILITY ANALYSIS

Previously published as:

Moon, J., Kulkarni, K., Zhang, L., and Linninger, A.A., “Parallel Hybrid Algorithm for Process Flexibility Analysis”, *Industrial and Engineering Chemistry Research*, 47 (21), 8324-8436, 2008.

1 Overview of flexibility analysis

Flexibility, along with the issues of operability, controllability, and reliability, are important design objectives for feasible and safe operation of chemical processes (Naraway & Perkins, 1993; Luyben & Floudas, 1994; Mohideen et al., 1996). Chemical plants should perform safely within acceptable quality ranges despite of operational disturbances and inherent uncertainties of the process parameters. For realistic process designs, it is important to quantify its degree of flexibility to handle uncertainty in process parameters as well as variations in operating conditions.

There are two kinds of mathematical programming methods for Rigorous flexibility analysis - the *feasibility test* and the *flexibility index* problems (Ostrovsky et al., 1994; Halemane & Grossmann, 1983). The *feasibility test* determines whether a given design specification performs feasibly over known ranges of uncertain parameters. The *flexibility index* problem seeks the maximum deviation of parameter from the nominal conditions the design can endure until it becomes infeasible (Swaney & Grossmann, 1985; Grossmann & Floudas, 1986). Several numerical approaches have been developed for solving the flexibility index problem. Swaney and Grossmann (1985) suggested a branch and bound algorithm which requires the assumption of critical points lying at a vertex. Grossmann and Floudas (1986) developed an active constraint set method to formulate the flexibility index problem as an MINLP. With few exceptions such as the α BB deterministic global

optimization by Floudas et al. (2001) or Lucia et al. (2005), previous flexibility solution efforts appears to only deploy local optimization techniques. In contrast to other design optimization problems in which locally optimal solutions may suffice, the solution to the flexibility index problem must be global to be rigorous. Therefore, local optimization is inadequate for flexibility analysis and global solution is a must.

Another important characteristic of the flexibility index problem is its inherent discontinuity and non-differentiability. The flexibility search space violates basic assumptions required by many deterministic local optimizations. The uncertain 2D space (θ_1 and θ_2) shown in Figure 1 illustrates this characteristic of the flexibility index. Despite continuity and differentiability of all constraints, the feasible deviation index, δ , plotted for rays emanating from the nominal conditions with angle α displays discontinuous and non-differentiable functional form. This non-differentiable δ space is problematic for gradient-based search methods.

In light of these challenges, we propose a novel *hybrid algorithm* for flexibility analysis. This algorithm will deploy a stochastic element to search the design space globally, while using local gradient information to explore a reduced state variable space. The novel *nearest constraint projection technique* is used to handle design constraints. It will also permit analysis of problems in which constraint derivatives are unavailable, such as large process models of legacy code or built with commercial black-box flowsheet simulators. It may also be useful in situations in which deterministic global algorithms are unsuitable. The proposed method also lends itself readily to parallelization, so that dramatic overall computational time reductions are possible.

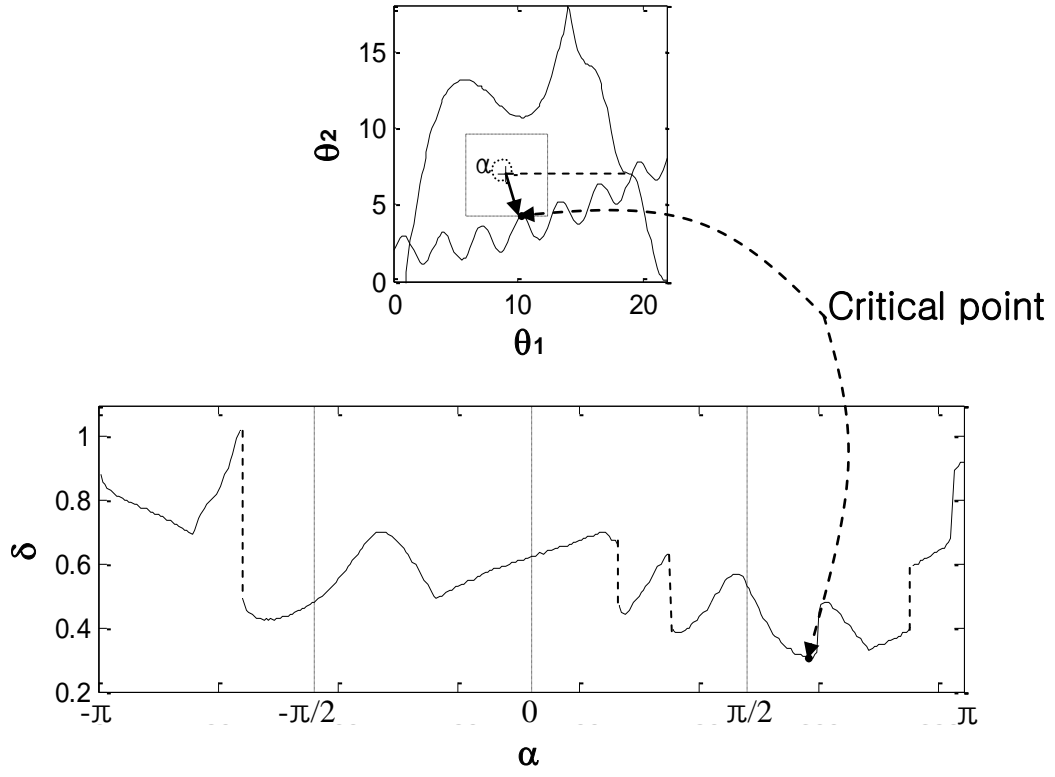


Figure 1: Depiction of the discontinuity and non-differentiability of the flexibility index problem. This graph shows the feasible deviation index, δ , plotted for rays emanating from the nominal conditions with angle α . This non-differentiable δ space is problematic for gradient-based search methods.

2 Methodology

2.1 Mathematical formulation of flexibility index problem

The scalar flexibility index, F , is defined as the largest normalized deviation from nominal conditions a design can tolerate while still ensuring feasible operation. Flexible process design aims at choosing design variables and controls so that the flexibility of the design is largest. For a design to be called flexible, it should at least more than unity which means that all expected deviation can be handled without violating any design constraints. At least more than unity which means that all expected deviation can be expected without violating any design constraints. The flexibility index, F , is given in system (1).

$$\begin{aligned}
 F &= \min \delta \\
 \text{s.t. } &\psi(\mathbf{d}, \boldsymbol{\theta}) = 0 \\
 &\psi(\mathbf{d}, \boldsymbol{\theta}) = \min_{\mathbf{z}} u, \\
 \text{s.t. } &h_i(\mathbf{d}, \mathbf{z}, \mathbf{x}, \boldsymbol{\theta}) = 0, i \in I \\
 &g_j(\mathbf{d}, \mathbf{z}, \mathbf{x}, \boldsymbol{\theta}) \leq u, j \in J
 \end{aligned} \tag{1}$$

where \mathbf{d} corresponds to the vector of design variables, \mathbf{x} is the vector of state variables, \mathbf{z} is the vector of controls, and $\boldsymbol{\theta}$ is the vector of uncertain parameters. The equality constraints, h_i , enforce mass and energy balances and equilibrium relations; inequality constraints, g_j , typically represents operational and design conditions. The scalar δ measures the *infinity norm* of the parameter deviation in the normalized uncertain space, as defined in equation (2).

$$\delta = \max_j \left(\frac{|\theta_j - \theta_j^N|}{\Delta \theta_j} \right) = \left\| \frac{|\theta_j - \theta_j^N|}{\Delta \theta_j} \right\|_{\infty}, j = 1, \dots, p \tag{2}$$

where, p is the number of uncertain parameters, θ_j^N and $\Delta \theta_j$, refers to the nominal value and the expected deviation of the uncertain parameter, j . The function $\psi(\mathbf{d}, \boldsymbol{\theta}) = 0$ defines

the feasible region boundary in the space of uncertain parameters, as shown in Figure 2 (Swaney & Grossmann, 1985).

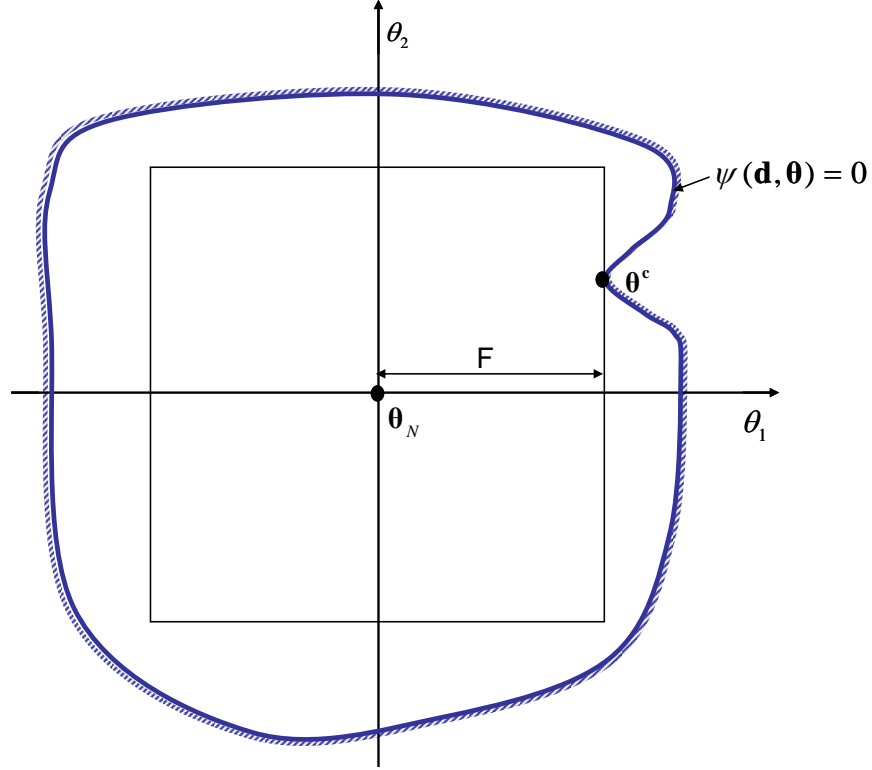


Figure 2: Graphical representation of the feasible region boundary, $\psi(\mathbf{d}, \boldsymbol{\theta}) = 0$, and the critical point, $\boldsymbol{\theta}^c$. Geometrically, the flexibility index F , corresponds to the dimension of the hyper rectangle touching the feasible region boundary at the critical point.

The flexibility index problem aims at identifying the critical point, $\boldsymbol{\theta}^c$, corresponding to the worst case scenario on the feasible region boundary closest to the nominal conditions. Geometrically, the flexibility index, corresponds to the dimension of the hyper-rectangle just touching the feasible region boundary at the critical point. Unfortunately, the feasibility function $\psi(\mathbf{d}, \boldsymbol{\theta})$ is usually not explicitly given, so that finding globally the closest critical point is not a simple task. Moreover, solving for critical points of the feasibility function locally in one sector of the uncertain space, does not teach anything about the flexibility in another region. Hence, the $\psi(\mathbf{d}, \boldsymbol{\theta})$ surface needs to be *searched globally*. The next

subsection will introduce a hybrid algorithm to identify the critical point and the flexibility index globally.

2.2 Hybrid algorithm

We have chosen a *hybrid* method combining the robustness of stochastic global search with the superior performance of Newton-type deterministic techniques. The main idea of the hybrid algorithm is to generate samples of uncertain parameter realizations located precisely on the feasible region boundary. To accomplish this task we use a genetic search based on the principles of natural selection and inheritance (Holland, 1975; Goldberg, 1989; De Jong, 1975). Candidate solutions generated by the stochastic search (Deb, K. 2000; Kim & Myung, 1997) are then projected onto the boundary of the feasible region while rigorously satisfying all state equations. This *nearest constraint projection technique* ensures that all candidate solutions are placed exactly on the flexibility boundary. To satisfy equality constraints, we adopt gradient-based methods such as quasi-Newton methods. Since all candidate solutions of the evolving population are members of the feasible region boundary, the sample with the minimum infinity norm is the critical point. Thus, the proposed hybrid algorithm exhibits both stochastic and deterministic features.

The information flow of the hybrid algorithm is depicted in Figure 3. In the first step, an initial random sample population is generated in the space of the uncertain parameters, θ . In the next step, each crude sample is projected onto the feasible region boundary, $\psi(\mathbf{d}, \theta) = 0$. The details of this nearest projection technique will be described in section 3. This step also deploys Newton-type methods to satisfy equality constraints corresponding to the reduced state space. The ‘fitness’ of each projected candidate solutions is inversely proportional to the infinity norm, δ . In the natural selection step, samples with superior

‘fitness’ scores are chosen to produce competitive offspring for the next generation. Candidate solutions with low fitness are unlikely to reproduce and tend to disappear. Offspring, θ_{child} , are computed by combining the parameter values of the selected parents, θ_1, θ_2 according to the arithmetic crossover formula with one random variable, α , given in equation (3).

$$\theta_{child} = (1 - \alpha)\theta_1 + \alpha\theta_2, \quad 0 \leq \alpha \leq 1 \quad (3)$$

Mutations occurring with given likelihood change the states of a single candidate solution according to a random event drawn from the probability density function given in equation (4).

$$\theta'_n = \theta_n + \sigma N_n(0,1) \quad (4)$$

Here, σ is the standard deviation of the normal distribution $N_n(0,1)$. Mutations are needed to counter clustering of the candidate solutions in the search space and helps explore the entire search space uniformly. The overall algorithm terminates when the specified generation limit is reached. The critical point is the candidate solution with the highest fitness. Since all candidate solutions lie on the feasible boundary, the minimal δ -deviation marks the critical point.

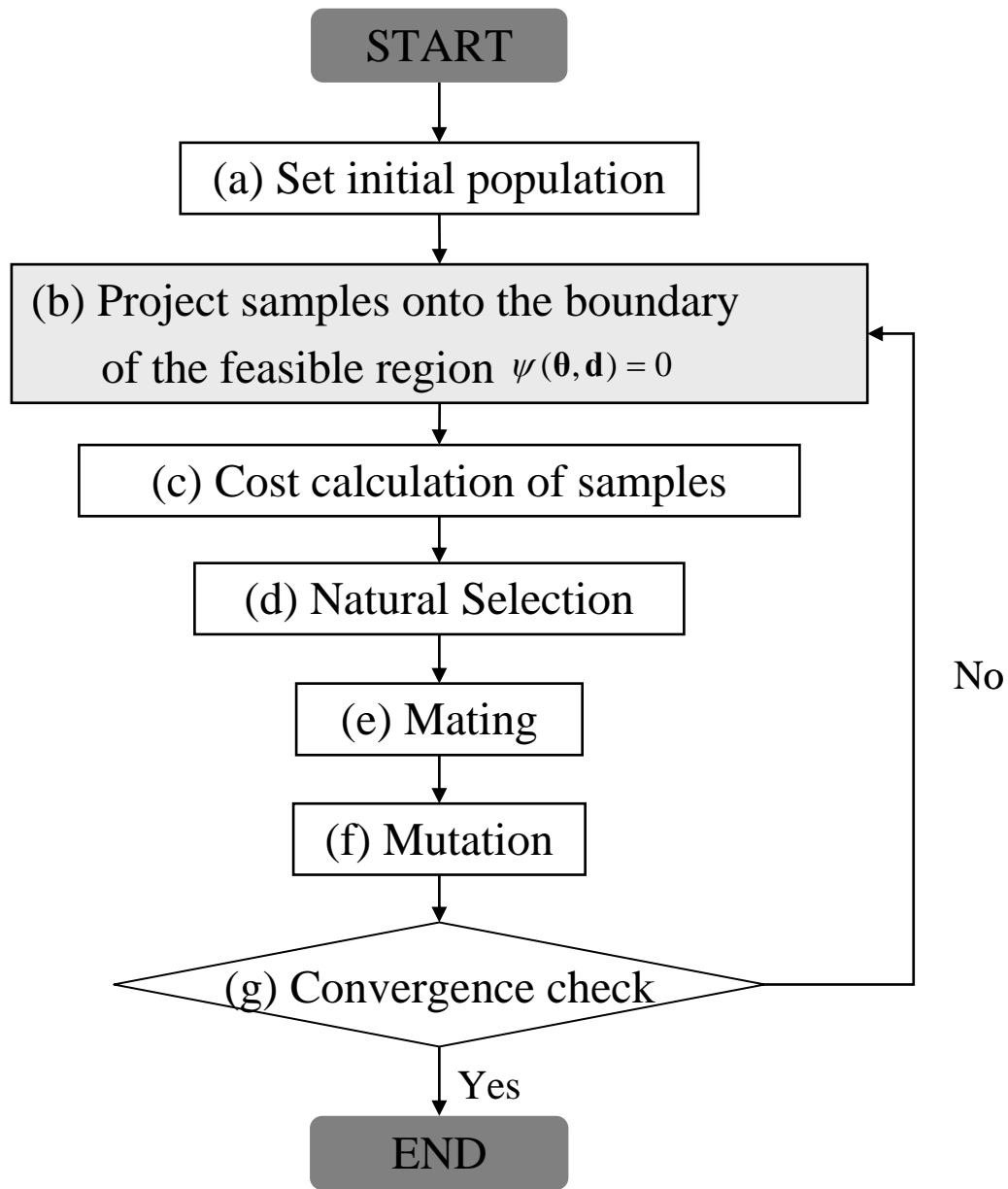


Figure 3: Flowchart of the proposed hybrid algorithm.

3 Nearest constraint projection technique

Our hybrid algorithm needs to maintain an evolving population of candidate solutions located precisely at the feasibility boundary. The crude samples generated by genetic inheritance do not lie of the flexibility boundary in general as schematically depicted in Figure 4. Therefore, we project the parameter values of the crude samples onto the feasibility boundary with exact methods. This correction process known as *repair* procedure is implemented efficiently as a one-dimensional directional search (Garey & Johnson, 1974; Press et al., 2007; Michalewicz, 1994). The search direction is selected to be collinear to the direction of the uncertain parameter dimension, θ^* , corresponding to the specific coordinate marking the infinity norm. The required parameter corrections are implemented iteratively to ensure precise location of the feasibility boundary, while simultaneously satisfying all state equations. Nevertheless, a *repair procedure* in a single direction is merely a one-dimensional search requiring but small computational effort. The projection is similar to the well-known *line search strategies* deployed in step-size controlled Newton methods.

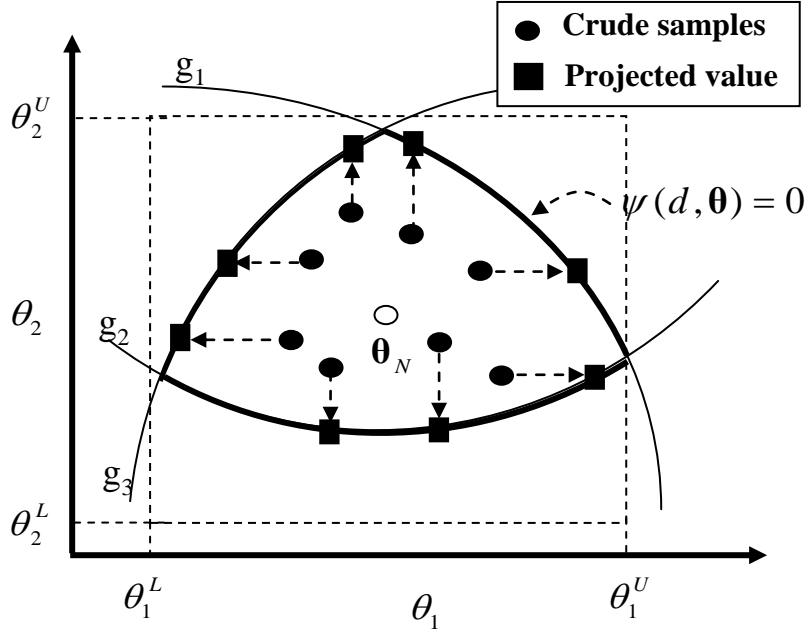


Figure 4: A schematic depiction of crude samples and the one-dimensional projection to the feasible boundary, $\psi(\mathbf{d}, \boldsymbol{\theta}) = 0$. The crude samples generated by genetic inheritance do not lie of the flexibility boundary in general. One-dimensional projection brings the samples to the feasibility boundary.

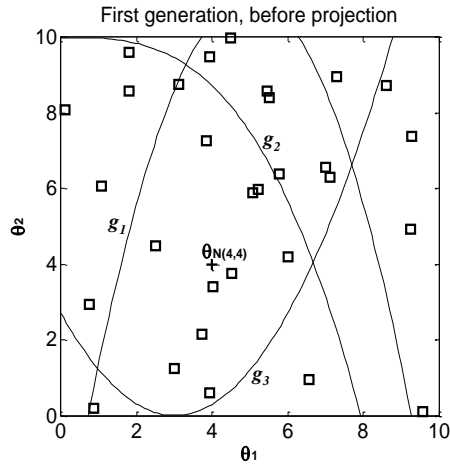
3.1 Illustration of a nonlinear two dimensional flexibility problem

The evolution of the hybrid mechanism is illustrated graphically with the help of an example with two uncertain parameters in (5).

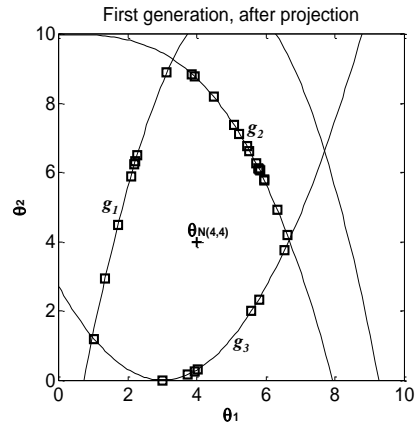
$$\begin{aligned}
F &= \min \delta \\
s.t. \quad &g_1(\boldsymbol{\theta}) \text{ or } g_2(\boldsymbol{\theta}) \text{ or } g_3(\boldsymbol{\theta}) = 0 \\
&g_1 = 30(\theta_1 - 5)^2 - 550 + 50\theta_2 \leq 0 \\
&g_2 = \theta_1^3 - 500 + 50\theta_2 \leq 0 \\
&g_3 = 15(\theta_1 - 3)^2 - 50\theta_2 \leq 0 \\
&\boldsymbol{\theta}^N - \delta\Delta\boldsymbol{\theta}^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^N + \delta\Delta\boldsymbol{\theta}^+ \\
&\Delta\theta_1, \Delta\theta_2 = \pm 2 \\
&\boldsymbol{\theta}^N = (4, 4)
\end{aligned} \tag{5}$$

Figure 5 shows snapshots of samples belonging to successive generations leading to the global solution.

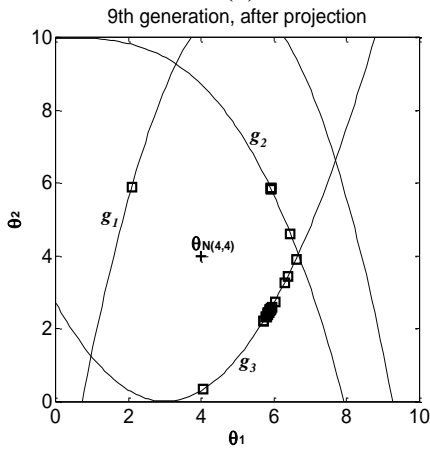
As the evolution progresses, each sample is projected onto the feasibility boundary delineated by the nearest constraint. We have observed that computational effort for bringing the crude samples to the feasibility boundary diminishes as the solutions draw closer to a critical region. After 12 generations, the candidate solutions coalesce around to the critical point (5.7429, 2.2571). This example was solved repeatedly with different nominal points in each run. Table 1 and Figure 6 summarize the results and the performance parameters. We found that the flexibility index was correctly detected in all instances within reasonable CPU time.



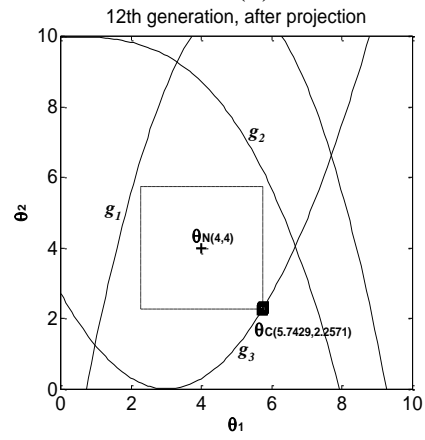
(a)



(b)



(c)



(d)

Figure 5: Snapshot of for flexibility index problem. Samples coalesce to the critical point (5.7429, 2.2571).

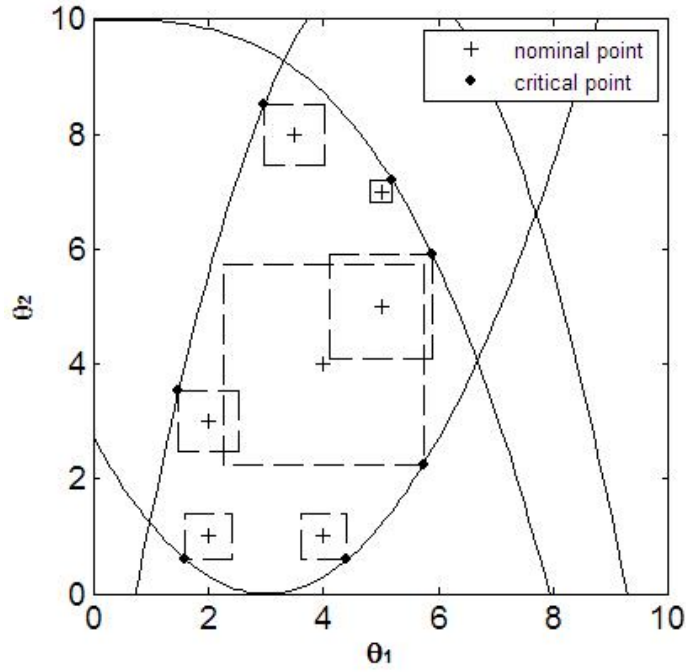


Figure 6: Critical points and independent operational regions corresponding to different nominal points in the uncertain parameter space of example. The proposed method correctly identifies the size of hyper-rectangle in each situation.

Table 1: Critical points and flexibility indices corresponding to different nominal points

Nominal point	Critical point found	Flexibility index	CPU Time(msec)
5,5	5.8975,5.8975	0.4488	203
2,1	1.5935,0.5935	0.2033	204
4,4	5.7429,2.2571	0.8715	201
3.5,8	2.9707,8.5293	0.2646	198
2,3	1.4712,3.5286	0.2644	204
4,1	4.4065,0.5935	0.2033	200
5,7	5.1954,7.1954	0.0977	203

3.2 Nearest constraint projection with controls

The flexibility of a design problem can be drastically enhanced with adjustable controls, z . To better understand the impact of controls on the feasibility function, ψ , consider the example in (6) visualized in Figure 7 with two uncertain parameters, θ_1 and θ_2 , and one control, z .

$$\begin{aligned} g_1 &= (\theta_1 - z)^2 + \theta_2^2 - 1 \leq 0 \\ g_2 &= z - 1 \leq 0 \\ g_3 &= -z \leq 0 \end{aligned} \tag{6}$$

For a fixed z value, the feasible region, $\psi(\mathbf{d}, \boldsymbol{\theta}) \leq 0$, is a circle in the θ space; the feasible region changes for different realizations of the control levels, z , as depicted in Figure 7a. The feasible region $\psi(\mathbf{d}, \boldsymbol{\theta}) \leq 0$ is the unified gray region made possible by permissible control adjustments z . Thus, the enlarged area projected onto the θ space for all permissible z levels is the feasible region shown in Figure 7b. In general, an explicit expression for the feasible region boundary with control, $\psi(\mathbf{d}, \boldsymbol{\theta}) = 0$, is not available. Hence, we augment the *nearest constraint projection technique* to move infeasible samples to the feasible search space with additional control adjustments. This technique will be shown to explore both the space of uncertain parameters, $\boldsymbol{\theta}$, as well as the controls, z .

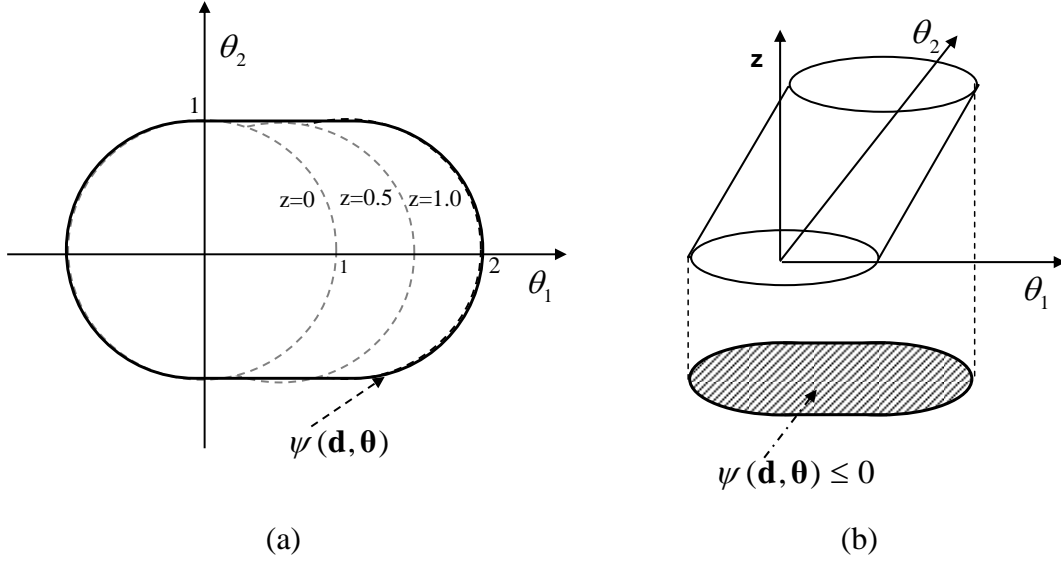


Figure 7: Geometric representation of the feasible region, $\psi(\mathbf{d}, \boldsymbol{\theta}) \leq 0$, in the presence of control, (a) depicts constraint variation due to changing z values in the $\boldsymbol{\theta}$ space. (b) shows the projection of the z space onto the $\boldsymbol{\theta}$ space to obtain, $\psi(\mathbf{d}, \boldsymbol{\theta}) \leq 0$. The feasible region is the unified gray region made possible by permissible control adjustments z .

With control degrees of freedom, projection onto the feasibility boundary becomes more challenging, because control especially with feedback adds arbitrary nonlinearity and complexity to the design constraints. Rather than insisting on rigorous projection, we propose an opportunistic zig-zag search which is easy to implement and computationally inexpensive. Since the projection on the feasibility boundary is performed for each crude sample of the population, the zig-zag control projection need not be correct in every instance. As long as one sample of the genetic search yields the flexibility expansion achievable with adaptable control, the overall algorithm will proceed to the global solution from generation to generation in a statistical sense. Hence, projection with zig-zag search needs to succeed merely in a statistical sense true to the stochastic nature of the proposed methodology.

3.3 Implementation of the nearest constraint projection technique

This section introduces the detailed implementation of the nearest constraint projection technique with *zig-zag* search. The example of Figure 8 illustrates the essential steps of the method for a problem with one uncertain parameter and one control. It graphically represents the simultaneous movement in the space of the uncertain parameter, θ , and control, z . For a fixed control level z , the thick line delineates the range that the parameter θ can assume without violating design constraints. It is the objective of the projection technique to identify the largest possible θ extension achievable by suitable control adjustments within permissible range, z_{\min} and z_{\max} .

In the example, the maximum tolerable θ extension is marked by P_5 . For a given starting value $P_1(\theta, z)$, the algorithm searches in the θ -direction outward from the nominal condition until it hits the feasibility boundary at P_2 . When there are no controls, this first step completes the projection correction. With control, repeated adjustments to the control, z , and search direction, θ , are necessary until the furthest point P_5 belonging to the feasibility boundary is found. A detailed examination of the *zig-zag* search follows:

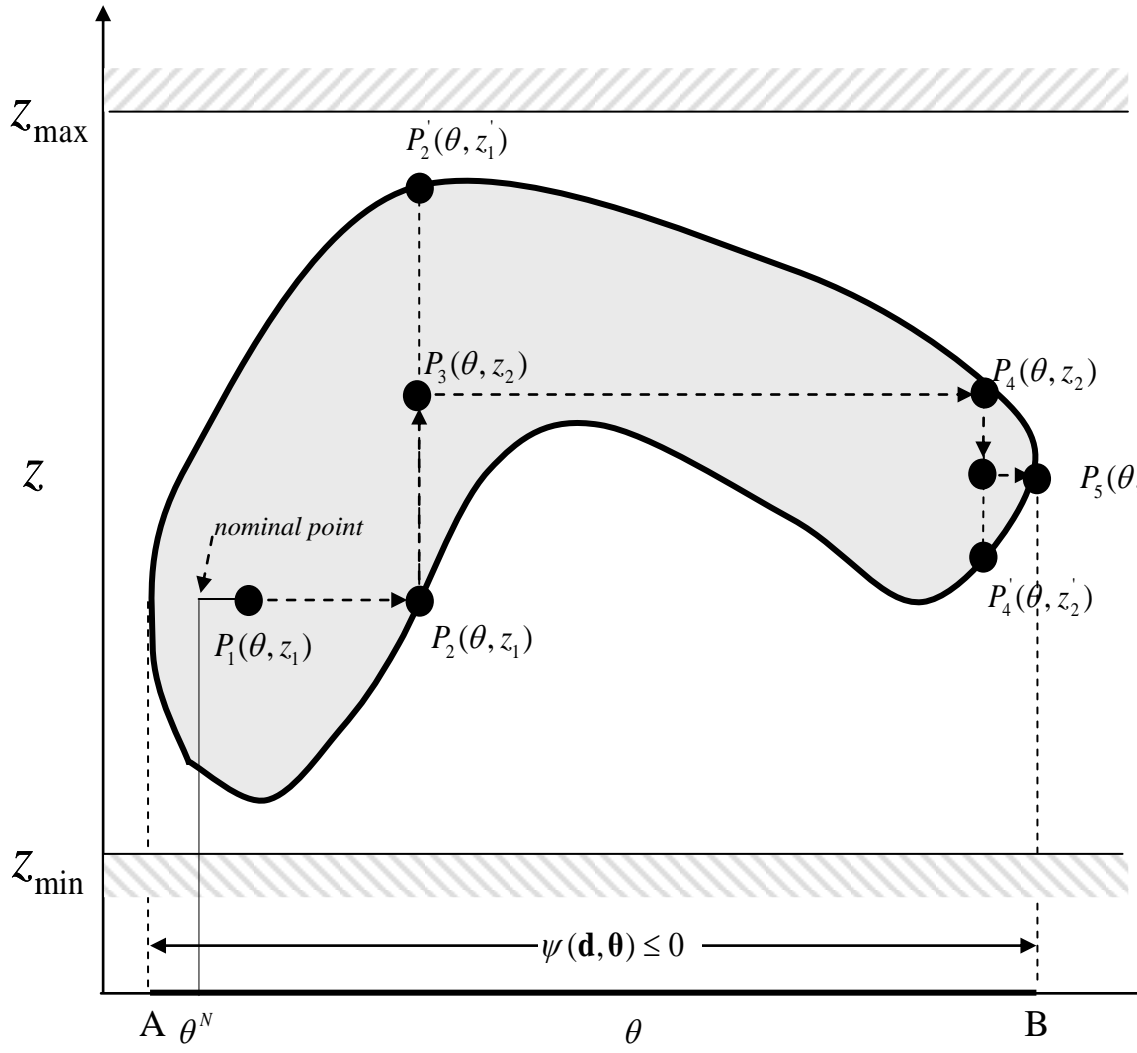


Figure 8: Graphical representation of the *nearest constraint projection technique*.

Step 1. *Select the initial search direction θ^* .* First, the projection direction is decided to correspond to the specific component of the parameter vector, θ , with the maximum scaled deviation from the nominal point. The maximum scaled deviation equals the infinity norm given in equation (2).

Step 2. *Determine the orientation ω .* Once the critical component and its direction, θ^* , has been found, the orientation of movement needs to be determined. The orientation parameter ω can be computed as in (7) for the one-dimensional move to lead away from the nominal point:

$$\left. \begin{array}{l} \omega = 1, \text{ if } \theta^N < \theta^* \\ \omega = -1, \text{ if } \theta^N > \theta^* \end{array} \right\} \forall \theta \in \Theta \text{ where } \Theta = \{\theta \mid \psi(\mathbf{d}, \theta) < 0\} \quad (7)$$

Step 3. *Line search in θ^* -direction -direction.* Line search starts from P_1 in the direction of increasing θ , for a fixed control level, $z=z_1$, until it meets the constraint boundary at P_2 .

$$\theta_{i+1}^* = \theta_i^* + \omega \Delta_{step} \quad \Delta_{step} = 0.01 \times \Delta \theta^* \quad (8)$$

Here, Δ_{step} , is a sufficiently small positive step size, typically 1% of the interval range $\Delta \theta^*$. Additional safeguards maintain adjustments within the upper and lower bound, θ^u and θ^L . If the flexibility problem has no adjustable controls, the projection method terminates at this point. When adjustable controls are available, steps 4-6 for the *zig-zag* search need to be traversed.

Step 4: *Control Adjustment.* At the current position P_2 , further movement in the positive θ^* direction causes infeasibility. However, with suitable control action a higher θ -level might be reachable. The algorithm performs line search in the positive z direction

until a constraint boundary is met at P_2' for $z = z_1'$, or the maximum control range z_{\max} is encountered.

Step 5: Expanding the feasibility range. To explore whether increased control action does expand the feasible region, the θ space needs to be explored for different control levels. We propose to choose a new control level, z_2 , given as the average of z_1 and z_1' .

$$z_2 = \frac{z_1 + z_1'}{2} \quad (9)$$

Step 6: Test new control level. At the newly chosen control level, z_2 , we proceed repeatedly with steps 3, 4 and 5, until no more control movement is possible without causing a constraint violation. For deciding the correct z -direction, there are three possibilities as depicted in Figure 9. In type-1, the candidate solution is located on the lower boundary; thus increasing the current control level expands the feasible region as shown in Figure 9a. In type-2, the candidate solution hits the upper boundary; thus decreasing current control level augments the feasible region as shown in Figure 9b. Candidate solution positions falls on an extreme point so that an increase or decrease of the current control level would make the sample infeasible. This case is shown in Figure 9c. The numerical test in (10) determines the correct move in the z -direction for the three types by examining the feasibility of two adjacent points: $G^+(\theta, z_k + h, \mathbf{d})$, $G^-(\theta, z_k - h, \mathbf{d})$, where h is a small scalar.

$$\begin{aligned} G^+ &= \max_j g_j(\theta, z_k + h, \mathbf{d}) \\ G^- &= \max_j g_j(\theta, z_k - h, \mathbf{d}) \end{aligned} \quad (10)$$

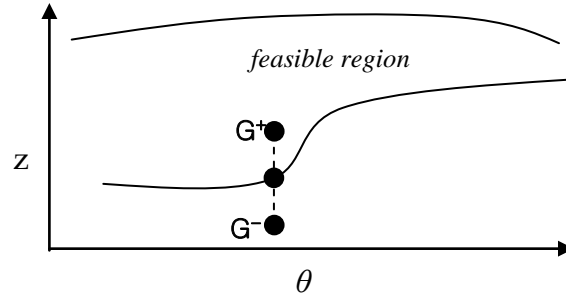
If $G^+ < 0$ and $G^- > 0$, this means the current point is of type 1. Thus, the control level, z , must be increased (type-1). If $G^+ > 0$ and $G^- < 0$, then the control level, z , must be decreased (type-2). When both G^+ and G^- are infeasible, this means the current point

is the candidate critical point which does not allow further movement in the θ space (type-3).

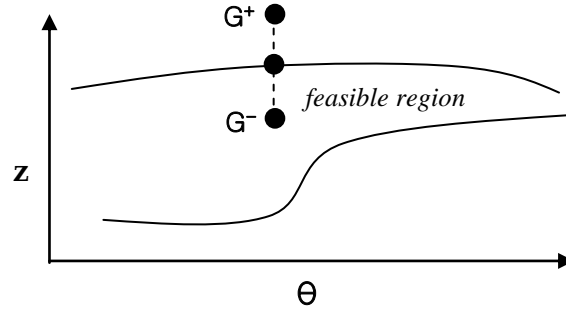
The final point encountered is a member of the feasibility boundary, $\psi(\mathbf{d}, \theta) = 0$. Thus starting from the initial point, P_1 , the algorithm performs *zig-zag* movements in θ and z space as depicted in Figure 8.

Projection with open boundaries. It should be pointed out that the proposed projection technique is also applicable for the case of open constraint boundaries, the bounds of parameter ranges are enforced in steps 3. Thus, when there are no design constraints limiting the expansion, the algorithm automatically stops at the upper or lower bound of the permissible uncertain parameter realizations, θ_i^u and θ_i^l .

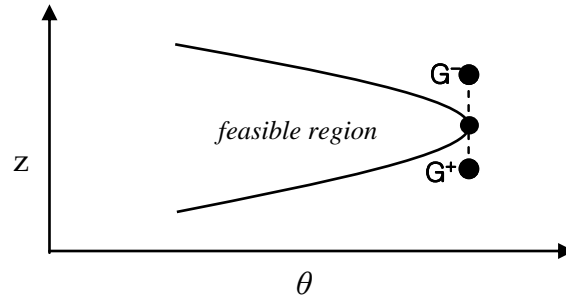
Feasibility function evaluation and state variable solving. We need to ensure that the *projection process* produces corrected samples that satisfy the state equations. Hence, the uncertain parameter θ and states variable \mathbf{x} are updated in accordance with the help of Newton-type methods. We used Newton method and Quasi Newton method (Press et al., 2007). Both methods used numerically approximated Jacobian matrices as starting points. Thus, the overall algorithm has a hybrid structure due to a genetic search in the parameter space combined with a gradient-based solution of the equality constraint set in the reduced state space. This combination harnesses the robustness of the stochastic genetic method as well as the efficiency of the gradient-based equation solving methods. In addition, it saves computational time by avoiding the solution of the full state space with genetic search.



(a)



(b)



(c)

Figure 9: Three possible types for deciding correct direction of control moves. (a) For a type 1, $G^+ < 0$, $G^- > 0$. (b) For a type 2, $G^+ > 0$, $G^- < 0$. (c) For a type 3, $G^+ > 0$, $G^- > 0$.

3.4 Multiple controls

The discussion of the projection technique presented in previous section considered only one control. For flexibility analysis with multiple controls, all possible controls should be considered for expanding flexibility. Projection with one control leads to the local extreme point (this is called *local termination*). We continue the *zig-zag* search for other controls until we reach a point which cannot be extended in spite of changing controls. Unfortunately, an increase in the number of controls leads to a combinatorial explosion. Hence the proposed algorithm is suitable for complex nonlinear problems with only modest number of adjustable controls.

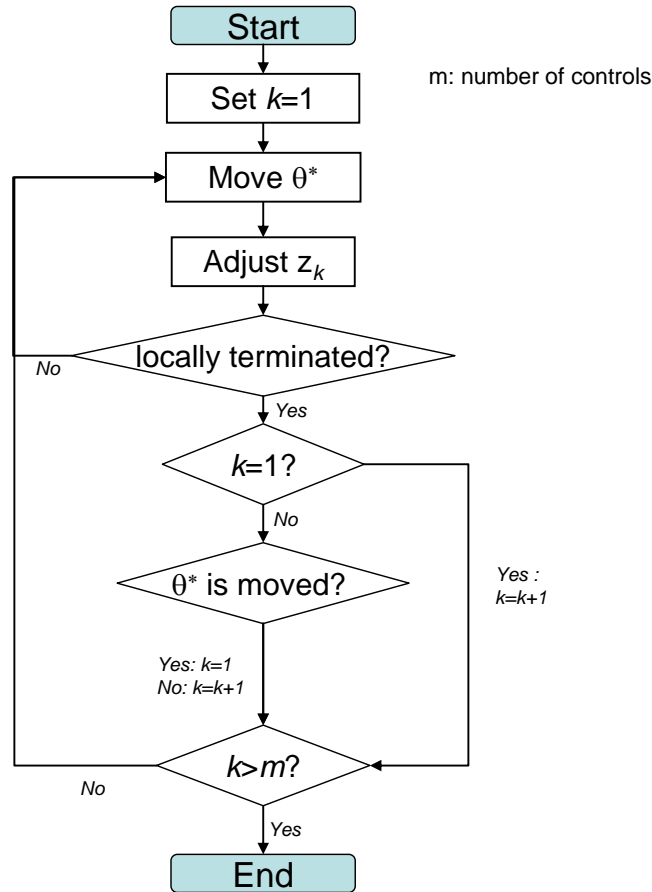


Figure 10: Flowchart of the nearest constraint projection technique with multiple controls. Projection with one control leads to the local extreme point (*local termination*). We continue the *zig-zag* search for other controls until we reach a point which cannot be moved in spite of changing controls.

In order to better illustrate the projection technique for multiple controls, consider the following simple example that has one uncertain variable and two controls as depicted in Figure 11. This example traverses each step of the flow chart presented in Figure 10. Note again that the *zig-zag* search need not be rigorous in each run, as long as one sample succeeds in identifying the largest range, the conditions of a convergence of the ensemble in a statistical sense would be satisfied.

$$\begin{aligned} g_1 &= \theta^2 + z_1^2 + z_2^2 - 1 \leq 0 \\ \theta^N &= 0, \Delta\theta = \pm 1 \end{aligned} \tag{11}$$

Let us assume that the starting point is $P_1(\theta = 0.1, z_1 = 0.4, z_2 = 0.5)$.

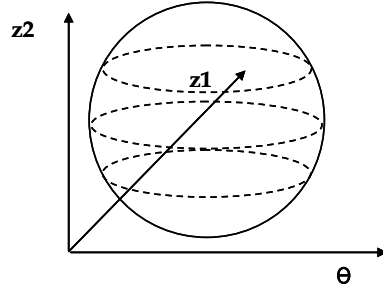
Step 1: *Projection in (θ, z_1) space:* This step involves projection in the space of θ and z_1 for a fixed value of $z_2 = 0.5$. Thus, starting from P_1 , the proper projection point $P_2(\theta = 0.75, z_1 = 0.0, z_2 = 0.5)$ is obtained by following the projection steps for a single control as shown in Figure 11b.

Step 2: *Projection in (θ, z_2) space:* This step involves projection in the space of θ and z_2 starting from $P_2(\theta = 0.75, z_1 = 0.0, z_2 = 0.5)$ as depicted in Figure 11c. The outcome of this search is $P_3(\theta = 1.0, z_1 = 0.0, z_2 = 0.0)$ and corresponds to local termination at this step.

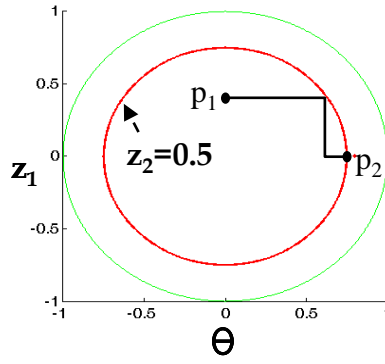
Step 3: *Projection in (θ, z_1) space:* Starting from $P_3(\theta = 1.0, z_1 = 0.0, z_2 = 0.0)$ obtained in step 2, we explore the (θ, z_1) space again. In this case, P_3 cannot be moved by changing z_1 because it is a point of type 3 in the (θ, z_1) space. So, this step locally terminates at $P_4 = P_3$.

Step 4: *Projection in (θ, z_2) space:* Starting from P_4 obtained in step 3, we explore the (θ, z_2) space again. Here, P_4 cannot be moved by changing z_2 , because it is a point of type 3 in the (θ, z_2) space. So, this step locally terminates at $P_5=P_4$.

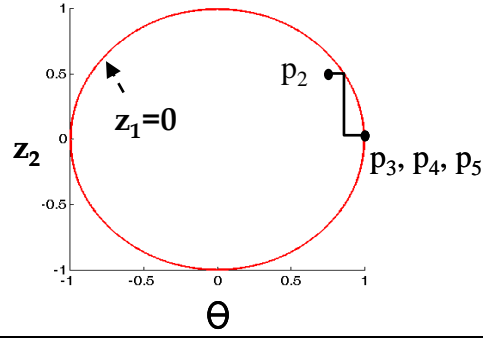
Step 5: *Checking global termination:* Since the projection technique exploring the space of both controls, z_1 and z_2 , independently yield the termination points $P_3=P_4=P_5$ ($\theta = 1.0, z_1 = 0.0, z_2 = 0.0$), we conclude that the algorithm globally terminates at this point.



(a)



(b)



(c)

Figure 11: Graphical illustration of the example for the example in section 3.4. (a) This example has one parameter, θ , and two controls, z_1 and z_2 . (b) In step 1, projection in (θ, z_1) space yields the local termination point P_2 . (c) In step 2, projection in (θ, z_2) space yields the local termination point P_3 , which is also the global termination point P_4, P_5 .

3.5 Overcoming problems with disjoint or complicated flexibility regions

Non-linearity introduced commonly by feedback control may create highly complex non-convex feasibility regions. In extreme cases, the feasibility space may even be divided into disjoint regions. We obtained good results in challenging case by reinitializing the projection method from different control levels as follows. We create multiple starting samples at different control level. Then, we follow step 1-5 for each starting point. The sample reaching the furthest distance from the nominal point holds the desired δ -value. The following example demonstrates the re-initialization process with multiple control starting points for the problem in (12).

$$\begin{aligned} g_1 &= z \sin z + 6.5 - \theta \leq 0 \\ g_2 &= -\sin 2z + 1 - \theta \leq 0 \end{aligned} \tag{12}$$

The search space for this problem is not convex as shown in Figure 12. Moreover, the feasible region falls into two separate not connected regions. In this example, the sample $\theta=1$ is projected onto the region $\psi(\mathbf{d}, \theta) = 0$. This projection starts with the control level $z=1$ (P_1), $z=2$ (P_2) and $z=7$ (P_3). Projection begins from a point like $P_1(1,1)$ and terminates at $P_1'(8.3, 2.0)$. This is not yet the most flexible point achievable with adjustable control and does therefore not constitute a point on the boundary of the feasible region, $\psi(\mathbf{d}, \theta) = 0$. When proceeding from the point like $P_3(1, 7)$, the solution $P_3'(14.4, 7.9)$ is found, which does belong to the feasibility boundary. For completeness, there may be some points like $P_2(1, 2)$ whose control and uncertain parameter pairing already violate constraints. In the current implementation, we abandon such samples without further exploration. In summary, multiple control level starting points are necessary to find candidate solutions on the feasible boundary in problems with complex constraints or

disjoint feasibility spaces. Large population sizes ensure dense exploration of the uncertain parameter and control variable space with high probability.

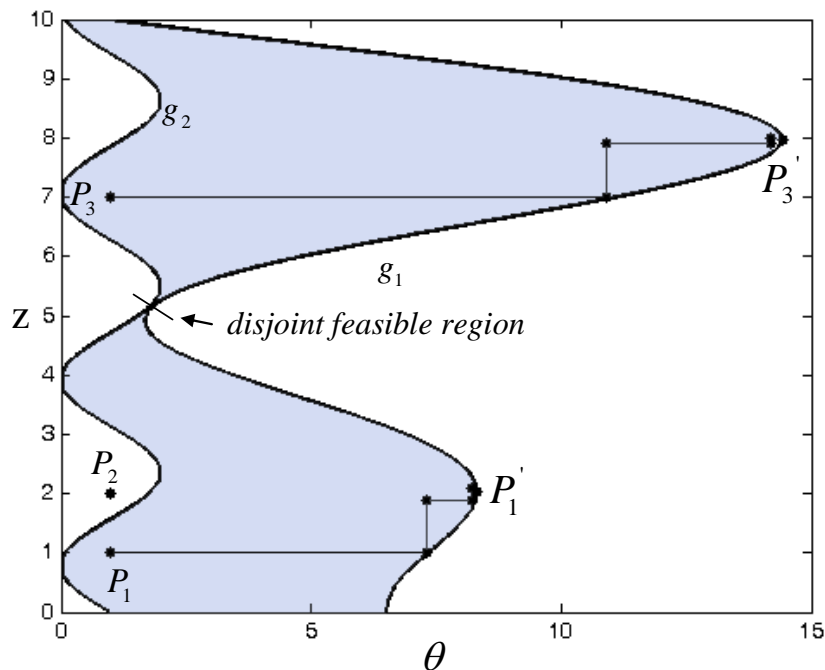


Figure 12: Projection with multiple control level in the disjoint feasible region. To efficiently explore the feasible region boundary, multiple control level starting points are necessary to find candidate solutions on the feasible boundary in problems with complex constraints or disjoint feasibility spaces.

3.6 Implementation

Problem formulation. A final comment explains the ease in formulating the flexibility analysis with the novel hybrid algorithm. We can directly use legacy code representing state equations and design specification inequalities. In contrast to deterministic methods, there is no need for algebraic manipulations to transform the design equations into a *min-max-min* flexibility problem, introduce artificial variables for active constraints, or provide derivative information of constraints or inequalities. The proposed algorithm is capable of computing the fitness function as well as conducting these projection methods with no user intervention. For the state space equality constraints, the

quasi-Newton approximates gradients with secant information without the need for analytical or numerical derivatives. Thus, the hybrid algorithm is suitable to conduct flexibility analysis on black box legacy code. The hybrid algorithm simply uses the embedded state equations to perform repeated function evaluation. This feature may be advantageous in industrial applications. The ease of setting up the mathematical problem formulation for global flexibility analysis is a plus of the proposed algorithm.

Choosing parameters. Like every numerical scheme, the hybrid method features several adjustable parameters. It is common knowledge that parameter setting criteria cannot guarantee solutions universally, and problem specific considerations are unavoidable. However, this limitation befalls any stochastic technique. Even deterministic methods require problem-specific adjustable parameters; their rigorous foundation often does not carry to a digital computer with finite precision. For these case studies we have adopted an empirical procedure to set initial population size, mutation rates, or number of generations to working reliably in all case studies. The population size should be at least 100 times the dimension of the uncertain space. In the case studies, the generation counter was 50. Population size can be traded against to the generation, so that smaller populations require more iteration. Typically, after two or three trial-and-error adjustments, applications converged reliably to the same solution independent of initial guesses or parameter choices. The mild inconvenience for tuning as described in these implementation guidelines may not constitute an unreasonable limitation of the proposed numerical method.

4 Applications

In this section, five applications demonstrate the performance of the hybrid algorithm. The first application critically compares the robustness and performance of our hybrid algorithm against a commercial MINLP. Additional applications demonstrate reliability and computational speed in classical test case studies from the literature. We also demonstrate the applicability of our hybrid method for a reactor-cooler system problem with three controls. Finally, an industrial size realistic polymerization reactor demonstrates substantial performance advancements realizable with parallel processing. All applications were implemented with on Intel core2 PC-CPU 2.4Ghz. Parallel computing examples were implemented on a decentralized local area network with up to ten PCs communicating with a simple TCP/IP inter-process protocol.

4.1 Application 1: Non-convex, nonlinear two dimensional problem

This application considers a simple two-dimensional non-convex problem with three non-convex inequality constraints including a periodically varying boundary. The flexibility problem is formulated as in system (13). The nonlinear constraints lead to critical points which do not necessarily lie on the vertex of the flexibility box.

$$\begin{aligned}
 F &= \min \delta \\
 \text{s.t. } & g_1(\boldsymbol{\theta}) \text{ or } g_2(\boldsymbol{\theta}) \text{ or } g_3(\boldsymbol{\theta}) = 0 \\
 & g_1 = 4\theta_2 + 9\theta_1 - 198 - 4\sin(2\theta_1) \leq 0 \\
 & g_2 = 20\theta_2 - (\theta_1 - 4)(\theta_1 - 8)(\theta_1 - 12) - 240 \leq 0 \\
 & g_3 = -\theta_2 + \sin(2\theta_1) + 2 + 0.0125\theta_1^2 \leq 0 \\
 & \boldsymbol{\theta}^N - \delta\Delta\boldsymbol{\theta}^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^N + \delta\Delta\boldsymbol{\theta}^+ \\
 & \Delta\theta_1, \Delta\theta_2 = \pm 3
 \end{aligned} \tag{13}$$

Multiple simulations experiments were conducted to test the new method with parameters: population size=30, maximum number of generations=30, selection ratio=0.5, and mutation ratio=0.01; and are summarized in Table 2.

Table 2: Critical points and flexibility indices corresponding to several nominal points for application 1. The super script * indicates the incorrect solution obtained with the MINLP method (TOMLAB).

θ^N	Our approach			Local MINLP (TOMLAB)		
	θ^C	FI	CPU Time (msec)	θ^C	FI	CPU Time (msec)
4,4	3.9499,3.1941	0.2686	218	3.9517,3.1940	0.2687	3,078
8,10	9.2873,11.0767	0.4291	202	9.3046,11.0674	0.4349	1,937
12,6	13.1388,5.0686	0.3796	196	13.1387,5.0684	0.3796	11,031
16,10	17.1950,10.9777	0.3983	234	17.1951,10.9778	0.3983	2,765
15,8	16.5772,6.4212	0.5263	207	17.4279,9.9876	0.8093*	3,328
15,12	16.5462,13.2651	0.5154	189	16.5463,13.2652	0.5154	3,828
9,7	10.2778,4.3115	0.8962	210	12.6009,4.0538	1.2003*	2,896

Figure 13a illustrates the outcomes of each flexibility index problems according to different nominal point. For each nominal point, the critical scenarios with its rectangular feasibility boxes are depicted. We compared the hybrid algorithm results with an active constraint method implemented with a TOMLAB dense branch and bound MINLP method with starting values initialized at the nominal point, θ^N . The MINLP algorithm failed in finding the correct solution in two cases. The details of Figure 13b shows that the MINLP methods broke down when the critical point does not lie on vertices as is the case for the flexibility problem for nominal points, $\theta^N(15, 8)$ and $\theta^N(9, 7)$. These results underscore how some off-the-shelf MINLP methods may be inadequate for rigorous flexibility analysis, and global algorithms have to be deployed.(Tawarmalani & Sahinidis, 2004; Androulakis et al., 1995). In addition, the MINLP solver also exceeded 10-15 times the CPU requirement of our hybrid method.

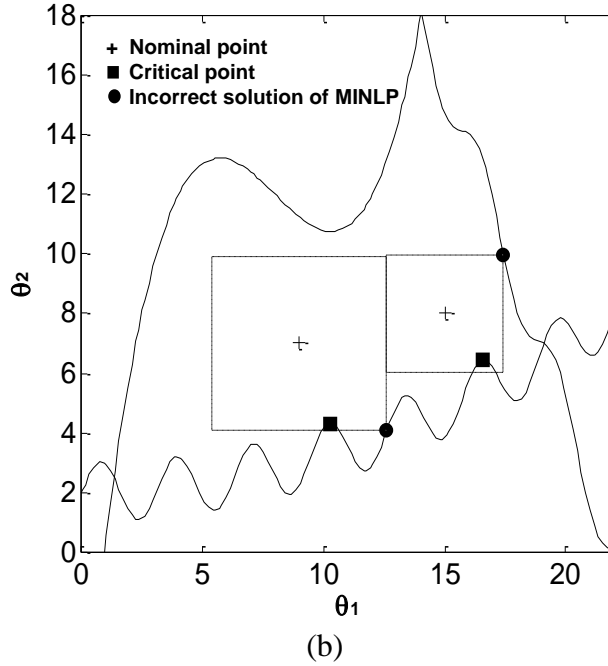
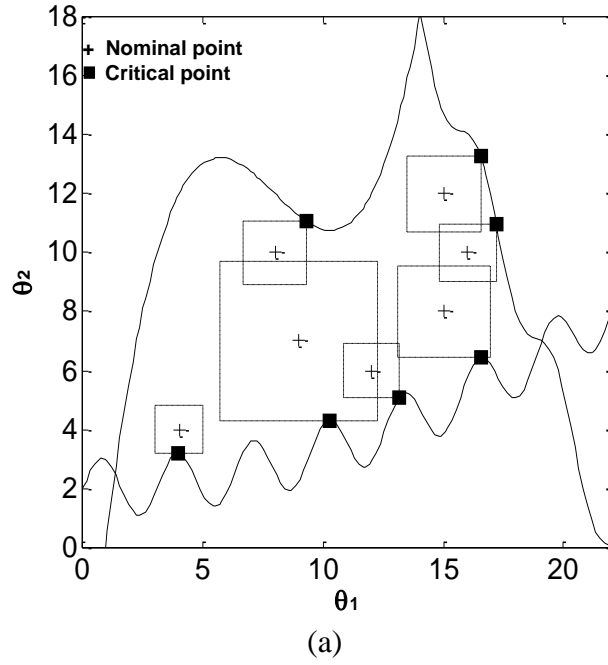


Figure 13: The results of application 1; (a) Seven nominal points and corresponding critical points found by our method. (b) The MINLP method failed when critical point did not lie on the vertex. Our method finds exact solutions even though the constraints are non-convex. The MINLP solver also required about 10-15 times the CPU times of our hybrid method.

4.2 Application 2: Heat exchanger network with varying heating duties

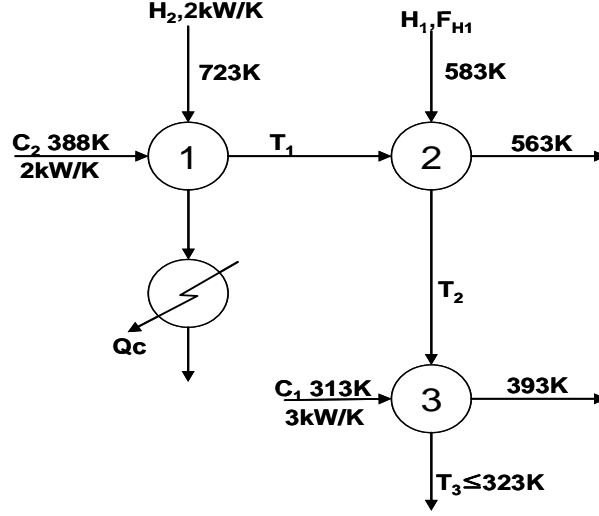


Figure 14: Heat exchanger network reproduced from Grossmann et al. (1986). F_{H1} is the uncertain parameter and Q_c is the control variable.

This application concerns the flexibility of the heat exchanger network shown in Figure 14 and initially used as a benchmark by Grossmann and Floudas (1986) with parameters given in Table 3.

Table 3: Heat exchanger variables, definitions and units.

Variable	Definition and units
F_{H1}	The heat flow rate of stream H_1 (kW/K)
Q_c	Heat load in the cooler (kW/K)

This application has only one uncertain parameter, the heat flow rate of stream, F_{H1} . Its nominal value is 1 kW/K and expected deviations are $\Delta F_{H1}^+ = 0.8$ and $\Delta F_{H1}^- = 0$. The heat load, Q_c , in the cooler is considered as an adjustable control. In this application, the heat load adds a challenge because this freely adjustable control parameterizes the constraints. The search for control decisions is expected to increase process flexibility. The following inequality constraints are obtained by eliminating the equality constraints of the problem.

$$\begin{aligned}
g_1 &= -25F_{H1} + Q_c - 0.5F_{H1} + 10 \leq 0 \\
g_2 &= -190F_{H1} + 10 + Q_c \leq 0 \\
g_3 &= -270F_{H1} + 250 + Q_c \leq 0 \\
g_4 &= 260F_{H1} - 250 - Q_c \leq 0
\end{aligned} \tag{14}$$

This application requires only one projection cycle, because it has only one uncertain parameter and one control variable. Figure 15 shows the geometry of the feasible region in the space of uncertain parameter F_{H1} and control Q_c as well as the trajectory of the zig-zag projection path with 83 moves. The problem converged to the critical point $F_{H1} = 1.118$, $Q_c = 40.66$ for any choice of initial guesses. This result agrees with the solutions offered by the original authors. The CPU time for this application was 123ms, which is acceptable. The application shows that our method was reliable, requires no derivative information, and converges in reasonable CPU times.

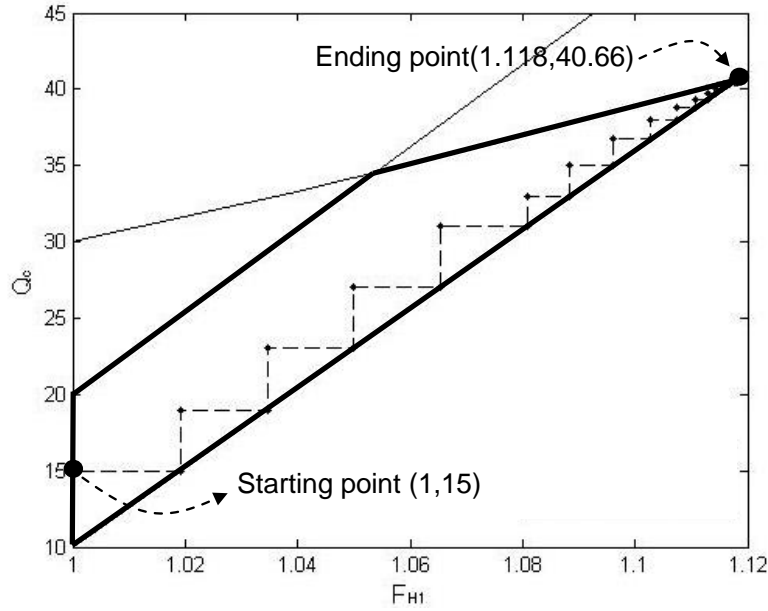


Figure 15: The feasible region of application 2 and the trajectory of the suggested projection technique starting from the point (1,15). Our method converged to the critical point $F_{H1} = 1.118$, $Q_c = 40.66$.

4.3 Application 3: Pump and pipe with uncertainty in operating pressure

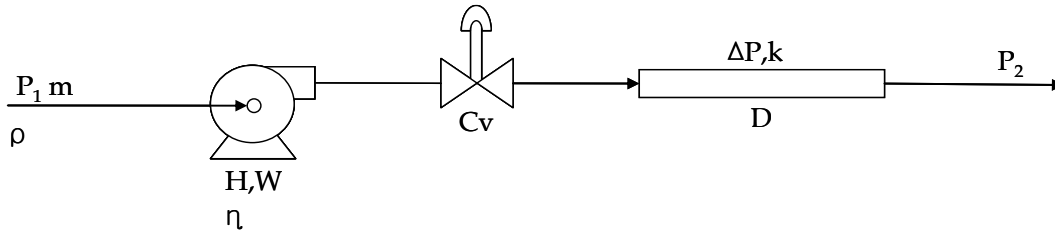


Figure 16: The flowsheet for the centrifugal pump and pipe case study.

This test application involves a centrifugal pump and a pipe as shown Figure 16 (Swaney & Grossmann, 1985; Grossmann & Floudas, 1986; Floudas et al., 2001). A description of the variables and units is provided in Table 4.

Table 4: The variables of pump and pipe application

Variable	Definition and units
m	Liquid flow rate (kg/s)
P_1	Inlet pressure of liquid flow rate (kPa)
D	Pipe diameter (m)
H	Pump head(kJ/kg)
W	The driver power (kW)
C_v^{\max}	The control valve size
P_2^*	Desired pressure(kPa)
η	The pump efficiency
k	Pressure drop in the pipe (kPa)
ε	Tolerance for the delivery pressure
ρ	The liquid density
r	The control range ($0 < r < 1$)

The pump transports the liquid at pressure, $P_1 = 1,000\text{kPa}$, and flow rate, m , to a destination at pressure, P_2 . The delivery pressure, P_2 , must remain within a tolerance, ε , of the desired level, P_2^* . In this application, P_2^* and m , are considered uncertain parameters, and, C_v , is the control variable. This problem has nonlinear constraints and one adjustable control. The nominal and the expected deviation of the uncertain parameters are shown in Table 5.

Table 5: Nominal points and expected deviation of application 3

Parameter	P_2^* (kPa)	m(kg/s)
Nominal value	800	10
+ deviation	200	2
- deviation	550	5

D, H, W and Cvmax are regarded as design variables and their values are D=0.0762 m, H=1.3 kJ/kg, W=31.2 kW, Cvmax =0.0577. The values of the other constants are $k = 1.101\text{E-}6\text{kPa}$, $\rho = 1000 \text{ kg/s}$, and $\eta = 0.5$, $\epsilon=20$. By eliminating all equality constraints, we arrive at problem (15).

$$\begin{aligned}
 g_1 &= P_1 + \rho H - \epsilon - m^2 C_v' - km^{1.84} D^{-5.16} - P_2^* \leq 0 \\
 g_2 &= -P_1 - \rho H - \epsilon + m^2 C_v' + km^{1.84} D^{-5.16} + P_2^* \leq 0 \\
 g_3 &= mH - \eta W \leq 0 \\
 0.3 &\leq C_v' \leq 120 \\
 250 &\leq P_2^* \leq 1000 \\
 5 &\leq m \leq 12
 \end{aligned} \tag{15}$$

For a complete discussion, the reader is referred to Swaney and Grossmann (1985).

A variable, C_v' , is introduced artificially for simplifying the model representation. The variable C_v' introduced in equation (16) serves as the control.

$$C_v' = \frac{1}{\rho C_v^2} \tag{16}$$

This application was solved with the following parameters: population size=400, selection ratio=0.5 mutation ratio=0.05, maximum number of generations=50. The critical point was $m=11.233$, $P_2^*=923.408$ with a flexibility index value of 0.617, which is the same result as in the original work (Grossmann & Floudas, 1986). Figure 17 depicts the location of the correct real critical point. The CPU time for robust global problem solution was 57.5 sec.

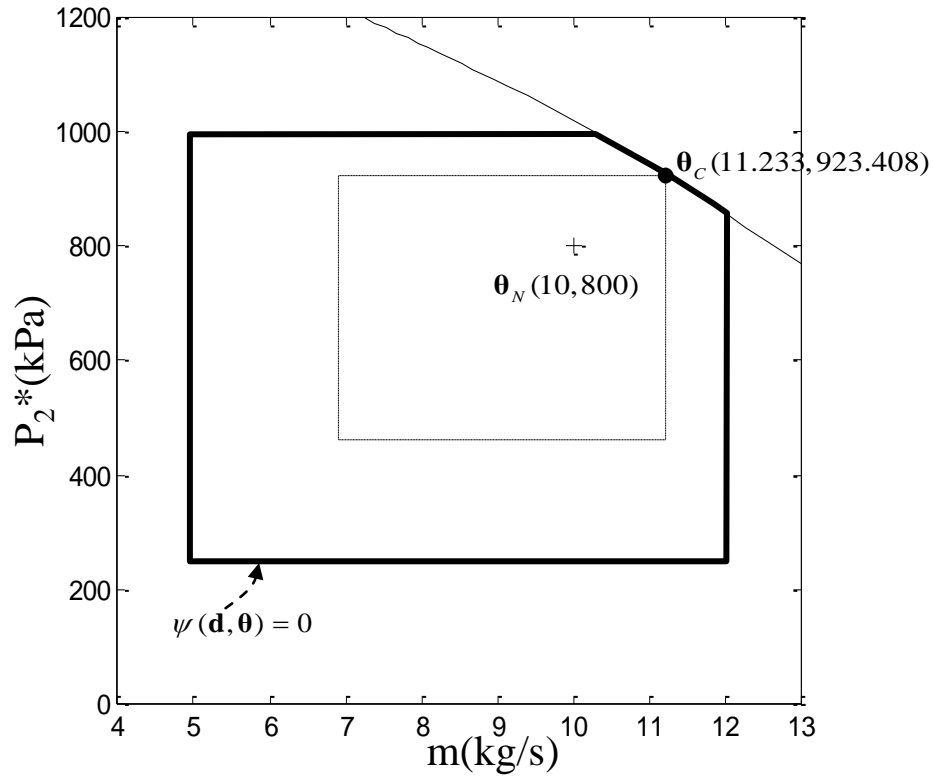


Figure 17: The result for the pump and pipe application depicting the critical point ($m=11.233$, $P_2^*=923.408$) as a corner of the hyper rectangle touching the feasible region.

4.4 Application 4: Reactor and cooler system with uncertain kinetics

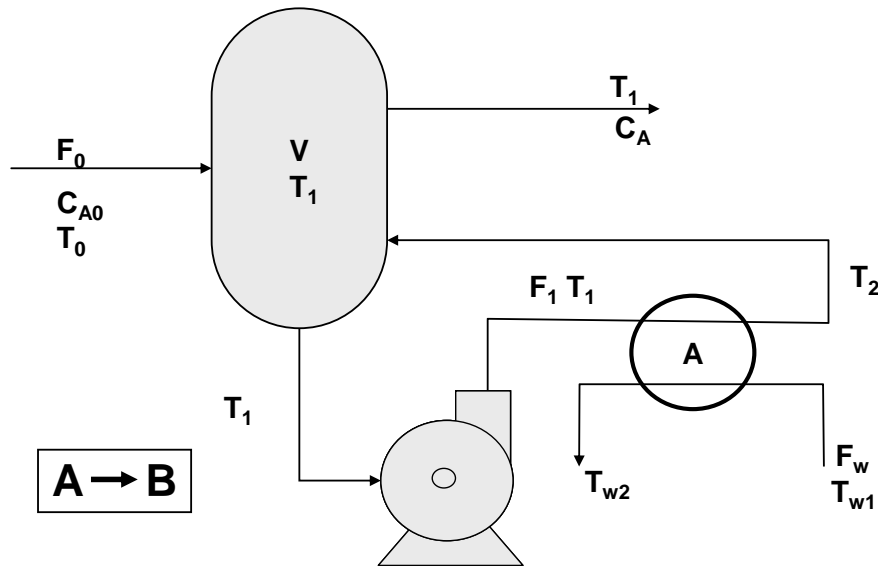


Figure 18: The flowsheet of first-order reactor system. It involves a heat exchanger and a reactor with first order exothermic reaction.

This case examines the flexibility of a simple flowsheet consisting of a reactor and a heat exchanger shown in Figure 18 (Floudas et al., 2001). The reaction is first-order exothermic, 90% of A is converted to B. The mathematical model involves mass balance, heat balances of the reactor and the heat exchanger. This problem features large number of equations, nonlinearity in the constraints and presence of three nonlinear controls. By eliminating equality constraints in the model, eight inequality constraints are obtained as in system (17).

$$\begin{aligned}
g_1 &= \frac{F_0 c_p (T_1 - T_0) + AU \Delta T_{ln}}{(-\Delta H)F_0 - F_0 c_p (T_1 - T_0) - AU \Delta T_{ln}} - \frac{C_{A0} k_0}{F_0} \exp(-E / RT) V \leq 0 \\
g_2 &= 0.9(-\Delta H)F_0 - F_0 c_p (T_1 - T_0) - AU \Delta T_{ln} \leq 0 \\
g_3 &= T_1 - 389 \leq 0 \\
g_4 &= -T_1 + 311 \leq 0 \\
g_5 &= -T_1 + T_2 \leq 0 \\
g_6 &= -T_{w2} + T_{w1} \leq 0 \\
g_7 &= -T_1 + T_{w2} + 11.1 \leq 0 \\
g_8 &= -T_2 + T_{w1} + 11.1 \leq 0
\end{aligned} \tag{17}$$

The design variables are the reactor volume $V=4.6\text{m}^3$ and the area of heat exchanger $A=12\text{m}^2$. The uncertain parameters of this application are the feed flow rate F_0 (kmol h^{-1}) and the reaction rate constant k_0 (h^{-1}). Table 6 lists nominal values and deviations of uncertain parameters.

Table 6: Nominal points and expected deviation of application 4

Parameter	$F_0(\text{kmol h}^{-1})$	$k_0(\text{h}^{-1})$
Nominal value	45.36	12
+ deviation	22.68	1.2
- deviation	22.68	1.2

The control variables are the reactor temperature T_1 , the temperature of the outlet stream after cooler T_2 , and the outlet temperature of the cooling water T_{w2} .

The hybrid algorithm parameters used in this application were population size=400, selection ratio=0.5, mutation ratio=0.05, maximum number of generations=50. Floudas et al (2001) reported a flexibility index of 0.064, while our algorithm found a critical point (85.19,12.40), which corresponds to a flexibility index of 1.75 with control variable values of $T_1=389\text{K}$, $T_2=389\text{K}$, $T_{w2}=350\text{K}$ as depicted in Figure 19. The CPU time was 68 sec.

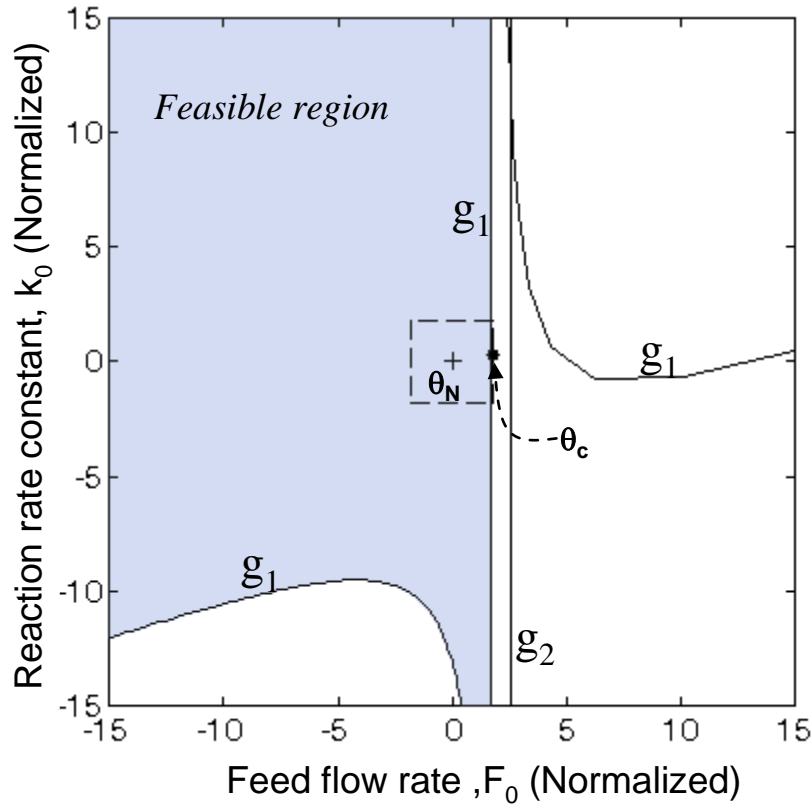


Figure 19: Normalized feasible region and critical point found using hybrid algorithm. We found critical point (85.19, 12.40) which lies on constraint g_1 . The control levels for this region is $T_1=389\text{K}$, $T_2=389\text{K}$, $T_{w2}=350\text{K}$.

4.5 Application 5: Polymerization Reactor with uncertain kinetics

The free radical solution polymerization of styrene in a jacketed CSTR is investigated (Maner, 1996). This case study is the most complex application due to its high dimensionality and adjustable controls. A schematic representation of the reactor depicting the inlet and outlet streams along with all the system variables is given in Figure 20.

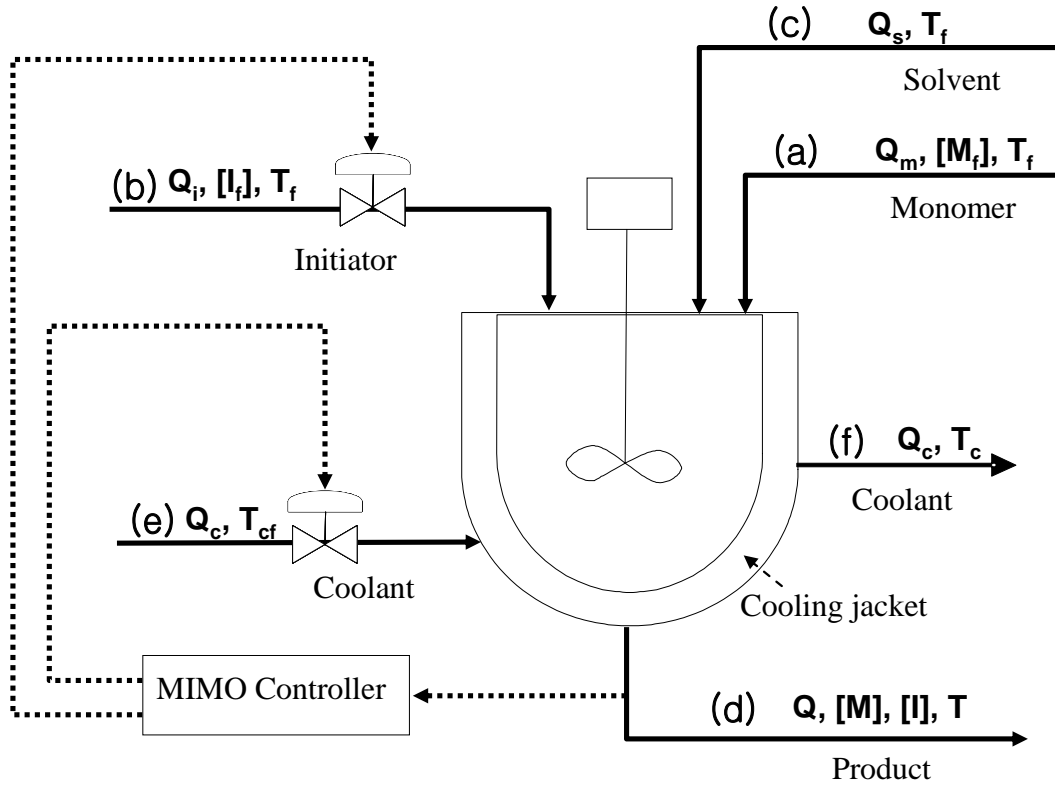


Figure 20: Polymerization reactor. (a) monomer: styrene (b) initiator: azo-bis-isobutyronitrile (AIBN), (c) solvent: benzene (d) the outlet stream: monomer, initiator, and the polymer (e) cooling jacket inlet (f) coolant efflux. The monomer, the initiator, the solvent are inlet streams and produce polymer.

The following three feed streams enter the reactor: pure styrene monomer (a), azo-bis-isobutyronitrile (AIBN), initiator (b) and benzene, solvent (c). The outlet stream (d) contains un-reacted monomer, un-reacted initiator, and the polymer. The reactor has a cooling jacket with inlet (e) and coolant efflux (f). A MIMO feedback controller maintains the desired polymer chain length within desired quality limits by manipulating initiator and coolant flows. The polymer chain length is a function of the ratio of the first and the zeroth moments of the polymer, the number average of molecular weight, D_1/D_0 . Steady state material and energy balances of this process are given in system (18):

$$\begin{aligned}
h_1 &: -k_p [M][P] + \frac{(Q_m[M_f] - Q_t[M])}{V} = 0 \\
h_2 &: -k_d[I] + \frac{Q_i[I_f] - Q_t[I]}{V} = 0 \\
h_3 &: 0.5k_t[P]^2 - \frac{Q_t D_0}{V} = 0 \\
h_4 &: M_m k_p [M][P] - \frac{Q_t D_1}{V} = 0 \\
h_5 &: \frac{(-\Delta H_r)}{\rho C_p} k_p [M][P] - \frac{hA}{\rho C_p V} (T - T_c) + \frac{Q_t (T_f - T)}{V} = 0 \\
h_6 &: \frac{Q_c}{V_c} \cdot (T_{cf} - T_c) + \frac{hA}{\rho_c C_{pc} V_c} (T - T_c) = 0 \\
k_i &: A_i \cdot \exp\left(\frac{-E_i}{T}\right) \quad i = d, t, p \quad \text{and} \quad [P] = \left[\frac{2f^* k_d [I]}{k_t} \right]^{0.5}
\end{aligned} \tag{18}$$

where k_t is the overall chain termination constant, k_p is the propagation constant and k_d is the initiation constant.

The product quality challenge lies in keeping the number average of molecular weight (NAMW) (D_1/D_0) between 52,569 g/mol and 64,361 g/mol expressed mathematically in the following two inequality constraints.

$$g_1 = -\frac{D_1}{D_0} + 52659 \leq 0 \tag{19}$$

$$g_2 = \frac{D_1}{D_0} - 64361 \leq 0 \tag{20}$$

Moreover, the reactor temperature should not exceed 423K.

$$g_3 = T - 423 \leq 0 \tag{21}$$

The uncertain variables of the process are A_d , E_d , A_t , E_t , A_p , E_p , Q_m and T_f . The control variables are Q_I and Q_c . Table 7 lists the nominal values and expected deviations of uncertain parameters. More process details can be found in Maner et al (1996).

Table 7: Nominal values and deviations of uncertain parameters of the polymerization reactor

Parameter	Nominal values	+ deviations	- deviations
-----------	----------------	--------------	--------------

Q_f (l/s)	0.105	0.0021	0.0021
T_f (K)	330	6.6	6.6
A_d (l/s)	5.95×10^{13}	1.1×10^{12}	1.1×10^{12}
E_d (K)	14897	297.4	297.4
A_t (l/mol s)	1.25×10^9	2.5×10^6	2.5×10^6
E_t (K)	843	16.86	16.86
A_p (l/mol s)	1.06×10^7	2.1×10^5	2.1×10^5
E_p (K)	3557	71.1	71.1

Results. This application was solved for three different reactor types: $V=1,000l$, $3,000l$, and $5,000l$. In the simulations, parameters were set as follows: population size=30,000, maximum number of generations=30, selection ratio=0.5, mutation ratio=0.05. The results in Table 8 show that only a $5000l$ reactor guarantees flexible operation in all expected uncertain scenarios.

Table 8: The observed flexibility indexes as a function of the increasing reactor volume

Volume (l)	Flexibility index
1000	0.425
3000	0.865
5000	1.115

The trend of larger volumes increasing flexibility makes sense intuitively. However, the current study only considers steady state performance. In a *dynamic flexibility* analysis considers simultaneous design and control decisions as a *function of time*, the trade-off between size and flexibility exhibits an optimum. Integrated design and control decisions for achieving dynamic flexibility are discussed elsewhere (Malcolm et al. 2007).

4.6 Parallelization

Stochastic algorithms like our hybrid algorithm tend to be computationally expensive due to many function evaluations. Fortunately, our proposed method is easy to parallelize. This option for massive parallelization will be shown as a major strength of the proposed

method. We have tested the *master-slave model* (Hauser, 1994) in our algorithm as shown in Figure 21. One processor, called the *Master* consists of two parts, the *GA module*, and the *scheduler*. The *GA module* implements selection, mating, and mutation procedures and keeps overall results. The *scheduler* controls communication between the master and the slave nodes. Moreover, it tracks the response time of each slave to optimally distribute computational function evaluation tasks in packages proportional to the available free CPU capacity. The optimal scheduling of jobs according to performance feedback helps avoiding *bottle necks*. According to the communication between GA module and scheduler, packages of uncertain parameter sets and required evaluation tasks are sent from the *master* to *slave nodes*. The *slave nodes* perform the projection technique onto the feasible region boundary, implement the Newton-type solution of the non-linear state equation sets and evaluate the flexibility δ of each candidate solution. The final performance of each sample in terms of a δ value and the corresponding corrected θ values positioned exactly on the feasibility boundary, ψ , are sent back to *master* node via a TCP/IP communication usually via a binary large objects (BLOB).

To demonstrate the advantages of parallelization, we solved the flexible polymerization reactor problem in parallel with increasing number of *slave* computer nodes. Figure 22 compares the required computational times achievable with a single processor with the parallel multiple processor network. The CPU time for this application with single processor was 102 minutes. With three slave processors the program took 34 minutes. This is three times faster than the single CPU approach. With six processors, computational time was cut to 20 minutes, which is still six times faster than a single processor. Finally, we used nine processors the CPU time was further reduced to 12

minutes. These experiments demonstrate that the speed of computation increases linearly with the number of processors. In all cases of the extensive collection of case studies, the hybrid algorithm exhibited correct and robust convergence as well as efficient performance.

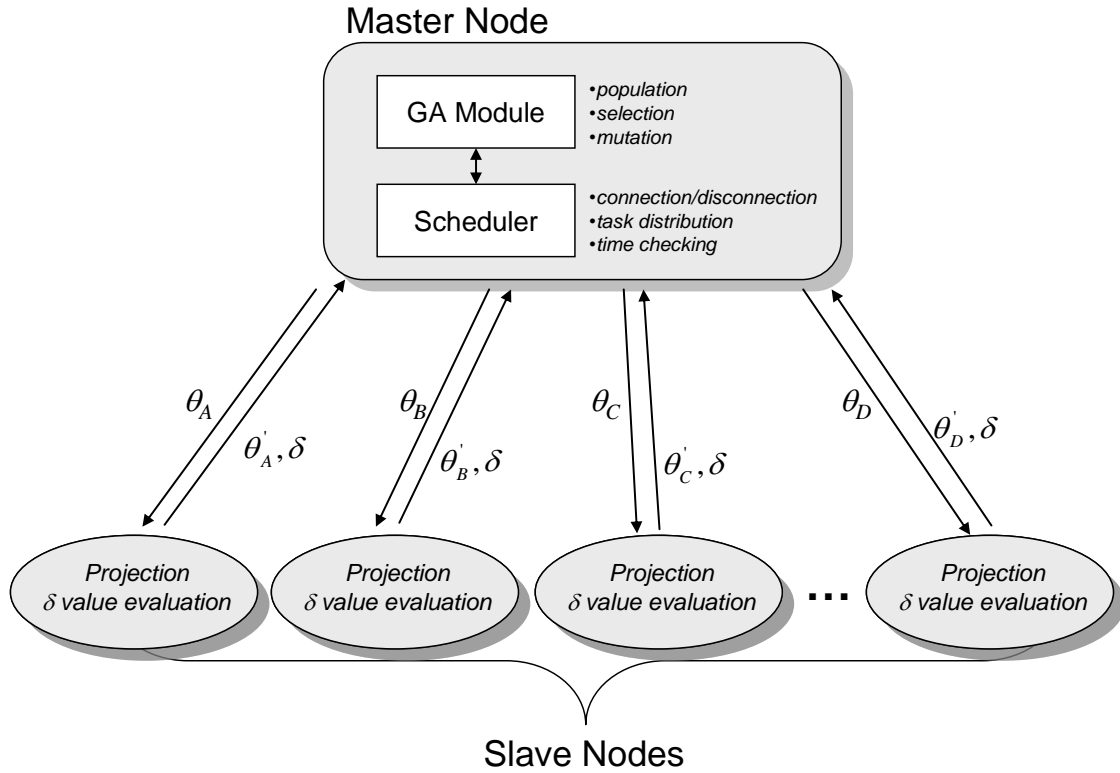


Figure 21: A master-slave framework for parallel implementation of the hybrid algorithm: the master executes GA operations and distributes sample θ to slave nodes. The slave nodes project θ onto the region of boundary and evaluate the cost and send it back to master.

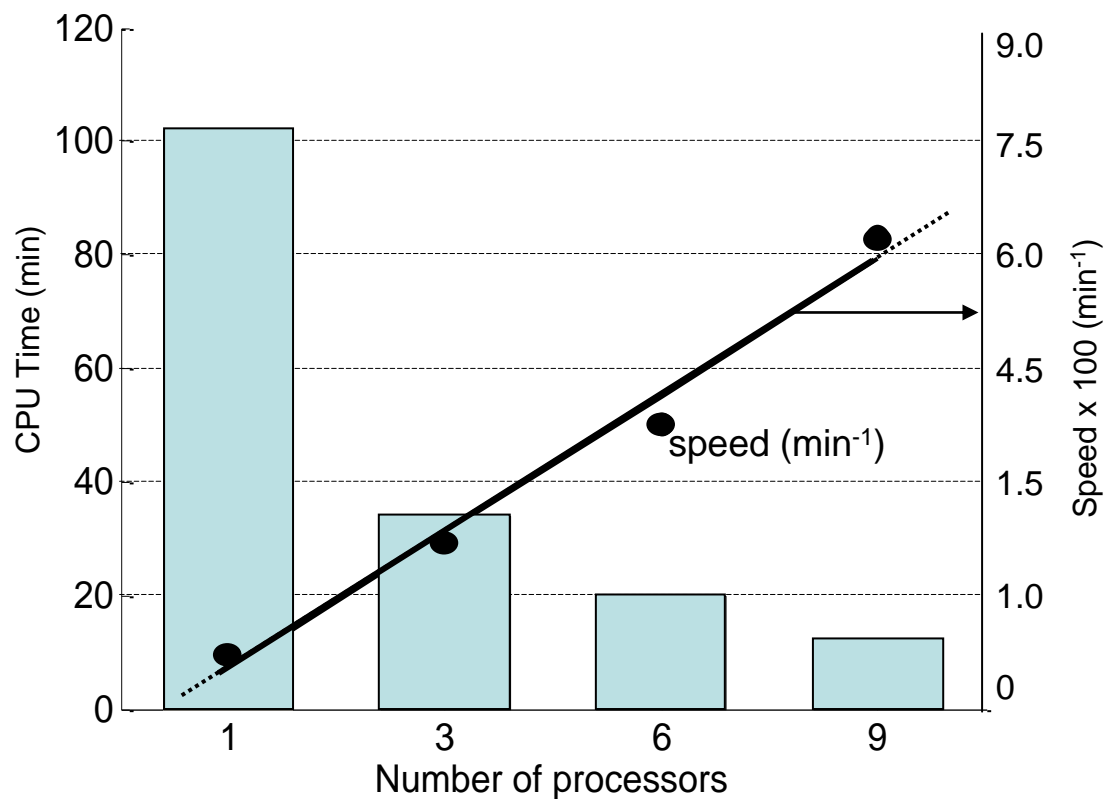


Figure 22: Computational time (left hand axis) vs number of processors in application 5 : 9 processor system is approximately one and half times faster than 6 process system, three times faster than 3 processor system, and nine times faster than a single processor system. The computational speed increases linearly with the number of processors. (right hand axis).

5 Conclusions

This chapter introduces a new hybrid algorithm combining the advantages of stochastic search techniques method a deterministic nearest constraint projection technique for the flexibility index problem. The algorithms may be useful in some cases where deterministic global optimization methods are impractical or unavailable. Due to the NP-hardness of the search space, one should not expect any algorithm to be superior in all cases; a list of selection criteria when to use the proposed hybrid framework follows.

The novel method appears to offer the following main advantages. The direct implementation using the hybrid algorithm does not require derivative information or introduction of artificial variables such as Lagrangian multipliers or problem transformation to a min-max optimization approach. The proposed method even admits the direct use of black box models or legacy code to perform flexibility analysis without manipulating the original problem formulation. Secondly, the proposed method is very robust without failing to identify the correct global solutions in all cases of a comprehensive test suite. Five applications were studied to demonstrate the methodology's effectiveness for solving non-convex flexibility index problems. Situations in which traditional local MINLP solvers fail to find the correct solution were discussed.

Finally, we found the performance of the method to be competitive even when using only a single CPU. Moreover, massive parallelization is easy to implement due to the simple problem formulation and the independent decomposable solution strategy offered by the evolving candidate solution population. A linear acceleration up to a ten nodes was demonstrated with a realistic problem. For large applications drastic reductions in required

computations times can be expected. The ease-of-use as well as the option for parallelization may be considered strong points of the proposed method.

When deploying stochastic method, robustness is often traded against drastic increases in CPU time. The proposed hybrid algorithm is promising in delivering both reliable convergence as well as reasonable performance in difficult global flexibility analysis.

Nomenclature (Chapter I)

A	area of heat exchanger (m^2)
C_v	valve constant ($kgm^3/kPa)^{5/s}$
C_v^{max}	control valve size
\mathbf{d}	vector of design variables
D	pipe diameter (m)
D_0	molar density(g/l)
D_1	mass density(g/l)
F	flexibility index
F_{H1}	heat flow rate of stream H_1 (kW/K)
F_0	feed flow rate of reactor inlet ($kmol\ h^{-1}$)
g_j	inequality constraints
h_i	equality constraints
H	pump head(kJ/kg)
k_d	initiation constant (h^{-1})
k_o	reaction rate constant (h^{-1}).
k_t	overall chain termination constant (h^{-1})
k_p	propagation constant (h^{-1})
m	liquid flow rate (kg/s)
N_n	normal distribution
Q_c	cooling water flow(l/s)
Q_I	initiator flow(l/s)
Q_m	monomer flow(l/s)
P_1, P_2	inlet and outlet of pressure (kPa)
P_2^*	desired pressure (kPa)
r	control valve range
T_1	reactor temperature (K)
T_2	temperature of the outlet stream after cooler(K)
T_f	inlet temperature(K)
T_{w2}	outlet temperature of the cooling water(K)
V	reactor volume(m^3)
W	driver power (kW)
\mathbf{x}	vector of state variables
\mathbf{z}	vector of controls
Z_{min}, Z_{max}	minimum and maximum control range

Greek symbols

δ	<i>infinity norm</i> , the parameter deviation
$\Delta\theta$	vector of expected deviations
Δ_{step}	Step size

$\Delta \theta^*$	interval range
ε	tolerance for delivery pressure
η	pump efficiency
θ	vector of uncertain parameters.
θ^N	vector of nominal value.
θ^c	vector of the critical point
θ^*	uncertain parameter dimension
θ^U, θ^L	upper and lower bound of uncertain parameter
ρ	liquid density
ψ	feasibility function
σ	the standard deviation
ω	direction orientation.

PART B: Integration of Design and Control

High-performance processes require a design that operates close to design boundaries and specifications, while the robust performance is guaranteed with satisfying all design constraints. To approach tighter boundaries of process performance safely, much attention has been devoted to integration of design and control that considers dynamic controllability and the design decisions simultaneously. However, the mathematical formulations of rigorous methods that solve design and control simultaneously are challenging and easily become intractable numerically and computationally. This part introduces a new mathematical formulation to reduce this combinatorial complexity of integrating design and control. We will show that a substantial reduction in problem size can be achieved using embedded control decisions within specific designs. These embedded control decisions avoid a combinatorial explosion of control configuration, using a full state space model that does not require a pairing of control variables and loops. The current capabilities of the methodology will be demonstrated using a realistic reactor column flowsheet.

CHAPTER II. OVERVIEW OF EMBEDDED CONTROL OPTIMIZATION

Previously published as:

Moon, J., Kim, S., and Linninger, A.A., “Integrated Design and Control under Uncertainty: Embedded Control Optimization for Plantwide Processes”, *Computers & Chemical Engineering* 35 (9), 1718-1724, 2011

1 Previous works and difficulties in integration of design and control

Integration of design and control received attention in the scientific community for the last 30 years and several methodologies have been developed. Controllability was studied based on such as Right Half Plane (RHP) zeros, relative gain analysis, stability analysis or linear-quadratic Gaussian (LQG)-based dynamic measures (Kuhlmann & Bogle, 2001; Luyben et al., 1996; Papalexandri & Pistikopoulos, 1994; Perkins & Wong, 1985; Psarris & Floudas, 1991). These analyses are simple and easy to apply, so they are suitable for large-scale processes, even though, they are limited to steady state or linear dynamic models. The trade-off between economic benefits and controllability with multi-objective criteria were also developed. (Bregel & Seider, 1992; Lenhoff & Morari, 1982; Luyben & Floudas, 1994; Palazoglu & Arkun, 1986). The main drawback of these approaches lies in difficulties to quantify the controllability for incorporation in the objective function alongside capital cost. Other approaches deal with a single economic objective function, while avoiding dynamic constraint violations. These methods used dynamic optimization to obtain the best design that satisfies all dynamic constraints (Bahri et al., 1997; Bansal et al., 2002; Contou-Carrere et al. 2004; Kookos & Perkins, 2001; Mohideen et al., 1996a, 1996b; Perkins & Walsh, 1996; Walsh & Perkins, 1994). Excellent reviews of integrated design and control methodologies can be found elsewhere (Sakizlis et al., 2004; Seferlis & Georgiadis, 2004). Unfortunately, few methodologies for design

and control integration are suitable for plantwide process scope because of its mathematical complexity. The conceptual problem of the integration of design and control under uncertainty is formulated as a stochastic infinite dimensional mixed integer dynamic optimization problem. The solution of integrated design and control problems usually requires much computational time, integer decisions, and non-convex equations. This problem poses an extreme challenge to existing mathematical programming techniques. Moreover, control feedback may introduce instability for certain parameter realizations. The mathematical complexity and large computational time requirement make it impossible to apply current optimization algorithms for a large scale, plant-wide process cases.

Our group (Malcolm et al.,2007) proposed a new method entitled *embedded control optimization*. This integrated design and control method reduces the combinatorial complexity of the nonpolynomial-hard search space. It delegates control decisions to a suboptimization step, which adaptively adjusts suitable control moves for a given design. Thus control decisions are embedded for each candidate design. Therefore, we propose to use embedded control optimization for the plantwide process which optimizes the control choices adaptively so that In next two sections of this chapter, the embedded control optimization methodology is reviewed.

2 Mathematical programs decomposition for design under uncertainty

Embedded control optimization adapts a two-level hierarchical design procedure as summarized in Table 9. In level-1, a mathematical process model is built and types of variable are defined. In level-2, the mathematical programming framework optimizes design and control decisions to maximize robust expected performance.

Table 9: Proposed decision hierarchy for integrated design and control (Malcolm et al, 2007)

<u>Level-1: Dynamic modeling, flexibility concepts and structural decisions:</u>	
Identify state variables, x , and formulate conservation laws and constitutive equations. Select design variables, d , controls, c , and characterize uncertainty sources.	
<u>Level-2: Design optimization:</u>	
Perform integrated design and control optimization steps with increasing level of complexity:	
<ul style="list-style-type: none"> • Mathematical modeling of the uncertain space • Dynamic stochastic optimization of the expected performance • Stability • Dynamic feasibility 	

2.1 Level-1: Modeling and structural decisions

The mathematical model based on fundamental conservation laws and first principles is built for dynamics of process in level-1. And four types of variables used in the model are defined. The types of variables are classified into four categories: (i) design decisions, d , (ii) control decisions, c , (iii) uncertainty sources, θ and ξ , as well as (iv) state variables, x . Table 10 summarizes variables categories of the proposed methodology.

Table 10: Variable types in integrated design and control (Malcolm et al, 2007)

Variable Type	Symbol	Type	Example
Design	d	Discrete	<i>Structural decisions: Connectivity</i>
		Continuous	<i>Reactor volume, Column length</i>
Control	c	Discrete	<i>Control configuration, Control type</i>
		Continuous	<i>Set points, Control tuning parameters</i>
State	x	Continuous	<i>Temperature, Composition, Pressure</i>
Uncertainty	θ	Time independent	<i>Parametric uncertainty</i>
	$\xi(t)$	Time dependent	<i>Variations due to seasonal changes</i>

Design variables is defined as two kinds- d , -discrete variables and continuous variables. Discrete variables represent structural decisions such as the connectivity or existence of units. Continuous variables represent equipment size or operating condition.

For instance, the choice of reactor type-CSTR or Plug flow reactor would be a discrete decision. The reactor size, or column length are continuous variables. The control variable set, c , represents alternative controller structures, tuning parameters, and set-points. Uncertain variables are categorized into two types- time invariant type (θ) and time variant type (ζ). Static uncertain parameters θ vary randomly within a certain value ranges. All periodical changing pattern of influences are represented as $\zeta(t)$.

2.2 Level-2: Problem decomposition and optimization

In level-2, the optimization problem defined in level -1 is solved to obtain the optimal design that is the most economic and robust to expected uncertainties and dynamic disturbances. As mentioned above, the problem is a stochastic infinite dimensional mixed integer dynamic optimization problem and the solution requires expensive computational time. To overcome the intractability of the original problem, Pistikopoulos and co-workers proposed a problem decomposition algorithm as shown in Figure 23 (Mohideen et al., 1996a). In their decomposition technique, the optimal design choices are made stochastically in a discrete sampling space of a stochastic framework. Control decisions are taken at the same level as design decisions. Since the discrete sampling space does not include all critical scenarios, an another search for critical constraint violations is required. Accordingly, the rigorous feasibility test explores whether the current design and control choices are feasible in the entire uncertain space. If a new critical scenario is found, it is added to the discrete sample spaces. Thus, this decomposition technique requires three steps: sampling (A), main optimization (B), and feasibility test (C). The embedded control strategy also adopts this decomposition framework, but avoids the combinatorial explosion of discrete control configurations by the embedded control principles.

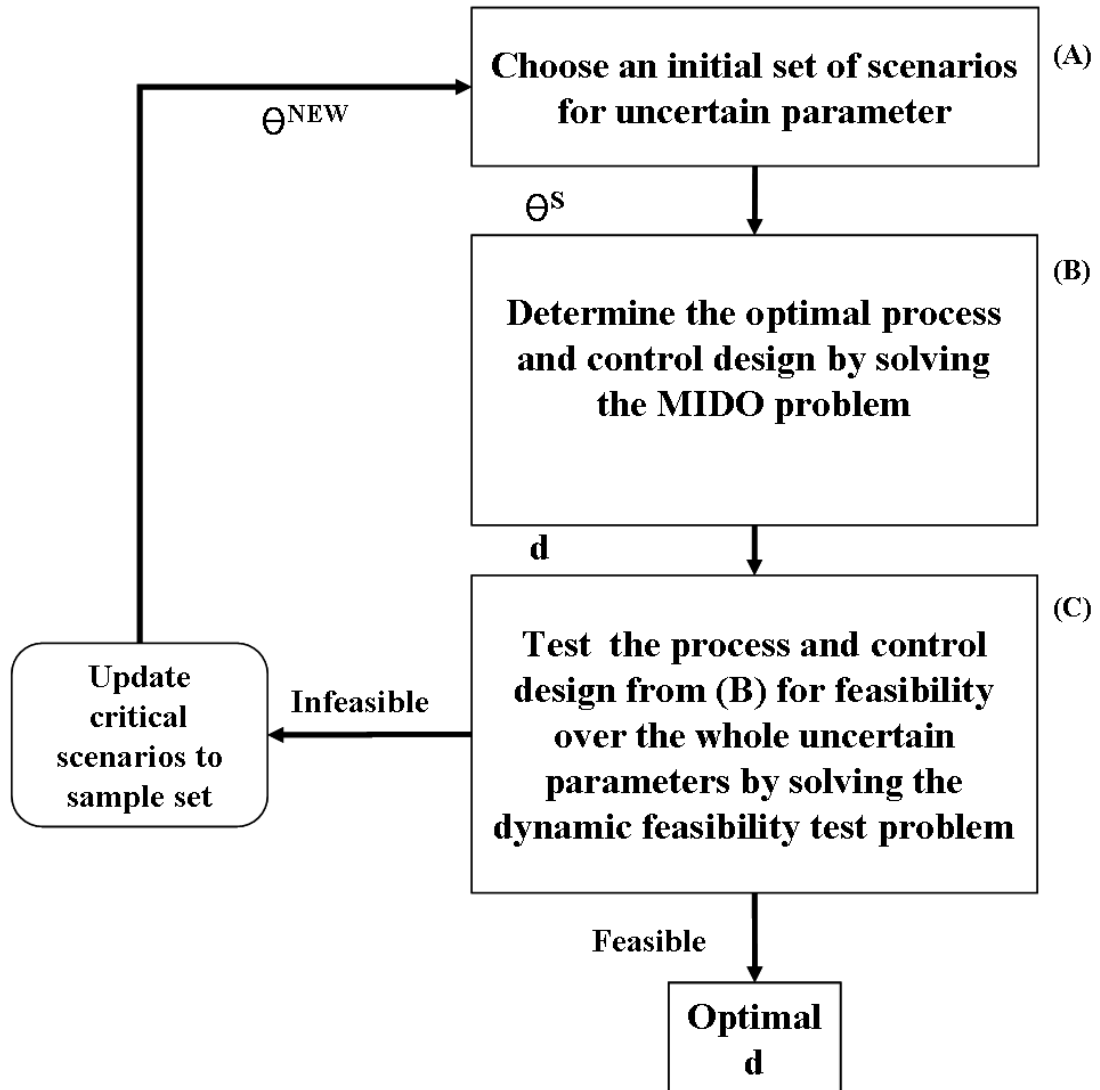


Figure 23: Decomposition algorithm for integrated design and control under uncertainty. Main optimization problem (B) is separated from feasibility test (C). (Mohideen et al., 1996a)

Several sampling techniques such as the Monte Carlo (James, 1985) or Latin hyper cube sampling (Mckay, Beckman, & Conover, 2000) can be used for creating a representative sample of the uncertain operations or parameters. In step (B), a probabilistic objective of the main optimization problem is minimization of total expected cost, Eq. (22). Equality constraints include conservation laws, h_c , Eq. (23), and the selected control algorithm, hCTR, Eq.(24). Inequalities, g , enforce safety, equipment and production constraints at specific instances in time or in an integral sense (Eq.(25)).

$$\min_{d, c, x(t)} \Gamma = \underbrace{\int_{t=0}^{t_{\max}} \int_{s \in \Omega} \omega^s \cdot C_1(d, c, x(t), \theta^s, \xi^s, t) ds dt}_{\text{Expected Operating cost}} + \underbrace{C_2(d, c)}_{\text{Capital Cost}} \quad \text{Minimize Total Expected Cost} \quad (22)$$

s.t.

$$h_c(d, \dot{x}(t), x(t), \theta^s, \xi^s, t) = 0, \quad \forall s \in \Omega \quad \text{Conservational Laws} \quad (23)$$

$$h_{CTR}(d, c, \dot{x}(t), x(t), t) = 0, \quad \dot{x}(0) = x^0 \quad \text{Control Algorithm} \quad (24)$$

$$g_j(d, c, x(t), \theta^s, \xi^s, t) \leq 0 \quad \forall s \in \Omega \quad \text{Process and Product Constraints} \quad (25)$$

Since the solution obtained in step (B) does not guarantee the process feasibility in the uncertain space, a rigorous feasibility test exploring the whole entire uncertain space must be followed in step (C). If critical scenarios causing failure are found, they are added to the sample set, and the design needs to be optimized again. Mathematically, the flexibility analysis of a given design d is formulated as a max–min–max problem as described in (26).

$$\begin{aligned} F(d) &= \max \delta \\ \text{s.t.} \quad & \max_{\theta(t) \in T(\delta, t)} \min_{z(t) \in Z(t)} \max_{j \in J, t \in [0, H]} g_j(d, x(t), z(t), \theta(t), t) \\ & h_j(d, \dot{x}(t), x(t), z(t), \theta(t), t) = 0, \quad x(0) = x^0 \\ & g_j(d, x(t), z(t), \theta(t), t) \leq 0 \\ & \delta \geq 0 \\ & T(t, \delta) = \left\{ \theta(t) / \theta^N(t) - \delta \Delta \theta^-(t) \leq \theta(t) \leq \theta^N(t) + \delta \Delta \theta^+(t) \right\} \end{aligned} \quad (26)$$

For this feasibility analysis, several programming techniques are available. (Dimitriadis & Pistikopoulos, 1995; Grossmann & Floudas, 1987; Moon et al., 2008; Swaney & Grossmann, 1985). However, because design and control are addressed simultaneously; this formulation easily leads to infinite permutations of structural control and design decisions. Moreover, the solution of design with control feedback is a non-polynomial dynamic mixed-integer global optimization problem. The embedded control strategy also adopts this decomposition framework, but avoids the combinatorial explosion of discrete control configurations by the embedding control.

3 Embedded control optimization

Even though the problem decomposition suggested by Pistikopoulos substantially reduces the size of problem, it is still challengeable because of combinatorial complexity of the NP-hard search space. Specifically, the control decisions such as feedback closed loops, or pairing of manipulated and control variables cause a combinatorial explosion in the number of possible process design and control realizations. We therefore propose to separate the design decisions from the control decisions by a decision position technique shown in Figure 24. The master level makes design decisions such as reactor dimensions, residence time, reactor length, and diameter that dominate dynamic process performance. Control decisions are not made at this level. The optimal control actions are dynamically computed during the dynamic simulation subprocess based on the optimal regulation applied to a linear full state space model. The full linear state space model is dynamically adapted with the moving horizon estimator. With the full state space model and a multiple-input and multiple-output control strategy, the embedded control algorithm avoids combinatorial problems of pairing control variables with manipulated variables. Moreover,

the control optimization is separated from the main design optimization. In the current stage, we chose a Linear Quadratic Regulator (LQR) to compute the best control action to minimize a simple cost function as given in (27) and (28). The optimality condition of this problem admits an analytical solution; therefore, its computation is very economical. Accordingly, the regulator chooses an optimal control moves for a given design. It also ensures closed-loop stability and circumvents the combinatorial challenge of control variable pairing.

$$\min_{K^*} J = \sum_0^{\infty} \left(x_k^T Q x_k + u_k^T R u_k \right) \quad (27)$$

$$\begin{aligned} \text{s.t.} \quad & \bar{x}_{k+1} = A\bar{x}_k + B u_k^* \quad \bar{u}_k^* = -K^* \bar{x}_k \\ & \text{With } u_k^* = u_{k-1} + \bar{u}_k^* \quad \text{and } \bar{x}_k = x_k - \text{Set Point} \end{aligned} \quad (28)$$

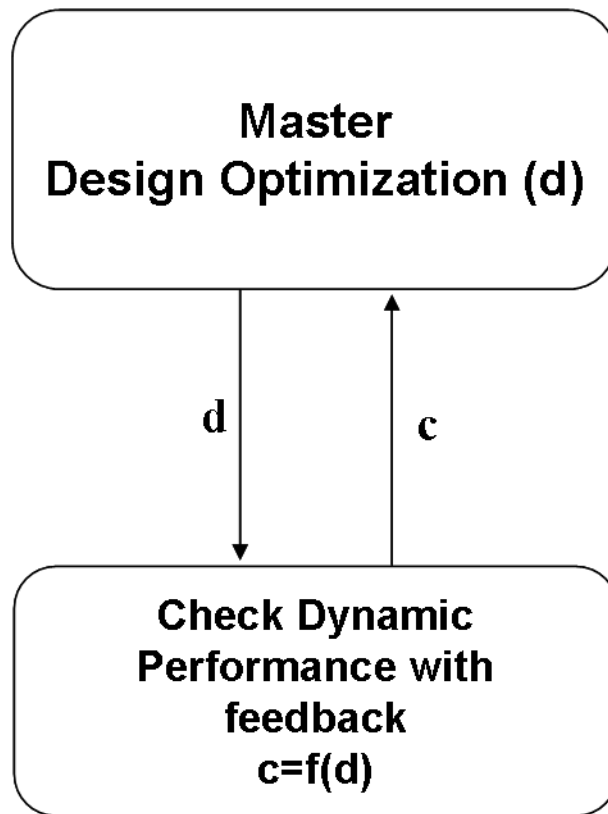


Figure 24: Embedded optimization. A two-hierarchical loop renders more tractable optimization problem to determine the best economical design that has also dynamic controllability.

The LQR needs a linear state space model. We produce a linear process model by mapping the nonlinear dynamic process equations into a linear state space model using linear identification methods. This identification is executed in every step of the discretized time horizon. The required input–output data sets are obtained by sampling the dynamic system model at suitably chosen sampling intervals. The adaptive identification solves a least square fitting problem. The sequential least squares method (Hsia, 1977) is suitable because it produces state space model updates recursively with little computational cost, without having to solve least square optimization problems (Malcolm et al., 2007). In this work, we also use a *Moving Horizon Estimation technique* (Gatzke and Doyle, 2002; Haseltine & Rawlings, 2005; Liebman, Edgar, & Lasdon, 1992; Rawlings & Bakshi, 2006). Moving horizon estimation is more efficient than full information estimation (FIE). Also it shows better performance with linearizing models than sequential least square approach used in our previous paper, because it only considers a recent data set for linearization. Our previous experience shows that the FIE method performs poorly in highly nonlinear processes so that the estimator and real process performance often disagreed, making it difficult for least square regulator to ensure satisfactory performance.

The basic strategy of moving horizon estimation is illustrated in Figure 25. It estimates the state and parameters using a moving data window of fixed size. When new observation become available, new data are added to the data window and the same amount of oldest data is removed from the window. It provides a generic approach to state and parameter estimation, which can be applied to both linear and nonlinear processes. When introducing the moving horizon technique, past dynamics are slowly discounted; moving horizon window captures process dynamics even in highly nonlinear transitions. We can

obtain a linearized model at certain discretized time with a finite set of data by solving a least square problem described in (29).

$$\begin{aligned} \min_{A,B} \quad & \sum_{i=k-H+1}^k \|x_i - \bar{x}_i\|^2 \\ \text{s.t} \quad & \bar{x}_{i+1} = A\bar{x}_i + Bu_i \end{aligned} \quad (29)$$

where x is process data vector, \bar{x} is linear model data, and H is horizon length

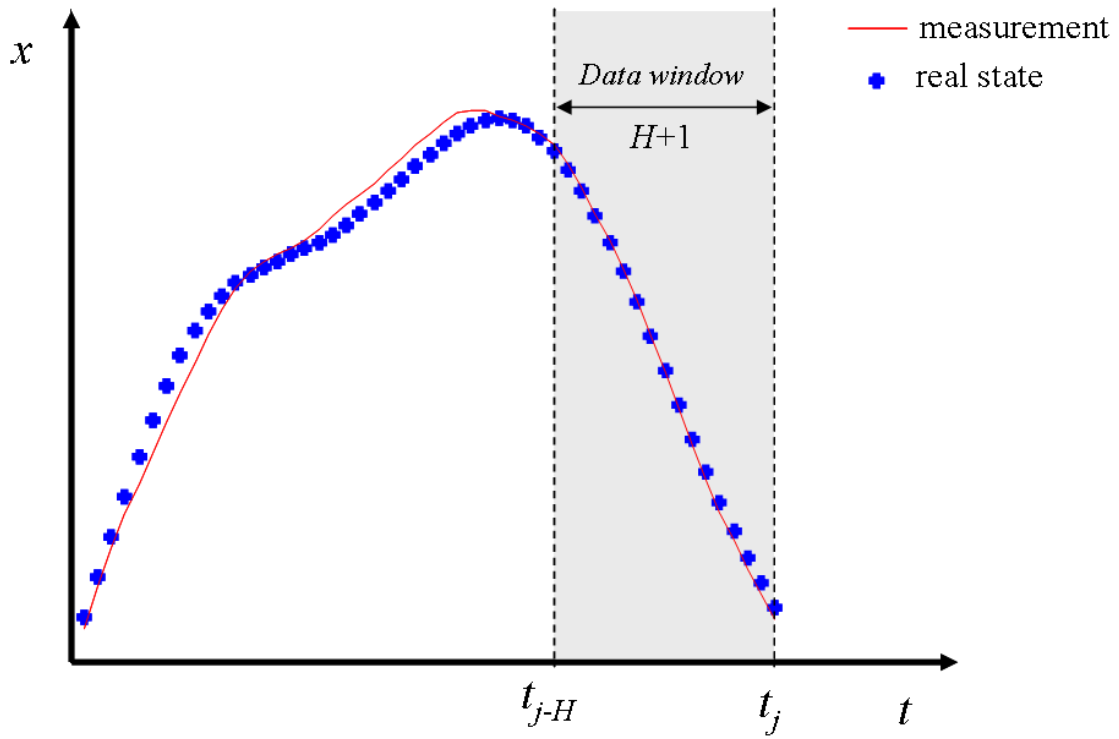


Figure 25: Moving horizon estimation-basic concept .The state is estimated from a horizon of the most recent $H+1$ output measurements that moves forward at each sampling time when a new measurement is available.

The next chapter demonstrates the effectiveness of embedded control optimization strategy by designing an entire process flowsheet with control and design integration under uncertainty.

CHAPTER III. DESIGN AND CONTROL INTEGRATION FOR ISOMERIZATION PROCESS

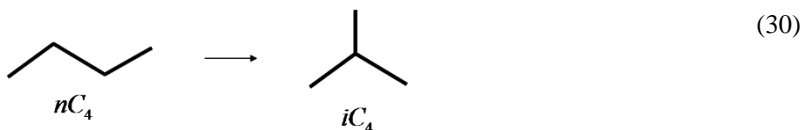
Previously published as:

Moon, J., Kim, S., and Linninger, A.A., “Embedded Control for Optimizing Flexible Dynamic Process Performance” , Industrial and Engineering Chemistry Research, 50 (9), 4993–5004, 2011

This chapter demonstrates the effectiveness of embedded control optimization for an entire flowsheet. We show the large scale case study- for integrated design and control of an isomerization process under uncertainty (Luyben et al., 1998). The aim of this case study is to determine optimal design specifications for optimal performance under uncertainty with reasonable control for dynamically flexible operations. This task of design and control integration should be done simultaneously with reasonable computational effort.

1 Isomerization Process description

Isomerization process converts normal butane to isobutene, as shown in (30).



This process is important because isobutane is a valuable primary feedstock component for motor fuel alkylation. Motor fuel alkylate is an environmentally superior blending component that has paraffinic, high-octane, low-vapor-pressure blending properties.

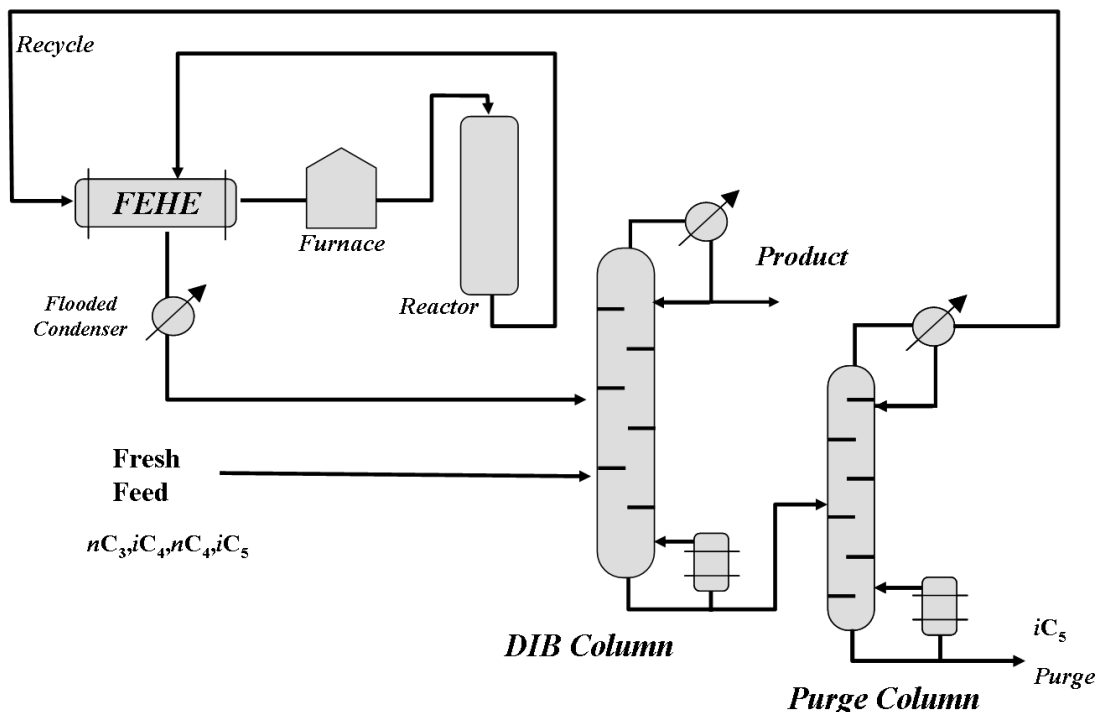


Figure 26: Isomerization process (Luyben et al., 1998). In this process, normal butane is converted to isobutane. The process is comprised of a reactor, a feed-effluent heat exchanger, and two distillation columns.

The isomerization flowsheet consists of a PFR, a Feed Fluents exchange heater, and two distillation columns, as shown in Figure 26. The input feed is the mixture of nC_4 and iC_4 . It also has a small amount of propane (nC_3) and isopentane (iC_5). Since the input feed already has some amount of iC_4 , it does not enter the reactor directly. It enters *Deisobutanizer* column (DIB) and some of iC_4 is separated from the input feed. Propane, which is the lightest component, also comes out in the distillate stream. Because of similar volatilities of iso/normal butane, it is hard to separate iC_4 from nC_4 . Thus, relatively higher number of tray and reflux ratio are required. The bottom feed of DIB goes into the purge column, where most of iC_5 is purged to the bottom stream. The upper stream is vaporized and goes into the reactor by passing the heat exchanger. In the reactor, some of the normal

butane is converted to isobutene in the vapor phase. We assume that this is the first order exothermic irreversible reaction as shown in (31).

$$r = kC_{nC_4}, \quad k = k_o \exp\left(\frac{E}{RT}\right) \quad (31)$$

where r is reaction rate, k is temperature-dependent reaction rate constant, k_o is pre-exponential factor, E is activation energy, and T is the reactor temperature (given in degrees Rankin).

Since the effluent of the reactor should be condensed and liquefied for separation, the input stream of the reactor should be preheated to initiate the reaction. To reduce the energy requirement, a feed effluent heat exchanger (FEHE) is inserted to recover the heat of the reactor effluent. However, preheating is insufficient; therefore, a furnace is used to heat the input stream to the desired temperature. The effluent of reactor is liquefied in the condenser and redirected to the DIB. The stream is above the fresh feed, because the concentration of isobutane in the reactor effluent is higher than that of the fresh feed.

2 Mathematical models for simulation

We used MATLAB to simulate the flowsheet. The isomerization flowsheet is composed of several units that are described as partial differential equations (PDEs). We discretized the pellet reactor with 10 collocation nodes to convert PDEs to ordinary differential equations (ODEs). Also, tray-by-tray models for the DIB column and the purge column simulation were used. In this study, we chose to neglect the dynamics of the heat exchanger and the furnace, because they have fast dynamic responses, compared to other units such as distillation columns or plug-flow reactors, so they are not the dominant units used to study integrated design and control. A dynamic mathematical model was

implemented in MATLAB with 505 variables, 501 equations, and 3 inequality constraints. Details of the mathematical model that describes the isomerization process are described in the Appendix A.

3 Operational constraints and control action strategy

We set two quality constraints and one safety constraint for operating this process. Following the report by Luyben et al. (1998) for the product quality, the mole fraction of nC_4 of the final product (x_1) should be less than 2%. For complete elimination of the heaviest inert (iC_5), the mole fraction of iC_5 (x_2) at the top stream of the purge column should be less than 0.1%. For safety reasons, the reactor pressure (x_3) should never exceed 700 psia (to prevent explosion). To satisfy quality and safety constraints, we use the reflux ratio (u_1) and the vapor ratio of DIB column (u_2), the reflux ratio of purge (u_3), and the input temperature of the reactor (u_4) as manipulated variables.

$$x_1 \leq 0.02 \quad (32)$$

$$x_2 \leq 0.001 \quad (33)$$

$$x_3 \leq 700 \quad (hard) \quad (34)$$

$$x_3 \leq 650 \quad (soft) \quad (35)$$

Among these constraints, the safety constraint (34), which involves the reactor pressure, must be enforced for all time periods of all possible uncertain scenarios. Violation of (34) is unacceptable at any times under any circumstances. On the other hand, it is impossible to keep the product specifications exactly at the set point target in a real operation. Therefore, the quality constraints for bottom and product streams are soft. However, the optimal process should deviate as little as possible for this target. The control

averts any disturbance and keeps the variables close to the desired values; however, it cannot do so without minimal departure of set points. For this reason, we add another conservative constraint (35) to avoid the violation of (34) at any time. Thus, (32), (33) and (35) are soft constraints and (34) is a hard constraint.

Table 11: Design, process, and manipulated variables

Variable Type	Symbols	Description
Design	d_1	Reactor volume size (ft ³)
	d_2	Total tray number of DIB
	d_3	Total tray number of purge column
	d_4	Heat exchanger size (ft ²)
Process	x_1	Mole fraction of nC_4 at top of DIB
	x_2	Mole fraction of iC_5 at top of purge column
	x_3	Reactor pressure
Manipulated	u_1	Reflux ratio of DIB column
	u_2	Vapor ratio of DIB column
	u_3	Reflux ratio of purge column
	u_4	Input temperature of reactor

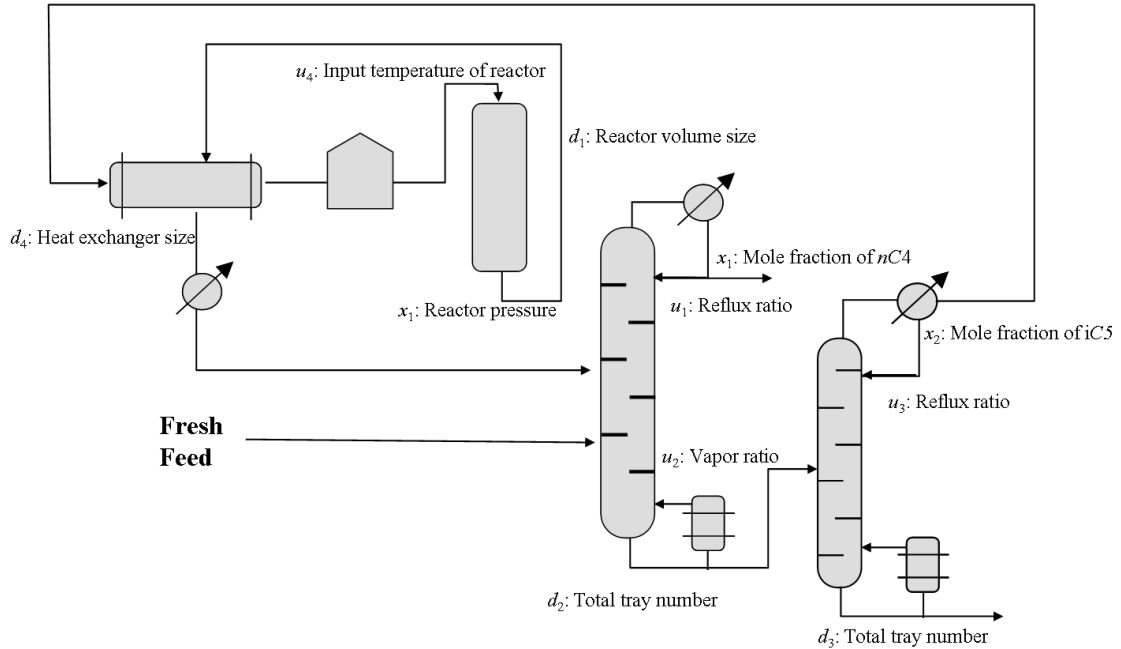


Figure 27: Design, process and manipulated variables in isomerization process case study.

4 Design variables

The design optimization features four key design variables, as shown in Table 11 and Figure 27: the reactor size (d_1), the total tray number of DIB (d_2), the total tray number of the purge column (d_3) and the heat exchanger size (d_4). The conversion ratio in the reactor is almost linearly proportional to the reactor size (d_1) as shown in Figure 28. Since a high conversion ratio decreases the amount of normal butane in the recycle stream, it also decreases the amount of normal butane in the top of DIB. As a result, a larger reactor size eases the product separation. Increasing the number of DIB trays improves the separation. Also, a larger number of trays in the purge column purges more isopentane. However, larger columns increase the total capital cost, so there is a tradeoff between controllability and capital cost. An economic tradeoff between the operating cost and the capital cost, with respect to heat exchanger size (d_4), also should be considered. A larger heat exchanger may reduce the energy needed in the furnace to keep the input temperature at a desired value. Again, larger heat exchanger areas need more capital cost.

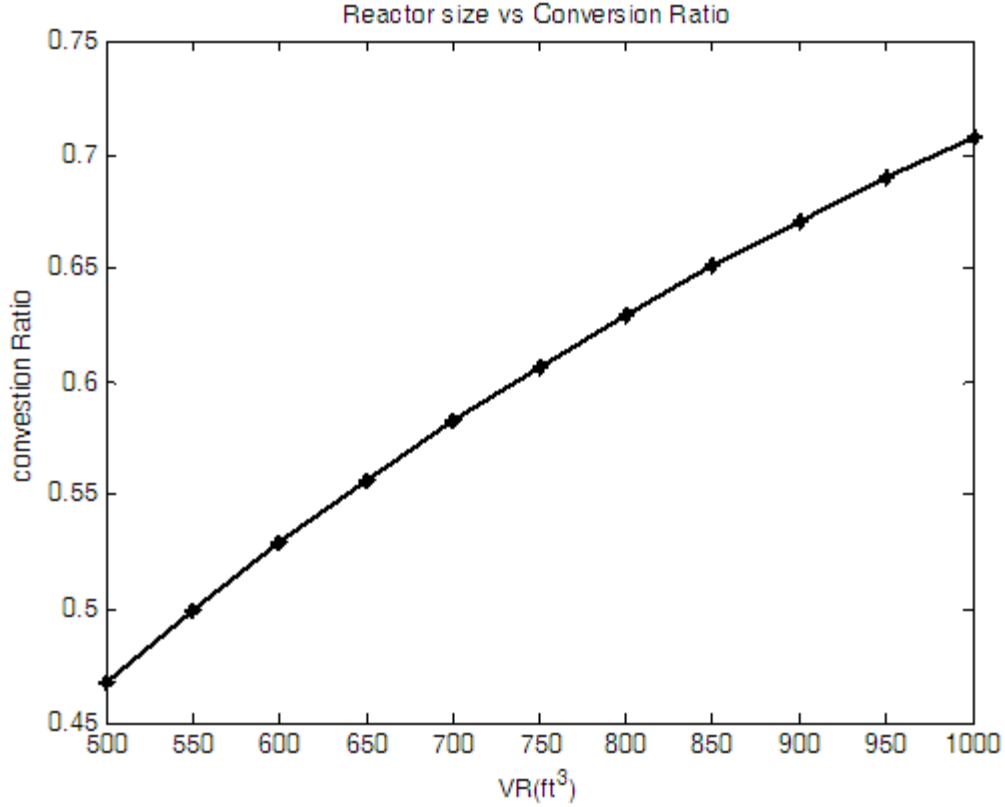


Figure 28: Reactor size vs conversion ratio. Conversion ratio is almost linearly proportional to the reactor size. This means a larger reactor will increase the controllability.

5 Uncertainty scenarios

To account for uncertainty in the real process operations, several expected uncertainty scenarios are considered for the design and control integration. We chose two scenarios: an instantaneous increase of the feed rate at certain time (t) and a dynamics composition change of nC_4 and iC_4 in input feed (w_1), as shown in Table 12. In Case 1, we doubled the feed rate at a certain time ($t = 1500$). In Case 2, we changed the composition of nC_4 and iC_4 in the input feed. In Case 3, we applied two dynamic disturbance scenarios at $t = 1500$.

Table 12: Dynamic disturbance scenarios

	Disturbance type	$0 < t < 1500$	$1500 < t < 3500$
w_1	Feed rate change	Fi: 578 lb-mol/hr	Fi: 1,156 lb-mol/hr
w_2	Feed composition change	nC_3 : 0.02 iC_4 : 0.24 nC_4 : 0.69 iC_5 : 0.05	nC_3 : 0.02 iC_4 : 0.04 nC_4 : 0.89 iC_5 : 0.05

For time-invariant uncertainty, we wish to investigate the impact of two main uncertain parameters associated with chemical reactions. The first parameter is the pre-exponential factor k_o , and the second is the heat of reaction, λ . Their nominal values and variances are illustrated in Table 13. We assume that 10% of variation exists from nominal points.

Table 13: Nominal values and expected deviation of uncertain parameters

	θ^N	$\Delta \theta^\pm$	variance
k_o (hr^{-1})	4×10^8	4×10^7	10%
λ (Btu/lbmol)	-3600	360	10%

6 Control Structures

Because this is a 3 by 4 system, we may consider pairing between process variables and manipulated variables. However, as shown in Figure 29, all process variables and manipulated variables are highly correlated, so it is not obvious how to pair variables. For example, x_1 is correlated with u_1 and u_2 . If the reflux ratio of the DIB column is increased, the impurity of product (x_1 , which represents the composition ratio of nC_4) will be decreased. If we increase the vapor ratio of the DIB column, x_1 also will be increased. In addition, x_1 is also highly correlated with u_4 (the input feed temperature). As u_4 is increased, the conversion ratio also is increased, which is helpful for separation. However, increasing u_4 may lead to higher pressure in the reactor, which violates safety constraint (34).

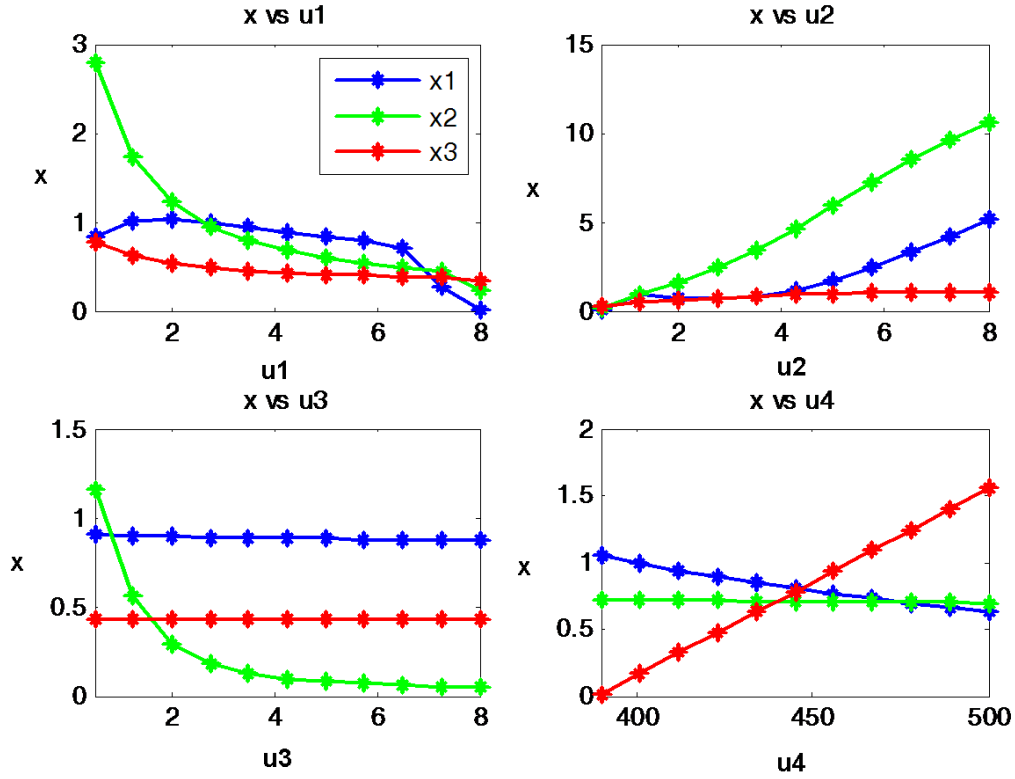


Figure 29: Relationship between x and u . All variables are highly correlated.

7 Performance of a Nominal Process

We computed two control strategies. First, we paired process variables with manipulated variables based on unit control. Hence, x_1 is paired with u_1 and u_2 , x_2 is paired with u_3 , and finally x_3 is paired with u_4 , as shown in Figure 30. Then, we tested this paired control structure and integrated structure with specific process designs ($d_1 = 550$, $d_2 = 40$, $d_3 = 20$, $d_4 = 10$). We simulated for 2500 discretized times and an activated controller after $t = 1000$. As shown in Figure 31, the classical control configuration failed to meet the requirement of the constraints after $t = 2500$. This controller did not use u_4 to satisfy

constraints (32) and (33). On the other hand, integrated control configuration succeeded in keeping all process variables under their set points after $t = 2500$, as shown in Figure 32.

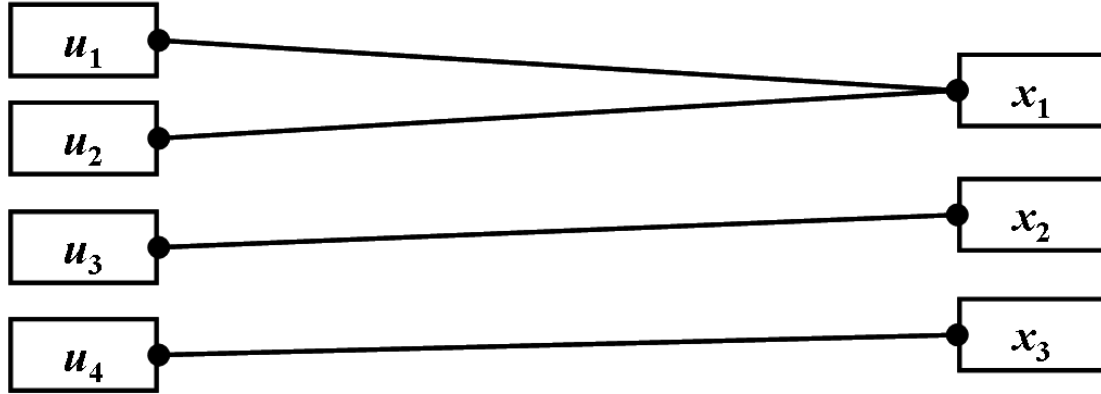


Figure 30: Classical control structure u_1 (reflux ratio of DIB) and u_2 (vapor ratio of DIB) are paired with x_1 (the mole fraction of nC_4 at the top of the DIB), u_1 (the reflux ratio of the DIB); u_3 (the vapor ratio of the purge column) is paired with x_2 (the mole fraction of nC_4 at the top of the DIB), and u_4 (the input temperature of the reactor) is paired with x_3 (the reactor pressure).

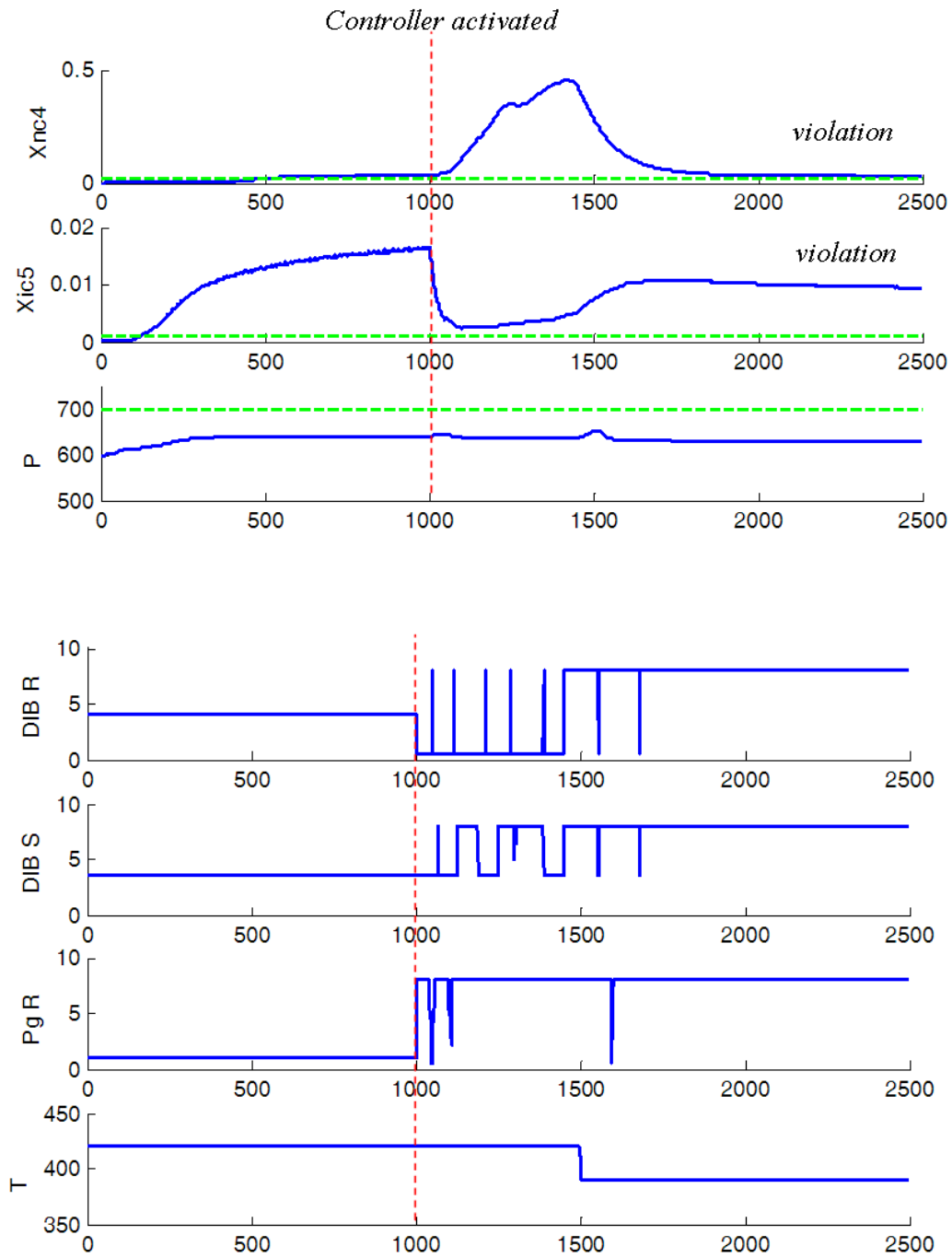


Figure 31: Result of the classical control structure. The classical control structure could not keep x_1 and x_2 under the set points until $t = 2500$.

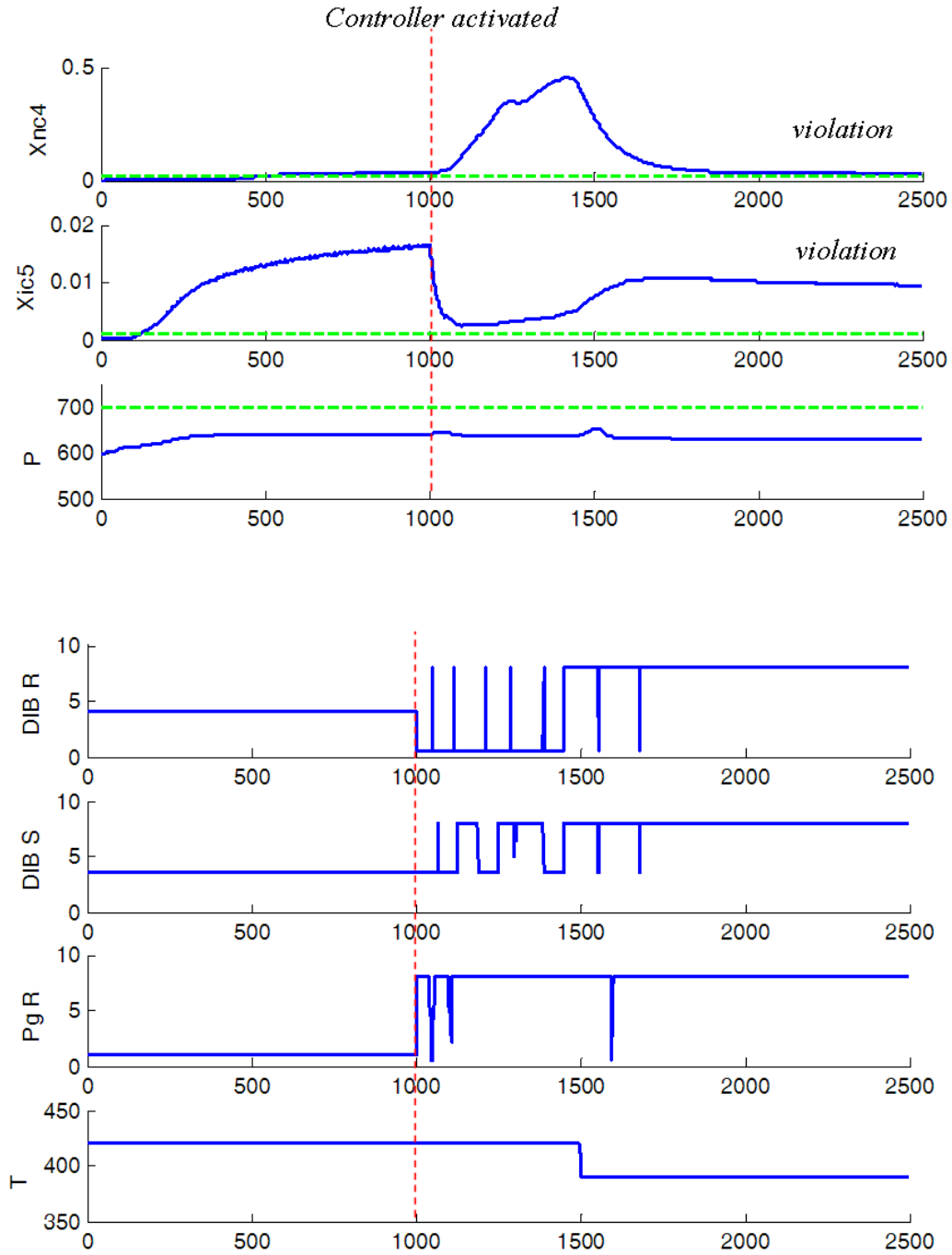


Figure 32: Result of the integrated control structure. It succeeded in keeping all process variables under the set points. Unlike the classical control structure, It used u_4 to control x_1 and x_2 .

8 Optimal design process

To maximize the performance, while, at the same time, planning flexible operation in expected scenarios, we performed the design optimization under uncertainty as follows. First, we collected uncertain samples (step A) and performed embedded control optimization (step B), in which the optimal control decision is made. Finally, we checked the feasibility of the design obtained in step B (step C).

As a first attempt to perform the stochastic optimization, we chose 10 samples in the uncertain space of reaction conditions using Latin hypercube method and evaluated the probabilities of each parameter set to calculate the expected cost, as shown in Figure 33.

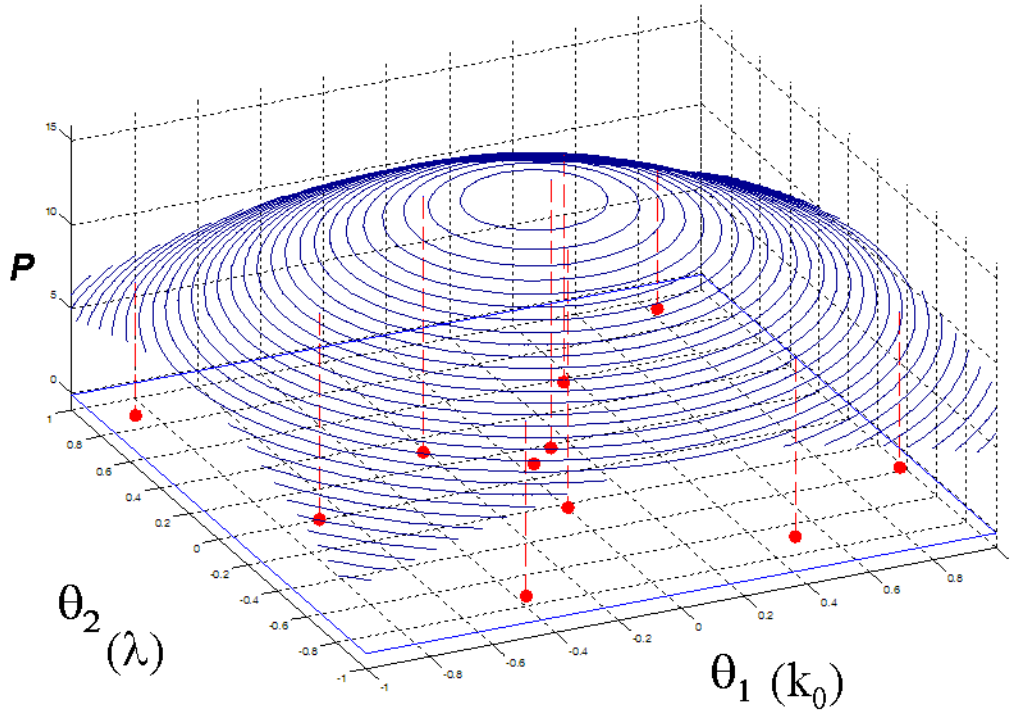


Figure 33: Probability of two uncertain parameters. We chose 10 samples using Latin Hypercube method.

We now wish to rigorously determine the design and manipulated variables, such that the process does not violate the constraint limits in any realization of the reaction conditions and dynamic performance due to uncertainty. For the capital cost, we considered the reactor, the columns, and the heat exchanger. For the expected operating cost, we considered energy consumptions in the furnace, the condensers, and the reboilers. The master level of this problem is to minimize the total annual cost.

$$\min_{d, c, x(t)} \Gamma = \underbrace{\int_{t=0}^{t_{\max}} \int_{s \in \Omega} \omega^s \cdot C_1(d, c, x(t), \theta^s, t) ds dt}_{\text{Expected Operating cost}} + \underbrace{C_2(d, c)}_{\text{Capital Cost}} + \text{penalty} \quad \text{Minimize Total Expected Cost} \quad (36)$$

$$s.t. \quad h_c(d, \dot{x}(t), x(t), \theta^s, t) = 0, \quad \forall s \in \Omega \quad \text{Conservation Laws} \quad (37)$$

$$g_j^{point}(\mathbf{d}, \mathbf{x}(t), \mathbf{u}(t), \theta^s, t) \leq 0, \quad t \in [H_1, H_2] \quad \text{point Constraints} \quad (38)$$

$$g_j^{path}(\mathbf{d}, \mathbf{x}(t), \mathbf{u}(t), \theta^s, t) \leq 0 \quad t \in [0, H_{end}] \quad \text{Path constraints} \quad (39)$$

Since constraints (32) and (33) are soft constraints, they do not need to be satisfied in all time periods. Thus, we implemented them as point constraints, to be satisfied at certain time instances during the period of operation, as defined in (38). In this case, values of $H_1 = 3250$ and $H_2 = 3500$ are used. On the other hand, constraint (34) is a path constraint, as defined in (39), since it should be satisfied for all simulation times.

To solve optimal design problem, a Nelder-Mead simplex method was used in the master level of our methodology, and the embedded control approach was used to adjust the optimal control decisions. For numerical solution in suboptimization, *control vector parameterization* (Feehery & Barton, 1998) was adopted. To handle the inequality constraints, a penalty is added to the objective function.

Next, a rigorous feasibility test of the current best design specification was performed. Initially, we performed a steady state feasibility test. A system that is infeasible

in steady state is not feasible in dynamic state either. If we find a critical point in steady-state feasibility test, then we return to the main optimization problem (step B) without doing a dynamic feasibility test. However, steady state feasibility does not guarantee all trajectories satisfy constraints. Therefore, we also performed dynamic feasibility test in which all the trajectories of the current design were checked to be feasible under steady state conditions.

For the feasibility test in this study, *Active Constraint Strategy* was adapted (Dimitriadis & Pistikopoulos, 1995; Grossmann & Floudas, 1987). A mathematical formulation for Active Constraint Strategy is given in (40).

$$\begin{aligned}
& \max \delta & (40) \\
& s.t. \quad h^i(\mathbf{d}, \mathbf{x}, \mathbf{u}, \boldsymbol{\theta}, t) = 0 \\
& \quad s^j + g^j(\mathbf{d}, \mathbf{x}, \mathbf{u}, \boldsymbol{\theta}, t) - \delta = 0 \\
& \quad \sum_i \mu^i \frac{\partial h^i}{\partial \mathbf{x}} + \sum_j \lambda^j \frac{\partial g^j}{\partial \mathbf{x}} = 0 \\
& \quad \sum_i \mu^i \frac{\partial h^i}{\partial \mathbf{u}} + \sum_j \lambda^j \frac{\partial g^j}{\partial \mathbf{u}} = 0 \\
& \quad \sum_j \lambda^j = 1, \quad \sum_j y^j = \dim\{\mathbf{u}\} + 1 \\
& \quad \lambda^j - y^j \leq 0, \quad s^j - U(1 - y^j) \leq 0 \\
& \quad \boldsymbol{\theta}^L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^U \\
& \quad y^j \in \{0, 1\}, \lambda^j, s^j \geq 0, i \in I, j \in J
\end{aligned}$$

where \mathbf{d} is the vector of design variables, \mathbf{x} the vector of state variables, \mathbf{u} the vector of manipulated variables and $\boldsymbol{\theta}$ the vector of uncertain parameters. Here s^j is are slack variables and μ^j and λ^j are lagrange multipliers. $i \in I$ and $j \in J$ are the index sets for equality constraints and inequality constraints, respectively.

To handle differential equations, we used a collocation method to represent differential equations by polynomial approximation (Dimitriadis & Pistikopoulos, 1995).

Also, we simplified the dynamic feasibility by examining only weights at time nodes instead of examining the course of all trajectories between collocation nodes. We admit that there could be critical points between time nodes that violate dynamic constraints. This exercise was beyond the range of the current work.

9 Results

Case 1: Design and Control Optimization with Feed Rate Change. The first case considers the increase of feed rate in fresh feed (F_i). We optimized the design under the dynamic disturbance w_1 . The initial guess for the case was $d_1=3000$, $d_2=70$, $d_3=40$, $d_4=1200$. We optimized with 10 time-invariant uncertain samples. In the first trial, we obtained the design specification $d_1=285$, $d_2=89$, $d_3=35$, $d_4=1539$. With the steady feasibility test, a critical point ($k_0 = 3.6 \times 10^8 \text{ hr}^{-1}$, $\lambda = -3960 \text{ Btu/lb-mol}$) was identified. Since the steady feasibility test yielded a critical point, the dynamic flexibility test had not been tested yet. Thus, we optimized design and control decision again with 10 original uncertain scenarios augmented by the new critical scenario. A second improved design was found to be $d_1=334$, $d_2=94$, $d_3=46$, $d_4=1585$, which is a more conservative design than the first one. With this design specification, in both steady-state feasibility tests and dynamic feasibility tests, no critical points were identified. The performance of the final design specification with the critical uncertain parameter point is shown in Figure 34. This simulation result is composed of three phases. We started the process at $t = 0$. As an initial point, we assume $x_1 = 0$, $x_2 = 0$, $x_3 = 600$. In the first 200 time steps, the controller was not switched on yet, so that sufficient data could be gathered for the process identification (phase I). At $t = 60$, a large system deviation occurred in the x_2 composition, as indicated in Figure 34, because the system identification had not stabilized yet. After $t = 200$, the control is activated and

x_2 is decreased by adjusting manipulated variables (phase II). When the disturbance w_1 hits the process at $t = 1500$, x_1 is increased, but without exceeding the upper limit (phase III). This scenario has no further challenges to remain within desired operational limits.

Case 2: Design and Control Optimization with Feed Composition Change. We performed optimization with disturbance scenario w_2 in Case 2. In this case, we initially obtained the optimal design $d_1 = 184$, $d_2 = 89$, $d_3 = 46$, $d_4 = 1545$ with 10 uncertain scenarios. In steady feasibility tests, we found a critical point ($k_0 = 3.6 \times 10^8 \text{ hr}^{-1}$, $\lambda = -3960 \text{ Btu/lb-mol}$). Accordingly, the design problem was solved again with this critical point incorporated in the scenario samples. The second optimization yielded a design with $d_1 = 193$, $d_2 = 93$, $d_3 = 46$, $d_4 = 1540$, which is less expensive than that in the flowsheet of Case 1. As shown in Figure 35, the disturbance w_2 increased the reactor pressure and the embedded control system adjusted the manipulated variables to keep all process variables below upper limits. In phase II, x_2 is increased during the startup process, which is similar to that observed for Case 1. Like Case 1, the controller adjusted the manipulated variables and kept all process variables within the desired limits. In phase III, the dynamic disturbance w_2 increased all process variables, as well as the product impurity (x_1) and the reactor pressure (x_3); however, these will meet, causing unacceptable constraint violations.

Case 3: Integrated Design and Control with Feed Flow Rate and Feed Composition Change. In this case, we performed integrated design and control optimization with both disturbance scenarios w_1 and w_2 . In the first iteration, the optimal flowsheet was $d_1 = 407$, $d_2 = 88$, $d_3 = 46$, $d_4 = 1542$. The dynamic feasibility test identified the critical point ($k_0 = 3.6 \times 10^8 \text{ hr}^{-1}$, $\lambda = -3960 \text{ Btu/lb-mol}$). Then, the design optimization is run again, with a final result of $d_1 = 470$, $d_2 = 90$, $d_3 = 46$, $d_4 = 1533$. This allows no phase

to have disturbances and no control (see Figure 36). After $t = 200$, the controller is engaged and keeps all variables below the upper limits. In the second phase, x_1 and x_3 were increased and exceeded the upper limits when two dynamic disturbances occurred.

All results are summarized in Table 14. The result of Case 3 is the most expensive, because it accommodates both disturbances simultaneously. The parameter values in steady state of optimal design of each case study are summarized in Table 15.

Table 14: Best design with ten original samples and a critical point

Case no		d_1	d_2	d_3	d_4	C_1	C_2	TAC
1	1 st	285	89	35	1,539	656	3,213	1,727
	2 nd	334	94	46	1,585	503	3,507	1,672
2	1 st	184	89	46	1,545	356	2,628	1,232
	2 nd	193	93	46	1,540	329	2,691	1,226
3	1 st	407	88	46	1,541	904,	3,861	2,191
	2 nd	470	90	46	1,533	659	4,184	2,054

C_1 : operating cost (k\$/year), C_2 : capital cost (k\$), TAC: total annual cost (k\$) $C_1+C_2/3$,

Table 15: Parameter values in steady states

Unit	Parameter	Case1		Case 2		Case 3	
		t<1500	t>1500	t<1500	t>1500	t<1500	t>1500
Reactor	Volume ft ³	334	334	193	193	470	470
	Conversion Ratio	0.94	0.78	0.68	0.72	0.98	0.88
	Input Temp.	405	404	432	419	390	396
	Output Temp	459	463	465	469	446	471
DIB	Total trays	94	94	93	93	90	90
	Feed Trays	26/56	26/56	37/56	37/56	36/54	36/54
	Reflux ratio	7.9	8.0	7.9	8.0	7.1	8.0
	Vapor ratio	7.7	7.7	4.5	5.1	7.6	7.7
	Reflux drum hold up, ft ³	850	850	850	850	850	850
	Base hold up, ft ³	1000	1000	1000	1000	1000	1000
Purge column	Total trays	46	46	46	46	46	46
	Feed Trays	23	23	23	23	23	23
	Reflux ratio	7.99	7.98	6.3	5.1	7.3	7.7
	Vapor ratio	170	170	139	115	158	167
	Reflux drum hold up ft ³	185	185	185	185	185	185
	Base hold up ft ³	200	200	200	200	200	200
FEHE	Area, ft ²	1585	1585	1540	1540	1533	1533
	Input temp. of hot stream, K	459	463	465	469	445	471
	Output temp. of hot stream, K	222	280	260	267	215	288
	Input temp. of cold stream, K	115	115	115	115	115	115
	Output temp. of cold stream, K	298	315	313	324	298	345

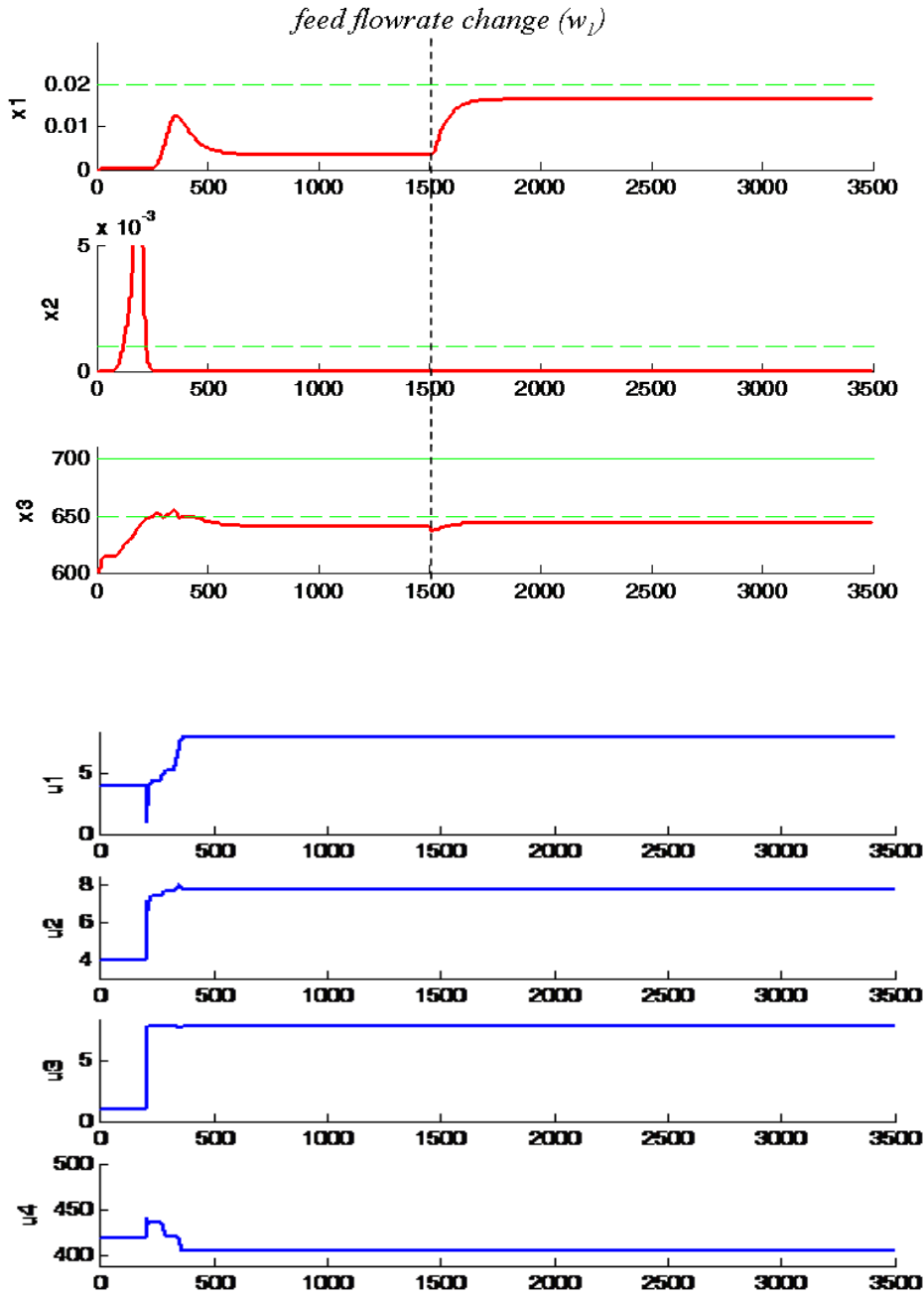


Figure 34: Simulation result of the final optimal design (Case 1): (a) trajectories of process variables x_1 - x_3 and (b) trajectories of manipulated variables u_1 - u_4 . In the identification phase (phase I), the controller is not activated, because sufficient data have not been collected for the moving horizon estimator. Phase II describes startup behavior. The controller kept all process variables under the upper limits. When a dynamic disturbance occurred, all values of process variables are increased. However, they did not exceed the upper limits but reached a new steady state.

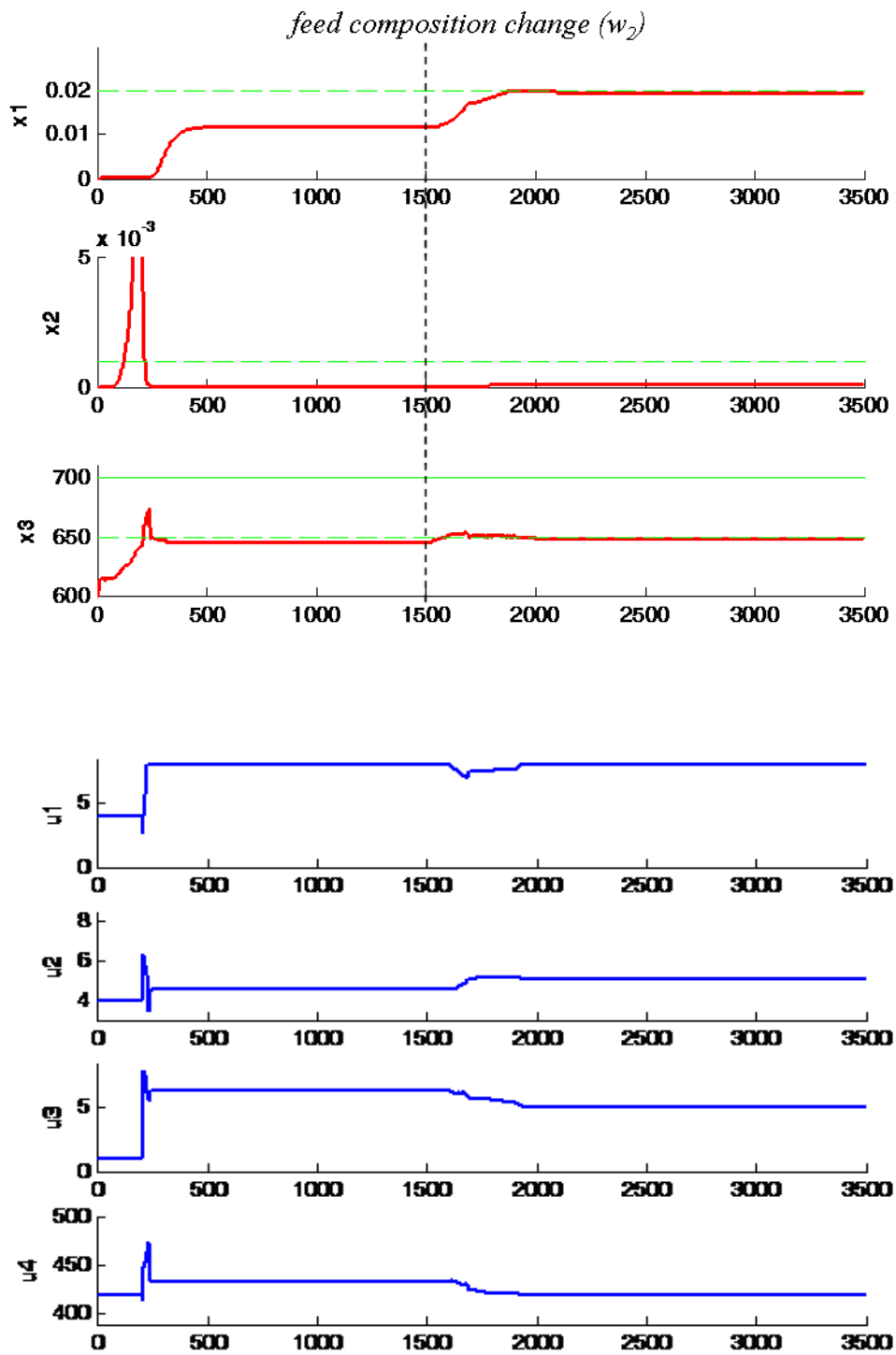


Figure 35: Simulation result of the final optimal design (Case 2). A feed composition change occurred at $t = 1500$, but the response of the embedded controller was satisfactory.

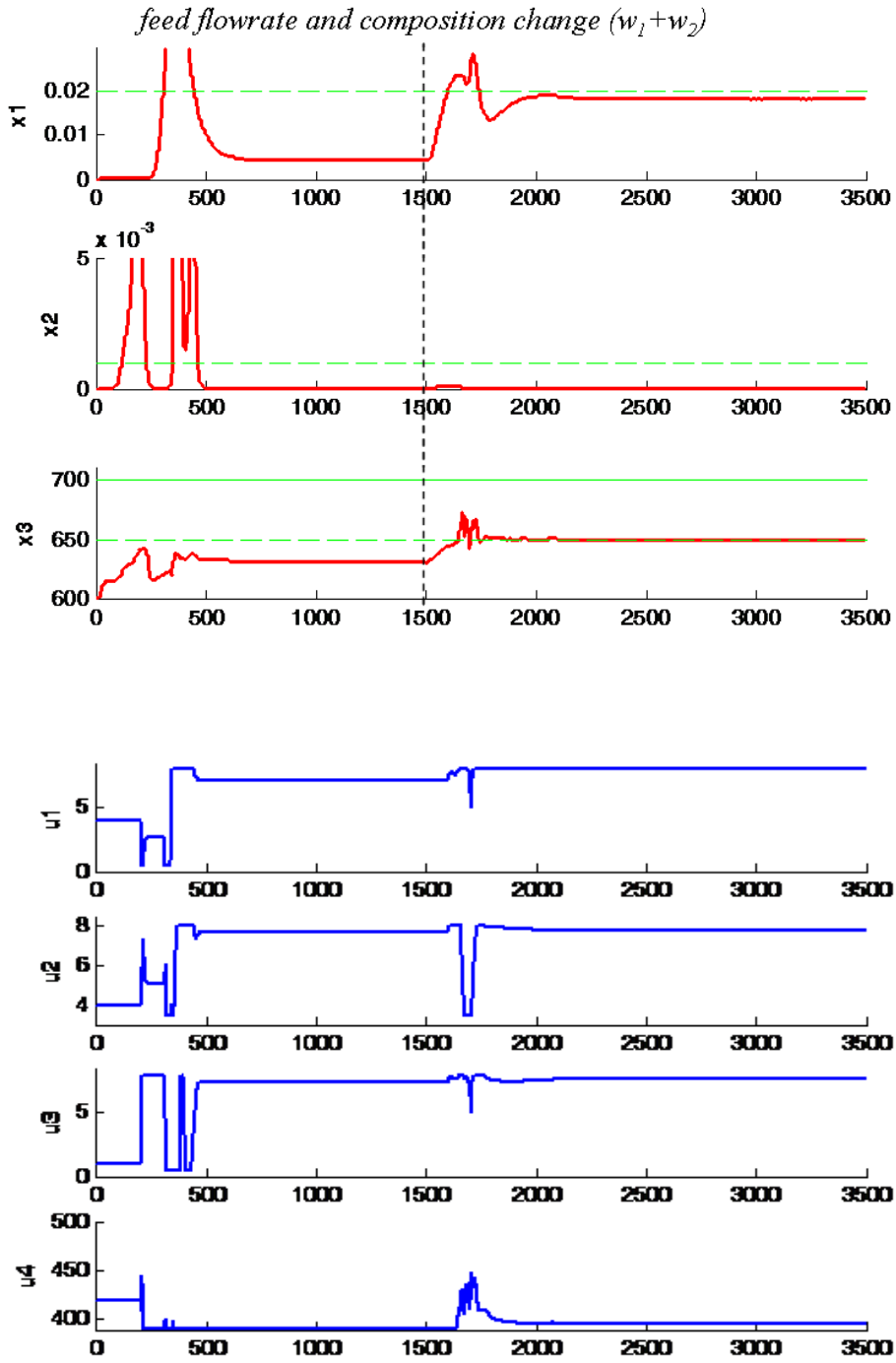


Figure 36: Simulation result of the final optimal design (Case 3). In this case, two dynamic disturbances occurred at $t=1500$, which caused increases in x_1 and x_3 increased. However, the embedded controller adjusted the manipulated variables to keep all process values below the upper limits.

10 Discussion

In this chapter we performed design optimization considering controllability with dynamic disturbance scenarios. The optimal design and control decisions obtained in the previous section were able to reject the expected dynamic disturbances satisfactorily.

In Case 1, disturbance w_1 -doubling the feed rate-decreased the conversion ratio, which resulted in an increase in x_1 . However, the optimal design with embedded control was robust enough to reject this disturbance. Even though x_1 was increased, its value did not exceed the upper limit.

Case 2 considered dynamic disturbance w_2 - feed composition change. Because the mole fraction of nC_4 in the input feed was increased in the disturbance scenario, it made the separation more difficult. As expected, the disturbance increased the mole fraction of nC_4 of the final product (x_1), as well as the top stream of the purge column (x_2) and brought the quality violation of constraint (33). However, the final optimal design chosen by embedded control optimization strategy had good controllability to accommodate this violation.

In Case 3, the design should be robust enough to safely operate for both disturbances at $t= 1500$. The trajectories of the process in Case 3 reflect tougher dynamic control challenges than observed for Case 1 and Case 2. To reject two disturbances, the final design of Case 3 required a larger size than that needed for Case 1 or Case 2. We found the same critical point ($k_0= 3.6 \times 10^8 \text{ hr}^{-1}$, $\lambda = -3960 \text{ Btu/lb-mol}$) in all cases. This point is located on the vertex of uncertain space. This point can be reasonably considered to be the worst-case scenario, because small k_0 makes the conversion ratio small and a small heat of reaction

(large absolute value) increases the reactor temperature, which results in an increase in pressure.

Our methodology is to recast an integrated design and control examined. Also, we wish to improve the quality of identification for highly nonlinear processes, using more-advanced identification, such as the subspace identification method or nonlinear model predictive control. However, these advanced algorithms are more expensive; therefore, a tradeoff between accuracy and performance of algorithms needs to be considered.

Nomenclature (Chapter III)

A	=	parameter matrix in state space model
B	=	parameter matrix in state space model
c	=	control decision
C_1	=	operating cost
C_2	=	capital cost
d	=	design decisions
d_i		design variables
E	=	activation energy, Btu/lb-mol
F_i	=	fresh flow rate, lb-mol/hr
F_R	=	Recycle flow rate , lb-mol/hr
H	=	moving horizon time
k		temperature dependent rate constants, hr ⁻¹
ko	=	preexponential factor (hr ⁻¹)
Q	=	covariance matrix
r	=	reaction rate, lb-mol / hr ft ³
R	=	covariance matrix
t	=	time, m
T	=	reactor temperature, K
K^*	=	control gain
U	=	overall heat transfer coefficient, Btu/ft ³ ·h·K
V	=	the reactor volume, ft ³
w_i		disturbance scenarios

X = state variable

Greek letters

δ = flexibility index

θ = static parametric uncertainty

ξ = time dependent uncertainty source or disturbance

Ω = uncertain space

λ = heat of reaction, Btu/lb-mol

PART C: Global Optimization for Finding Multiple Solutions

Investigation of multiple solutions in global optimization problems is important because multimodal objective functions are common in engineering and operational research. Several deterministic methods were developed for finding multiple solutions. In stochastic optimization technique fields, several genetic algorithms called ‘niche technique’ were developed for the multimodal situation. They were successful in some cases by maintaining the diversity of certain properties within the population, but still have some limitations. To tackle these problems, a ‘hybrid sequential niche algorithm’ is proposed in this part. This is based on sequential niche technique suggested by Beasley et al (1993). Like Beasley’s method, it finds all solutions sequentially. But our algorithm deploys a deterministic local search to accelerate speed, therefore ours is faster than the traditional niche methods. In addition, its robustness is enhanced because it uses variable niche size. It adaptively sets niche size for each solution, so it is more robust to find multiple optima even though the solutions are not evenly distributed. This algorithm was tested with several challengeable benchmark functions. Also it was used for finding multiple, optimal parameter sets in designing distributed systems.

Part C consists of two chapters. Chapter IV overviews global optimization, and reviews several deterministic and stochastic optimization methods. Chapter V presents details of hybrid sequential niche algorithm and several case studies.

CHAPTER IV. REVIEW OF GLOBAL OPTIMIZATION

Previously published as:

Moon, J., and Linninger, A.A., “A Hybrid Sequential Niche Algorithm for Optimal Engineering Design with Solution Multiplicity”, *Computers and Chemical Engineering*, 33 (7), 1261-1271, 2009.

Global optimization is an area of mathematics that involves the search for solutions to equation systems or maximum/minimum of a given objective function according to some conditions. Global optimization is distinguished from regular optimization (called *local optimization*). While the regular optimization focuses on finding the maximum or minimum in nearby local area, global optimization pursues to find maximum or minimum through the whole boundary. As shown in Figure 37, the global maximum is the highest location on the boundary, whereas the other two maxima are the local maxima. This is the main difference between a global maximum and local extrema. Global maximum is a point where its function values achieves the greatest in the boundary. Local maximum is area that is the highest value when compared to only its neighboring points. Global optimization is an important issue in engineering and science fields, because there are many optimization problems local optimization approaches cannot find global solution. Thus, the area of global optimization has been much paid attention in the last two decades.

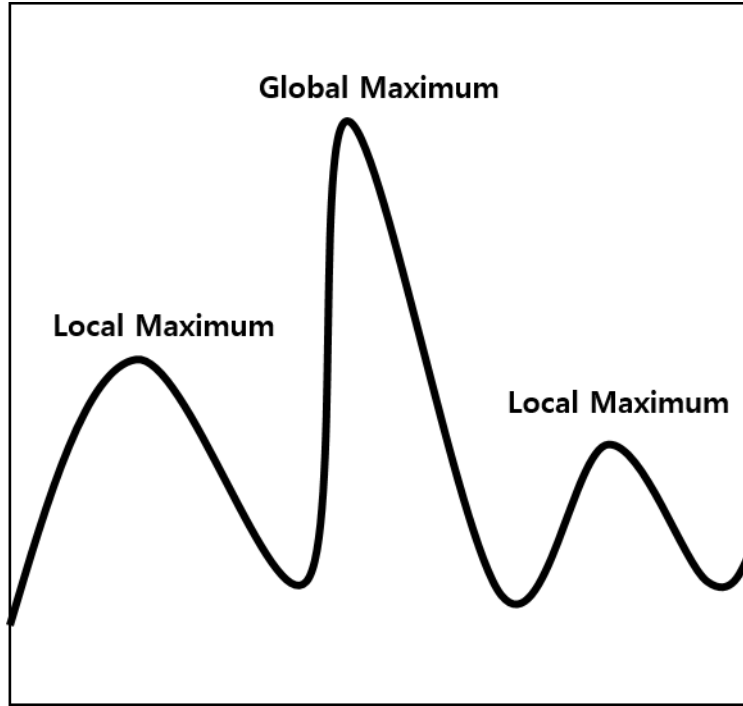


Figure 37: Example of a global maximum near local maximum locations.

Existing global optimization algorithms can be categorized into *deterministic method* and *stochastic method*. Deterministic method is a mathematical technique based on the concept that the next step can be determined precisely from the past behavior of a set of data. The behavior by the method is determined by initial state and inputs and never produce different result as long as inputs are same. Deterministic method includes trajectory methods that travel along the function and check for the global optimum, and also branch and bound methods. Each branch is checked for a global extremum, and then it is further branched until the global extremum is located. Stochastic methods incorporate with random numbers to predict the next step. It searches a set of points randomly, and choose some of points that are superior than others. Then the chosen is next step and this procedure is repeated until it finds optimum. Since the method uses random number in the algorithm, it present different behavior whenever it runs. Techniques for global

optimization can include characteristics of deterministic or stochastic methods individually, or a combination of both. While deterministic approaches guarantee ε -global minimum in mathematical sense, stochastic methods guarantee asymptotic convergence guarantees only at infinity for optimization problems.

1 Deterministic optimization methods

Several global optimization techniques have already been extensively researched, and are fairly successful at obtaining the global extremum for a series of functions. The examples of deterministic methods are the interval method, (Hansen, 1979); branch and bound procedures (Al-Khayyal and Falk, 1983; Horst and Tuy, 1987); cutting plane methods (Tuy et al. 1985); outer approximation approaches (Horst, 1992); Lipschitzian methods, (Hansen et al, 1992a,1993b); reformulation-linearization (Serali and Alameddine, 1992; Serali and C.H. Tuncbilek, 1992); the α BB method (Floudas, 2000); the homotopy method (Dunlavy & Leary, 2005), and global terrain methods (Lucia and Yang, 2002).

Among these deterministic methods, several deterministic methods-interval methods, homotopy method, α BB method, and BARON are reviewed in this section.

1.1 Interval method (Hansen, 1979)

The interval method finds a global extremum by producing bounds on the objective function. Then it divides the objective function into separate intervals and determines if the global extremum is located on that interval, and if is not located, it removes the interval from the domain. The method repeats this process until the interval is small enough. The bounds are based on the range of the objective function in the domain, as computed with interval arithmetic. Thus these bound may be too loose.

1.2 Homotopy method

Homotopy method converts an original problem into a simpler one that is easy to solve. Then it gradually deforms this simpler problem into the original and find the actual solution from the solution of simpler one already found, as a starting point, lastly it ends with the solution of the original problem. The homotopy-continuation method provides a smooth transition from an approximation to true solution(s) of $W(x) = 0$, using of a scalar homotopy parameter, t . The Newton homotopy has the form:

$$H(x, t) = W(x) - (1 - t)W(x_0) = 0 \quad (41)$$

Starting at $t = 0$, $H(x, 0) = 0$ is trivial to solve given any initial vector, x_0 . A homotopy path is generated as t increases to unity, where the true solutions are located. To track the homotopy paths, a predictor-corrector procedure is used. It determines if a local extremum lies in that section of the function, and if not, removes it from further consideration. If a local extremum is found, it checks to see if it is less than the current global extremum and then updates the global extremum value accordingly.

1.3 α BB method

α BB method was proposed for locating global minimum solution of constrained nonconvex problems by Floudas group (Androulakis et al., 1995; Adjiman et al, 1996). This is based on a branch and bound framework (Horst and Tuy, 1990) and converging lower and upper bounds. It deals with nonconvex minimization problem and guarantees global optimality mathematically. Lower bounds are obtained through the solution of convex programming problems and upper bounds based on solving nonconvex problem with local methods. The α BB method adapts an α parameter to assist in the convergence of functions. The method begins with the original problem with the complete feasible region

called the root problem. The lower-bounding and upper bounding procedures are applied to the root problem. If the upper and lower bounds meet, then an optimal solution is found and ends the whole procedures. If not, the feasible region is divided into two and these subproblems become children of the root search node. The algorithm is used recursively to the subproblems, producing a tree of subproblems. If an optimal solution is found in the subproblem, the rest of the tree is removed: if the lower bound for a node exceeds the best known feasible solution, global solution does not exist in the subspace. Therefore, the node is out from consideration and search for another node is implemented. This search procedure is performed repeatedly until all nodes are searched or removed.

1.4 Branch and reduced algorithm (BARON)

The *Branch And Reduce Optimization Navigator* (BARON) is another global optimization based on branch-and-bound technique (Sahinidis, 1996). BARON incorporates conventional branch and bound with range reduction techniques for tightening lower bounds. These reduction techniques are used to subproblems of the search tree in pre- and post-processing steps to reduce the search space and relaxation gap. BARON is hired as the part of a commercial software -GAMS.

1.5 Global terrain method

Global terrain method is a set of algorithms to find all solutions and saddle points based on moving along curves of the gradient vector field to both uphill and downhill movement. (Lucia and Feng, 2002). This methodology is an overall philosophy for moving from one stationary point to another that requires reliable local equation solving tools as well as reliable and efficient uphill exploration. Theoretical foundation for this work rests on the fundamental observations that (1) solutions and singular points are smoothly connected when the model functions are smooth; (2) valleys, ridges, ledges, etc. provide a

natural characterization of this connectedness; (3) valleys, ridges, etc. can, in turn, be characterized as a collection of constrained extrema over a set of level curves; (4) there is an equivalent characterization of valleys, ridges, etc. as solutions to generalized, constrained eigenvalue–eigenvector problems; and (5) the natural flow of Newton-like vector fields tends to be along these distinct features of the landscape.

The global terrain method consists of a series of downhill, equation-solving computations and uphill, predictor–corrector calculations. Downhill movement to either a singular point or solution is conducted using reliable, norm-reducing (complex domain) trust region methods. Uphill movement, on the other hand, is necessarily to a singular point and uses approximate uphill Newton-like predictor steps combined with intermittent corrector steps. Each corrector step is defined by calculating an extremum in the gradient norm on the current level set for the least-squares function can be shown to be equivalent to a solution to a generalized, constrained eigenvalue–eigenvector problem and helps ensure that valleys and ridges are tracked as closely as desired. Initial starting points are arbitrary, while starting points for subsequent subproblems defining movement from one stationary point to another are along appropriately determined eigendirections, since valleys and ridges are generalized eigenpathways. Collisions with boundaries of the feasible region and the presence of points at infinity are also addressed and the heuristic termination criterion based on the concept of limited connectedness is presented. Details of this methodology and the result of test runs are described in APPENDIX B.

2 Stochastic methods

This section addresses the review of stochastic optimization methods. Examples of stochastic methods are genetic algorithms (Goldberg, 1989); the scatter search approach

(Egea et al.,2007); simulated annealing (Kirkpatrick and Vecchi, 1983); and Particle swarm optimization (Edward and Kennedy, 1995).

2.1 Genetic Algorithms

The Genetic Algorithms are optimization methods which mimic behavior the processes of evolution in nature. This algorithm is based on randomized operators, such as selection, cross-over and mutation, inspired by the natural reproduction and evolution of the living creatures. It was developed by Holland (1975) and finally popularized by Goldberg (1989).

The Genetic algorithms perform following steps:

- Step 1-Start: Generate an initial population randomly.
- Step 2-Fitness: Evaluate the fitness for each individual in the current population $M(t)$
- Step 3-Selection: Select some superior individuals from population.
- Step 4-Crossover: Produce offspring by selected individuals via genetic operators.
Offspring replace individuals unselected in Step 3.
- Step 5-Mutation: Mutate some individuals with a certain mutation probability.
- Step 6-Iteration: Repeat Step 2 to Step 5 until the convergence is obtained.

Genetic algorithms are simple and robust search algorithms so that it is good for discovering good solutions rapidly for difficult high-dimensional problems. GAs are useful and efficient when the search space is complex or hard to be understood; mathematical analysis or derivative information is unavailable. Also the algorithms are easily parallelized like other evolutionary algorithms, customized and hybridized. Details of hybrid genetic algorithms will be considered next chapter.

2.2 Scatter search

Scatter search is a population-based method for combinatorial and nonlinear optimization problems. Scatter search uses strategies for combining solution vectors that are effective in a variety of problem formulation. Glover (1977) introduced Scatter search as a heuristic for integer programming. This search is known as flexible, since each of its elements can be implemented in various ways and degrees of complexity. The basic design to implement scatter search comes from the well-known “five-method template” (Laguna and Martí, 2003). Since Scatter Search does not have only a single uniform design, many variations can be derived from the basic template. By Laguna and Marti 2003, The basic templates are as follows:

1. *A Diversification Generation Method: Generate a collection of diverse trial solutions randomly as an input.”*
2. *An Improvement Method: improve a trial solution to more enhanced solutions. Neither the input nor the output solutions are required to be feasible, though the output solutions is usually expected to be improved.*
3. *A Reference Set Update Method: Build a reference set consisting of the best solutions found organized to provide efficient accessing by other parts of the method.”*
4. *A Subset Generation Method : Operate on the reference set, to produce several subsets of its solutions to create combined solutions.*
5. *A Solution Combination Method : Transform the subset of solutions produced by the Subset Generation Method into more improved solution vectors.*

2.3 Simulated annealing

Simulated annealing, introduced by Kirkpatrick and Vecchi (1983), simulates the annealing process in which an element is heated above its melting temperature then gradually cooled to produce the crystalline lattice, minimizing the energy probability distribution. Fast cooling the liquid prevents the elements from crystallization and it becomes an amorphous mass with a higher than optimum energy state. The key of successful crystallization is how to control the rate of change of temperature properly.

This method starts with a random new variable set. Then it evaluates cost functions of the new variable set. If the sum of cost of new variables set is better than that of the old, the new set replaces the old set. If not, the new set is selected if the following condition is satisfied.

$$r \leq e^{[f(p_{old}) - f(p_{new})]/T} \quad (42)$$

where r is a uniform random number and T analogous to temperature.

Thus, the new variable set is accepted with a certain probability. The new variable set can be made by changing the values of the old with random number, as follows

$$p_{new} = dp_{old} \quad (43)$$

where d is either uniformly or normally distributed about p_{old} .

If the new variables sets do not produce lower costs than old one any more after a certain number of iterations is done, the values of T and d are decreased by a certain percent. When the value of T is almost zero, the iteration is finally stops and locates the solution. There are various ways of defining *cooling schedule*-how to decrease T . One of most representative examples of cooling schedules are as follows:

$$T_n = T_o - n(T_o - T_n) / N \quad \text{linear decreasing} \quad (44)$$

$$T_n = 0.99T_{n-1} \quad \text{Geometrically decreasing} \quad (45)$$

$$T_n = c / \log(1 + n) \quad \text{Hayjek optimal} \quad (46)$$

where T_n is the temperature of step n , T_o initial temperature, N : total number of steps

The traveling salesperson problem (Kirkpatrick, 1983) was solved with Simulated annealing, and the algorithm has been applied successfully to various problems.

2.4 Particle swarm optimization

Particle swarm optimization is “a swarm intelligent technique inspired by the social behavior of animals such as birds flocking and fish schooling” (Edward and Kennedy, 1995). Like Genetic Algorithms, Particle Swarm Algorithm starts with a random population, but the particle swarm optimization adapts *evolution operators* such as crossover or mutation.

Each particle (called individual in other algorithms), moves the search space with an its own velocity, and save the best position found by itself. Each particle is regarded as a point in a D-dimensional space that adjusts ‘flying’ according to its own and others’ flying experience. The particles adjust velocities and positions based on the local and global solutions as follows:

$$\begin{aligned} v^{new} &= wv^{old} + c_1 \times r_1 \times (p^{lb} - p^{old}) + c_2 \times r_2 \times (p^{gb} - p^{old}) \\ p^{new} &= p^{old} + v^{new} \end{aligned} \quad (47)$$

where w : inertia weight v : particle velocity, p : particle position; r_1, r_2 : independent uniform random numbers; c_1, c_2 : learning factors, p^{lb} : best local solution, p^{gb} : best global solution.

The new velocity is updated by the best global solution associated with the lowest cost ever found by a particle and the best local solution associated with the lowest cost in the current population. Particle swarm optimization has relatively fewer parameters to adjust than other evolutionary algorithms. Inertial weight w is used to control the impact of the previous history of velocities on the current one. It results in a reduction of the number of iterations if the proper value is given. Initially, it is large value and then gradually decreased. The parameter c_1 is called a cognitive parameter and the parameter c_2 is a social parameter. They are used to guarantee the convergence. Many studies have been done to find the best values of c_1 and c_2 . (Edward and Kennedy, 1995; Ozcan and Mohan, 1999; Ratnaweera and Halgamuge, 2004) While c_1 and c_2 are used for the convergence of the population, r_1 and r_2 are used to maintain the diversity of the population. They are random variables between 0 and 1.

Since its fast convergence and easy implementation, the particle swarm optimization is applied in many areas such as neural network, dynamic web organizing, fitness prediction, mountain clustering, and parameter selection.

3 Comparisons of global optimization methods

3.1 Deterministic vs stochastic methods

Comparison between deterministic and stochastic methods were performed in several literatures (Liberti & Kucherenko, 2005; Blake, 1989; Decker & Aarts, 1991; Dixon et al, 1975; Zabinsky, 1998). The main advantage of deterministic methods is, they provide a theoretical guarantee of locating global minimum, or at least a local minimum whose objective function differs by at worst ϵ from the global minimum for a given $\epsilon > 0$. Stochastic methods only offer the global optimality in probability. However,

stochastic methods adapt better to black-box formulations or extremely ill conditioned objective functions than deterministic methods, because they usually rely on function evaluations rather than gradient and Hessian information, most of deterministic methods require.

3.2 A comparison of complete global optimization solvers (Neumair et al, 2005)

In addition to these methods there exist numerous commercial software programs that implement global optimization techniques. A Comparison of these commercially or freely available solvers is very time consuming and difficult work. Also the results are rely on what kind of test suite are used.

Neumair et al (2005) tested global solvers- BARON, GlobSol, ICOS, LGO, LINGO, OQNLP, Premium Solver, and the local solver MINOS. BARON, GlobSol, ICOS, LGO, LINGO, and Premium Solver are based on a deterministic branch and bound approach. OQNLP is a stochastic method which uses the scatter search and multistart heuristic algorithms. They concluded that among the currently available global solvers, BARON is best in terms of the performance and robustness. OQNLP, which is known as the best stochastic solver is close to BARON. However, none of the current global solvers is fully reliable: ICOS is excellent for pure constraint satisfaction problems, while slower than BARON, has excellent reliability properties. In problems which have less than 100 variables, BARON showed the success result of 90%, which is the best one among the solvers compared in the paper. But in problems with > 100 variables, OQNLP solves (within the imposed time limit) the highest percentage (72%) of problems.

3.3 “No free lunch theorems for optimization” (Wolpert & Macready, 1997)

So which algorithm is the best? Since Neumair’s results are based on specific test problems, it is hasty to say BARON or OQNLP is the best for all optimization problems.

Wolpert & Macready (1997) developed a framework to explore the connection between effective algorithm and the problems they are solving. With this framework and test, they concluded “the average performance of all search algorithms over all problems is equal” (No free lunch theorems). They warned the danger of comparing algorithms by their performance on small sample of problems. This also means, we have to use the right algorithm for the right problem and need problem-specific knowledge to be incorporated into the behavior of algorithm.

CHAPTER V. HYBRID SEQUENTIAL NICHE ALGORITHM FOR PROBLEMS WITH SOLUTION MULTIPLICITY

1 Introduction

Detection of multiple solutions in optimization problems is important because multimodal objective functions are common in engineering and physics. For this reason, algorithms for finding all solutions in multimodal problems are desirable. Several deterministic methods such as the *homotopy method* (Sun and Seider, 1995), *interval methods* (Stadtherr et al., 1995), *mixed integer nonlinear programming* (McDonald and Floudas, 1995), and *global terrain methods* (Lucia and Yang, 2002) are suitable for finding multiple solutions. In stochastic optimization, traditional genetic algorithms (GAs) always coalesce to only a single solution, thus they are not suitable for multimodal objective functions. Several GAs with ‘*niche technique*’ were adapted to address the multimodal problems (Beasley et al., 1992; Goldberg and Richardson, 1987; De Jong, 1975; Mahfoud, 1992; Miller and Shaw, 1996). They were successful in many cases by maintaining the diversity of certain properties within the population, but still have some limitations. First, they assume that all optima are evenly distributed. Second, these methods require *a priori* information about the number of solutions. Also the computational effort is larger than normal GA. Details of problems in niche techniques will be described later.

In this part, a *hybrid sequential niche algorithm* is described to create a reliable multimodal optimizer with reasonable performance. It uses a niche technique suggested by Beasley et al. (1993) to find all solutions sequentially. In addition, robustness is enhanced with *niche size adaptation*, which adjusts the niche size according to the problem space

topology in the vicinity of multiple optima. Our algorithm deploys a *deterministic local search*, therefore it converges much faster than the traditional niche methods.

1.1 Niche methods

A key point for locating multiple solutions with stochastic search is how to maintain population diversity. Traditional genetic algorithms coalesce to a single solution, thus traditional genetic algorithms do not offer a good strategy for optimization problems with many local optima. In biology, a *niche* refers ‘a different subspace to support different types of life’ (Miller and Shaw, 1996). In *niche genetic search*, we borrow this concept of nature to tackle solution multiplicity. The niche idea in multimodal search is illustrated in Figure 38. Around each local extremum, a subspace with limited capacity to support candidate solutions is introduced. The total capacity a niche can occupy is proportional to the quality of the extremum. The neighborhood of superior maxima can hold more individuals than inferior local maxima. This limitation mechanism prevents the population from converging to a single solution or getting fragmented into multiple extrema simultaneously. Several methods based on niche formation have been developed. De Jong (1975) and Mahfoud (1992) proposed a *crowding method*. Goldberg and Richardson (1987) introduced the mechanism *fitness sharing* to maintain the population diversity. Miller and Shaw (1996) proposed a *dynamic niche method* to reduce computational expense of fitness sharing. Beasley et al. (1993) developed a niche technique with sequential identification of multiple extrema.

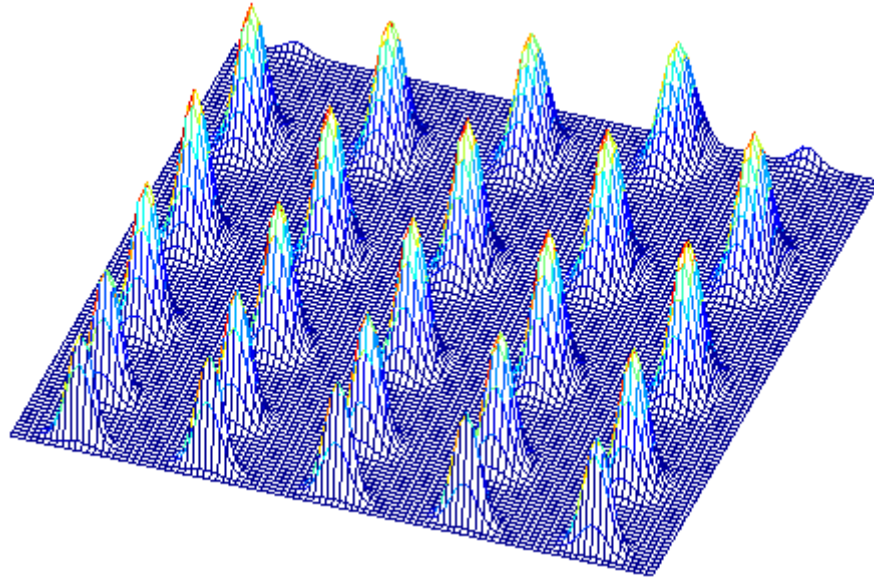


Figure 38: Multiple niches in two-dimensional search space. Each peak conforms a local extrema and niche area of attraction.

The sequential niche technique is an iterative method using knowledge from previous stages to avoid researching the area near previously found extrema. Blocking a niche obtained in the previous stages, sequential niche algorithm proceeds to exploring different regions of the search space. The solution strategy advances sequentially, updating information about the set of previously detected niches. The idea of avoiding part of the feasible region is also used in tunneling methods (Levy and Montalvo, 1985) as well as global terrain methods (Lucia and Yang, 2002). To implement this concept, a penalty known as '*derating function*' is dynamically incorporated into the objective function. The derating function bars the population from reentering existing niches in later stages. The sequential niche technique only needs small population thus requiring less calculation times than other niche methods. Therefore, we adapted a sequential framework for the engineering design problems with multiple solutions. However, in contrast to previous work, we also incorporate gradient-based deterministic optimization techniques to improve performance. The resulting speed-up is critical in large engineering design problems.

A specific difficulty for niche methods concerns proper determination of a suitable niche radius (Deb and Goldberg, 1989; Beasley et al., 1993). If the radius is chosen too small, candidate solutions converging near the extremum may be mistaken for a new solution. If the radius is too large, two or more optima may be lumped together erroneously into a single niche, thus failing to distinguish separate local optima. To tackle this difficulty, we will introduce an *adaptive* technique to choose the niche size automatically.

1.2 Hybrid genetic algorithms

The strength of genetic algorithms - simplicity and robustness- are often offset by excessive resource requirement. To accelerate the overall performance, stochastic GA elements can be combined with a local deterministic search. This *hybrid scheme* combines advantages of the robust GA with the fast convergence of gradient-based optimization thus leading to better performance. In a *two-stage hybrid method*, GA iteration proceeds until individuals begin to coalesce in some region. Then, the method switches to local optimizer for fast computation of the precise extremum coordinates using a gradient-based deterministic algorithm. Thus hybrid methods improve both the speed of finding solutions as well as accuracy. Li and Aggarwal (2000), and Sabatini (2000) adopted a gradient-based local search technique to their hybrid genetic search. Also Chelouah and Siarry (2003) combined continuous genetic algorithms with Nelder-Mead simplex algorithms (Nelder and Mead, 1965). However, few hybrid methods address multimodal optimization.

2 Methodology

This section introduces a new *two-stage* hybrid niche algorithm outlined in Figure 39. A sequential niche technique performs global search. Newton, Quasi-Newton or simplex algorithms are deployed to precisely locate multiple solutions. Newton-based local

search may be used when derivative information of the cost function is available. When derivative information is not available or the search space is ill-conditioned, the Nelder-Mead downhill simplex method may serve as the local optimizer. In the first stage, GA iterations proceed until the population of candidate solutions converges into a regional cluster. An event defining a new niche is identified, once the majority of individuals coalesce around the best individual inside the cluster within a *niche* radius. Starting from the rough solution called *best-in-cluster (BIC)*, the algorithm switches to a local optimizer to precisely locate the extremum in the second stage. The distance of the precisely calculated extremum with respect to rough *BIC* defines the *niche radius*. Specifically, the niche radius is proportional to the Euclidean distance between the local extremum and *BIC*. The precise local extremum together with the newly computed niche radius completes the definition of a new *niche*. The procedure continues to search for additional extrema. Once again returning to a new genetic search stage, a new population sample is generated randomly.

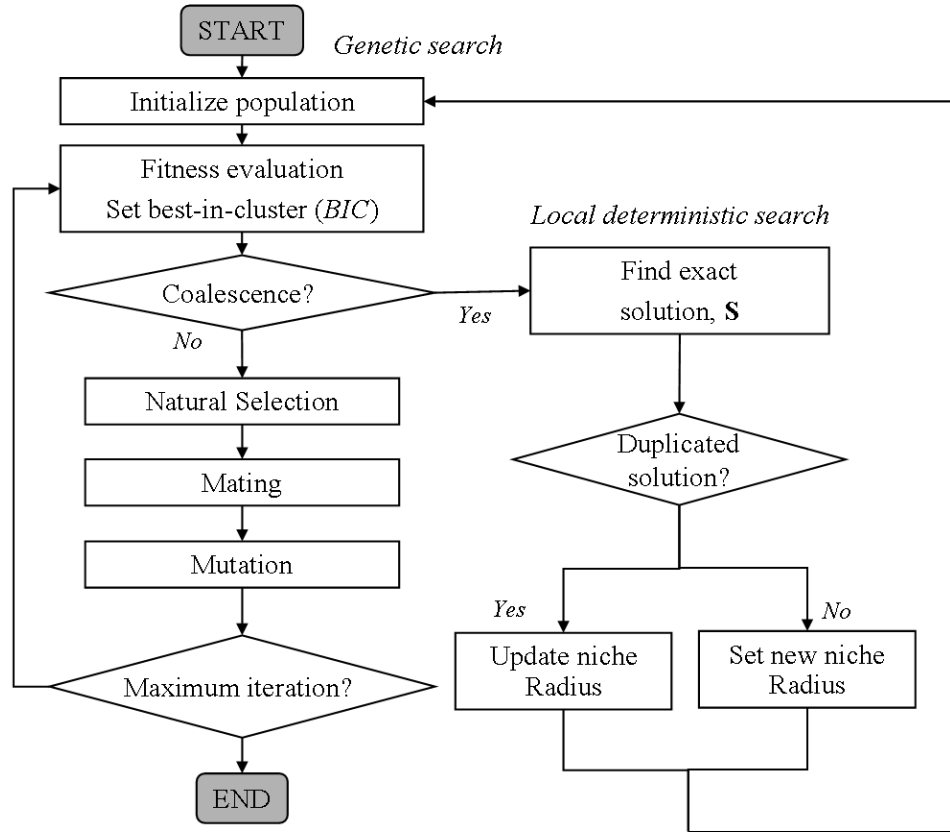


Figure 39: Flowchart of the two-stage hybrid sequential niche algorithm. Genetic search of stage1 runs until termination condition is satisfied. When the majority of the individuals coalesce around the best-in-cluster, stage 2 activates a local optimizer. Taking the *BIC* solution as the initial guess, the local gradient-based method converges to the precise local minimum **S**. The niche radius is adapted according to the distance between *BIC* and the solution **S**.

If the precise solution S is new, a novel niche is registered. If the precise solution has already been identified, the radius of its niche is augmented to block an expanding area of attraction. Thus, niche radii are dynamically adjusted, which ensures locating all solutions even when their distribution is not uniform. The proposed technique requires no *a-priori* guesses for the number and distribution of solutions. The algorithm terminates after a preset generation or solution count has been reached. A detailed description of the new algorithm follows in next subsections.

2.1 Population size

We start the hybrid methods with a random ensemble of individual candidate solutions or initial guesses. In classical methods, the population size should be augmented by a factor corresponding to the expected solution multiplicity (Goldberg, 1989). In sequential niche techniques, the entire population is devoted to searching only one local extremum at a time, thus being less vulnerable to distraction of multiple extrema. Thus, the population size in our algorithm does not require an increase corresponding to the solution multiplicity, although the existence of more extrema will incur a longer sequence to locate them all.

2.2 Derating functions and modified fitness

The main feature of sequential niche techniques aims at avoiding re-exploring the niche regions already detected in previous stages. To implement this feature, the fitness function is modified by incorporating a penalty barring the area of attraction of previous niches. The modified fitness function $M(\mathbf{x})$ is composed of original fitness $fit(\mathbf{x})$ multiplied by a penalty implemented by a dynamically adjusted *derating function* $G(\mathbf{x}, S_j)$ as in eq (48).

$$M(\mathbf{x}) = \text{fit}(\mathbf{x}) \cdot \prod_{j=1}^p G(\mathbf{x}, \mathbf{S}_j) \quad \text{with} \quad \text{fit}(\mathbf{x}) = F_{\max} - F(\mathbf{x})$$

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$
(48)

The vector, \mathbf{x} , encodes the chromosome of an individual. $F(\mathbf{x})$ is the scalar cost function. F_{\max} represents an upper bound of possible objective function values. A *derating function*, $G(\mathbf{x}, \mathbf{S}_j)$, penalizing vicinity of the current individual, \mathbf{x} , to previously detected extrema, \mathbf{S}_j . In eq (49), \mathbf{S}_j is the j^{th} solution found with its niche radius, r_j . The Euclidian distance between \mathbf{S}_j and \mathbf{x} is \mathbf{d} .

$$G(\mathbf{x}, \mathbf{S}_j) = \begin{cases} 0 & \text{if } d < r_j \\ 1 & \text{otherwise} \end{cases} \quad \text{with } d = \|\mathbf{x} - \mathbf{S}_j\|$$
(49)

Accordingly, the derating function value of individuals located within an existing niche is zero leading to a modified fitness of zero. Individuals with zero fitness are not reproduced in natural selection.

2.3 Natural selection, mating, and mutation

In the natural selection step, individuals with superior *fitness* values are chosen to produce competitive offspring for the next generation. Individuals with low fitness are unlikely to reproduce and tend to disappear. Specifically, zero fitness will prevent individuals from reproducing or being selected into the next generation. The chromosome vectors of offspring, $\mathbf{x}'_c, \mathbf{x}''_c$, are computed by combining the parameter values of a pair of parents, \mathbf{x}_a , and \mathbf{x}_b , according to the *arithmetic crossover* formula with one random variable, δ , given in eq (50).

$$\mathbf{x}'_c = (1 - \delta) \mathbf{x}_a + \delta \mathbf{x}_b, \quad 0 \leq \delta \leq 1$$

$$\mathbf{x}''_c = \delta \mathbf{x}_a + (1 - \delta) \mathbf{x}_b, \quad 0 \leq \delta \leq 1$$
(50)

Mutations. In each reproduction cycle, we choose a given number of individuals for mutation corresponding to a fixed mutation ratio. This adjustable parameter was set to 5% in the case studies. A single trait corresponding to the k^{th} component of a individual selected for mutation is altered according to the random event given in eq (51).

$$x'_k = \begin{cases} \min \left[x^{up}, x_k + \sigma (x_k^{up} - x_k^{low}) \right] & \sigma \geq 0 \\ \max \left[x^{low}, x_k + \sigma (x_k^{up} - x_k^{low}) \right] & \sigma < 0 \end{cases} \quad (51)$$

Here, x'_k is k^{th} trait of mutant \mathbf{x}' randomly perturbed from the k^{th} trait of original individual. x_k^{up} refers to an upper bound, x_k^{low} to a lower bound of permissible trait values. σ is random scalar number between -1 and 1. Mutations help expand the stochastic search into unexplored areas of the search space.

2.4 Niche Detection and precise extremum location

In the two-stage hybrid algorithm, coarse genetic search is followed by deterministic local search. The local optimizer is activated once candidate solutions of the genetic stage coalesce around a cluster. The condition for detecting a new cluster involves the computation of the population density in the region. Clustering has occurred when a given fraction ε of the entire population N_{pop} is contained with an r -neighborhood around the currently best candidate solution (\mathbf{x}_{BIC}). The best-in-class (BIC) will also play a role in dynamic niche radius adaptation. The r -neighborhood is established with the help of the distance function $sh(d, i)$.

$$\begin{aligned}
m &\geq \varepsilon \cdot N_{\text{pop}} \\
m &= \sum_{i=1}^{N_{\text{pop}}} sh(d, i) \\
sh(d, i) &= \begin{cases} 1 & \text{if } d < r_c \\ 0 & \text{otherwise} \end{cases}, \quad d = \|\mathbf{x}_i - \mathbf{x}_{BIC}\|
\end{aligned} \tag{52}$$

Here, m is a region membership counter. The fraction of population, ε was set to 0.6 in the case studies presented in this chapter. The parameter r_c refers to the cluster radius. If condition (52) is satisfied, the population has clustered around the currently best individual. In addition, the niche detection criterion triggers the hybrid method to switch to a local optimizer. Local deterministic optimization is used to rapidly compute the precise extremum coordinates inside the cluster. In the case studies, we report the performance of the steepest descent method. Alternatively, Newton, Quasi-Newton, Trust Region (Tang et al., 2005; Zhang et al. 2007) or Nelder Mead Simplex algorithm could be used for local deterministic search.

2.5 Automatic niche radius adjustment

Deb and Goldberg (1989) suggested setting a fixed niche radius, r , as a function of the degrees of freedom, n , and the number of expected solutions, p as in eq (53).

$$r = \frac{\sqrt{n}}{2^n \sqrt{p}} \tag{53}$$

This rule requires *a-priori* knowledge of the number of solutions and can be expected to work properly when extrema are evenly spread throughout the whole search space. Often it is impossible to anticipate the number and distribution of multiple solutions. Consequently, eq (53) is impractical in many real world applications. Ideally, every niche should have different size depending on the topology of the neighborhood. Fixed niche radius, r , can fail to detect multiple extrema when they are close to each other or cause

unnecessarily long computations. A serious problem associated with fixed niche radius strategies especially in hybrid search is illustrated in Figure 40. The example addresses a one dimensional optimization problem with the objective function given in eq (54).

$$\begin{aligned} \min F(x) &= 12 - (3 \sin x + x) \\ 0 &\leq x \leq 10 \end{aligned} \tag{54}$$

This example has the global minimum \mathbf{S}_1 and one local minimum \mathbf{S}_2 in the range of $0 \leq x \leq 10$. Several fixed niche radii –unity, $\pi/2$, and π – were tested. In the first run, individuals converge to point \mathbf{B}_1 by genetic search. Once clustering occurs according to criterion (52), the local search is launched to find the exact solution \mathbf{S}_1 . The niche N_1 corresponding to the fixed niche radius around \mathbf{S}_1 is shown in grey. In the next run, individuals coming to lie inside N_1 are penalized. Unfortunately, the fixed niche radius is too small so that new clustering occurs near the niche boundary about points \mathbf{B}_2 or \mathbf{B}_3 . Using these \mathbf{B}_2 or \mathbf{B}_3 clusters as starting points, the same solution \mathbf{S}_1 , is repeatedly re-discovered without advancing the overall progress. The fixed radius algorithm would repeat this futile pattern until exhausting the maximum iteration counter without ever locating the second local minimum \mathbf{S}_2 . The fixed radius strategy also fails when using radii in the range of $r < \pi/2$ or $r > 3\pi/2$. The fixed radius strategy only succeeds when the exact half period, π , between local minima is chosen. However, exact knowledge about the precise distance between solutions is not available in general, so that fixed radius strategies are often inadequate. Fixed niche radius strategies fail to locate multiple minima especially when using hybrid algorithms as illustrated in this example.

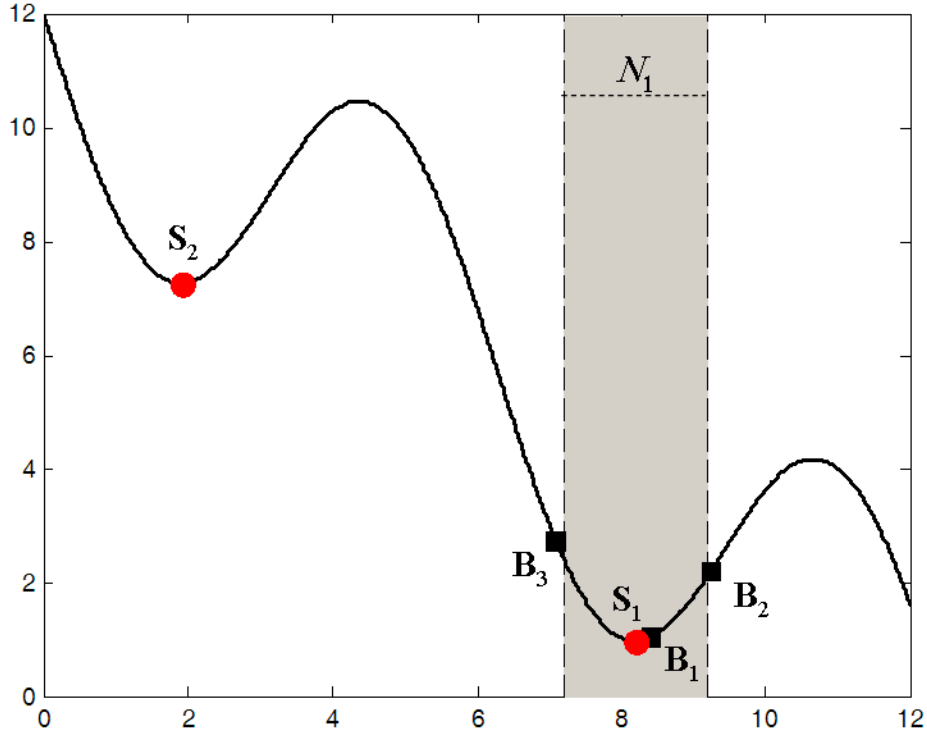


Figure 40: Example demonstrating difficulty with fixed radius strategies in hybrid genetic algorithms. If fixed radius is used with hybrid genetic algorithm, the local solution S_2 can not be found, because genetic search repeatedly clusters close to niche around B_3 or B_2 . This problem is known as trapping.

2.5.1 Variable Niche Radius

To overcome this problem, we developed a *variable niche radius* strategy. It adaptively determines the niche radius as a function of the distance between the best-in-cluster (*BIC*) and the precise extremum. We found this method to be simple, yet more effective than previous strategies like *cooling* (Jelasity, 1998), *dynamic niche clustering* (Gan and Warwick, 2001) or *Covariance Matrix Adaptation Evolution Strategy* (Shir and Back, 2006). Furthermore, it can be effectively integrated into a hybrid schemes, which cannot be done with prior techniques.

In our algorithm, the niche radius is determined by the distance between the best-in-cluster, \mathbf{B} , and local extremum, \mathbf{S} . Around the local extremum, \mathbf{S} , the scalar r defines the spatial extension of the niche. In order to account for ill-conditioned regions in the vicinity of the extrema, we suggest two types of niche shapes. When the optimal region is symmetric with the optimum located in the center of the optimal region as depicted in Figure 41a, we recommend a hyper-spherical model for the niche area of attraction. Formally this symmetry condition would be satisfied for fitness functions with condition numbers of their Hessian close to unity. Accordingly, the size of niche radius is given in eq (55),

$$r = \|\mathbf{S} - \mathbf{B}\| + \alpha \quad (55)$$

where α is default minimum niche radius preventing zero radii in case \mathbf{B} and \mathbf{S} coincide. If a subsequent search converges to the same extremum, \mathbf{S} , the niche radius is expanded as depicted in Figure 41b. The adaptation automatically enlarges the niche region to prevent repeated clustering.

Asymmetric optimal regions occur in cases of Hessians with large differences in their eigenvalues. For these problems, we propose a hyper-rectangular box to approximate the niche's area of attraction. The length of each side rectangle is computed according to eq (56)-(57),

$$\left. \begin{aligned} r_k^+ &= |S_k - B_k| + \alpha \\ r_k^- &= \alpha \end{aligned} \right\} \text{ if } B_k \geq S_k \quad (56)$$

$$\left. \begin{aligned} r_k^+ &= \alpha \\ r_k^- &= |S_k - B_k| + \alpha \end{aligned} \right\} \text{ if } E'_k < E_k \quad (57)$$

where S_k is the k^{th} trait of the extremum \mathbf{S} , B_k is the k^{th} trait of best-in-cluster \mathbf{B} . The scalar r_k^- measures the length of left side of k^{th} box dimension, while r_k^+ is the length of right side of k^{th} box dimension. Figure 41c illustrates the hyper-rectangular niche area with highly asymmetric topology and off-center local extremum.

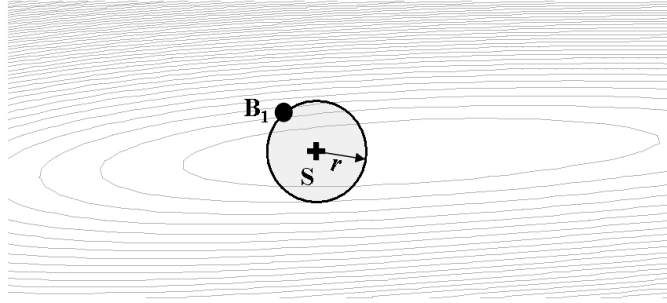
The update of the rectangular niche region requires a maximum operator when the same extremum is found from in a subsequent search stage. In case of a large cluster converging to the same niche, the sides of the hyper box are expanded so that the largest dimensions of sides ever found are taken into consideration. This operation is implemented in eq (58).

$$r_{\text{new},k}^+ = \max(r_{1,k}^+, r_{2,k}^+), \quad r_{\text{new},k}^- = \max(r_{1,k}^-, r_{2,k}^-) \quad (58)$$

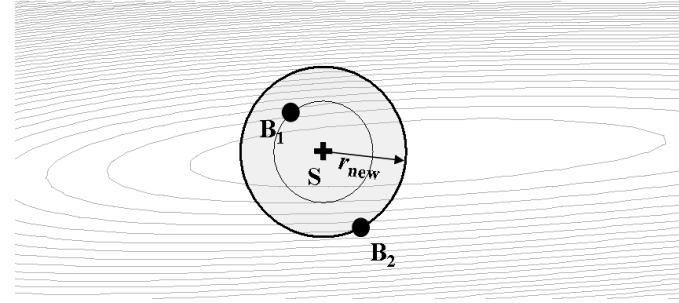
A new niche box extension N_3 takes maximum of the side lengths corresponding to the points \mathbf{B}_2 and \mathbf{B}_1 as depicted in Figure 41d. The hyper-rectangular strategy also requires an adjusted derating function given in eq (59). The hyper-rectangular niches lead to smaller niche areas when compared to the hyper-spherical approach, but is safer for ill-conditioned problems.

$$G(\mathbf{x}, \mathbf{S}_j) = \begin{cases} 0 & \text{if } d_{j,k} < r_{j,k} \\ 1 & \text{otherwise} \end{cases} \quad \forall k \quad (59)$$

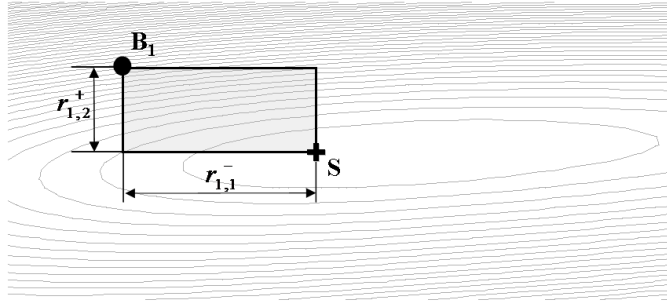
$$\text{with } d_{j,k} = |S_{j,k} - x_k|, \quad r_{j,k} = \begin{cases} r_{j,k}^+, & \text{if } x_k \geq S_{j,k} \\ r_{j,k}^-, & \text{if } x_k < S_{j,k} \end{cases}$$



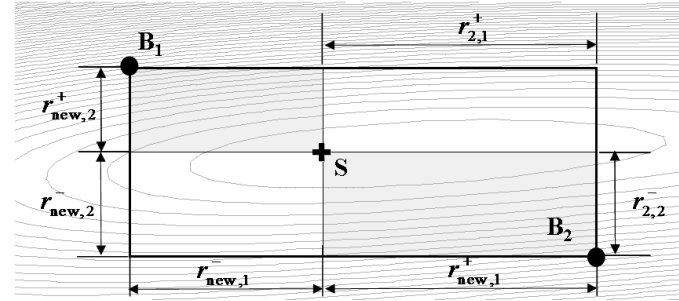
a)



b)



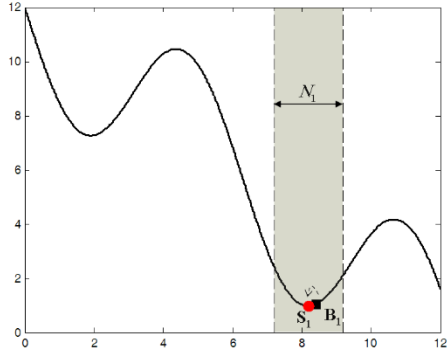
c)



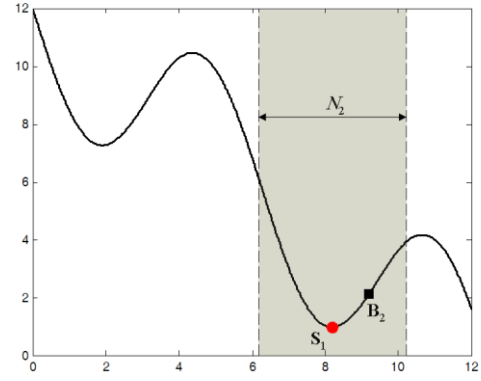
d)

Figure 41: Two different niche types. a) In hyper-spherical type, the niche radius size is $r = \|\mathbf{B} - \mathbf{S}\| + \alpha$. \mathbf{B}_1 is best-in-cluster (*BIC*) found by the genetic algorithm, and \mathbf{S} is a real solution found by a local optimizer starting from \mathbf{B}_1 . b) In hyper-rectangular type, only one length side of rectangular is decided from *BIC* c) If same solution \mathbf{S} is found from different initial point \mathbf{B}_2 , niche radius r is augmented to r_{new} . d) In the hyper rectangle strategy, the niche size is updated by taking each maximum length side.

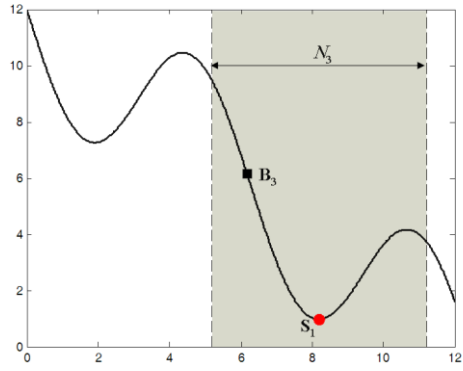
We solved the previously problematic example using the two variable niche size methods. Figure 42 shows how the hybrid sequential niche algorithm determines the global optimum \mathbf{S}_1 and local optimum \mathbf{S}_2 without the user specifying the niche radius. We use a hyper-spherical niche type because this geometry of optimal region is not skewed. After finding the solution \mathbf{S}_1 from \mathbf{B}_1 , niche N_1 is set as depicted in Figure 42a. In next two runs, the same solution \mathbf{S}_1 is found again from \mathbf{B}_2 or \mathbf{B}_3 . The niche area is updated from N_1 to N_3 through these runs as shown in Figure 42b, c. Finally, \mathbf{S}_2 is found from \mathbf{B}_4 . However, like all numerical methods, adjustable parameters like the minimal radius parameter α , or the fraction of the population clustering ε are required. In the following case studies, we recommend for α a range of one to ten percent of the variable range, in example above $\alpha=1$, and clustering parameter of $\varepsilon=0.6$.



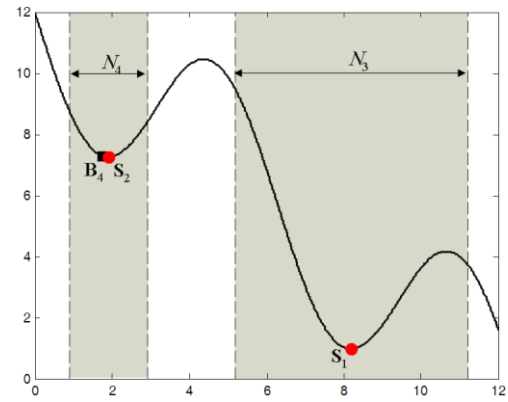
a)



b)



c)



d)

Figure 42: The example of using variable radius size in hybrid genetic algorithm. a) The first solution, S_1 is found from B_1 b, c) Niches are updated from N_1 to N_2 , from N_2 to N_3 . d) Finally, S_2 can be found from B_4 .

3 Benchmark test cases

In this section, several test functions *-Himmelblau* (**H**) and *Grienwank* (**G**), Bohachevsky (**B**), modified *Vincent* (**V**), and *Ackley* (**A**) functions are used to verify our algorithm's efficiency and robustness. These functions were frequently tested in many papers and considered challenging to find all extreme points. These problems have unconstrained multimodal objective functions. Mathematical formulations of the test functions are given in Table 16, topology of the search spaces are depicted in Figure 43 (**H**), Figure 44(**G**), Figure 45(**B**), Figure 46(**V**), and Figure 47(**A**). Functions **H** and **G** have several maxima/minima with the same objective function value, so the extrema are equally good. The functions, **B**, **V**, and **A** have many local minima, but possess a unique global minimum. Our algorithm is also compared with our implementation of Beasley's sequential niche technique. In our hybrid algorithm, steepest descent method is deployed as a local optimizer.

Table 16: Test functions

	Name	Function	Range
H	Himmelblau	$H(x) = 200 - (x_1^2 + x_2 - 1)^2 - (x_1 + x_2^2 - 7)^2$	$[-6, 6]^2$
G	Grienwank	$G(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-10, 10]^n$
B	Bohachevsky	$B(x) = \sum_{i=1}^{n-1} (x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) - 0.4 \cos(4\pi x_{i+1}) + 0.7)$	$[-10, 10]^n$
V	Vincent (Modified)	$V(x) = -\frac{1}{n} \sum_{i=1}^n x_i \cdot \sin(10 \cdot \log(x_i))$	$[0.25, 10]^n$
A	Ackley	$A(x) = -20 \cdot \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}\right) + 20 + e$	$[-10, 10]^n$

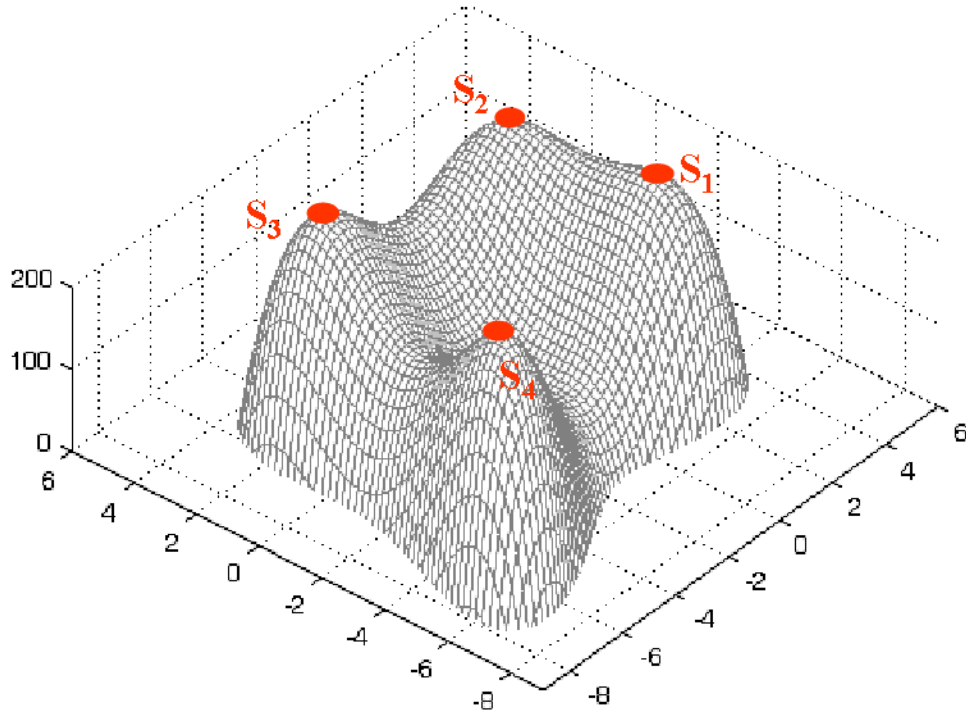


Figure 43: Himmelblau's function (H). It has four maximum- S_1 (3.58,-1.86), S_2 (3.0,2.0), S_3 (-2.815,3.125), and S_4 (-3.78,-3.28).

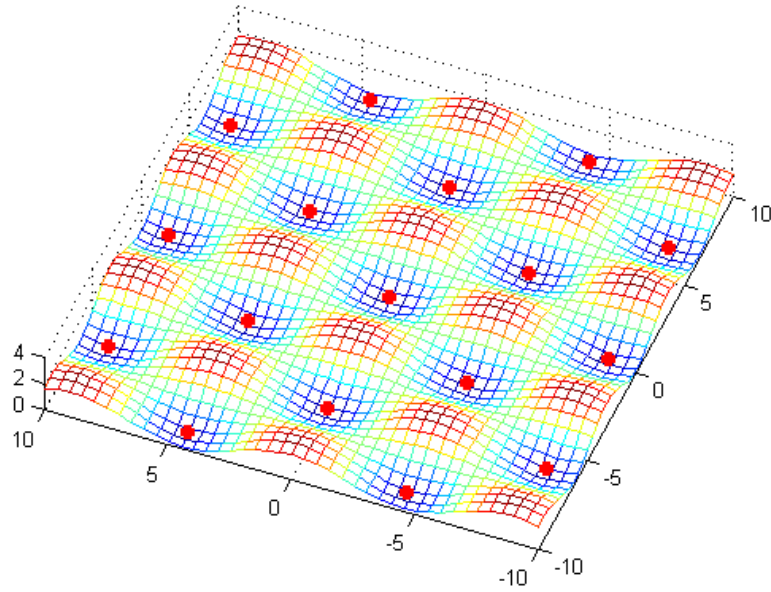


Figure 44: Two-dimensional Griewank's function (**G**). It has 17 equal minima in range of $[-10,10]^n$.

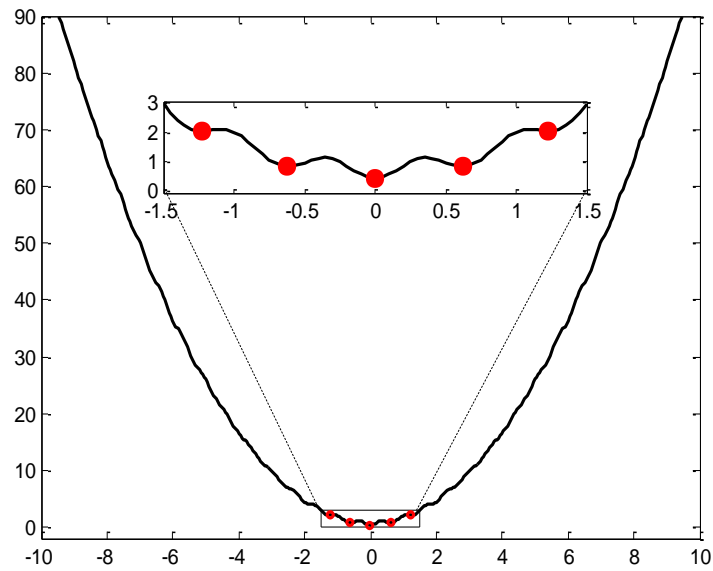


Figure 45: One-dimensional Bohachevsky's function (**B**). It has 5 minima range of $-10 < x < 10$ and search space is asymmetric.

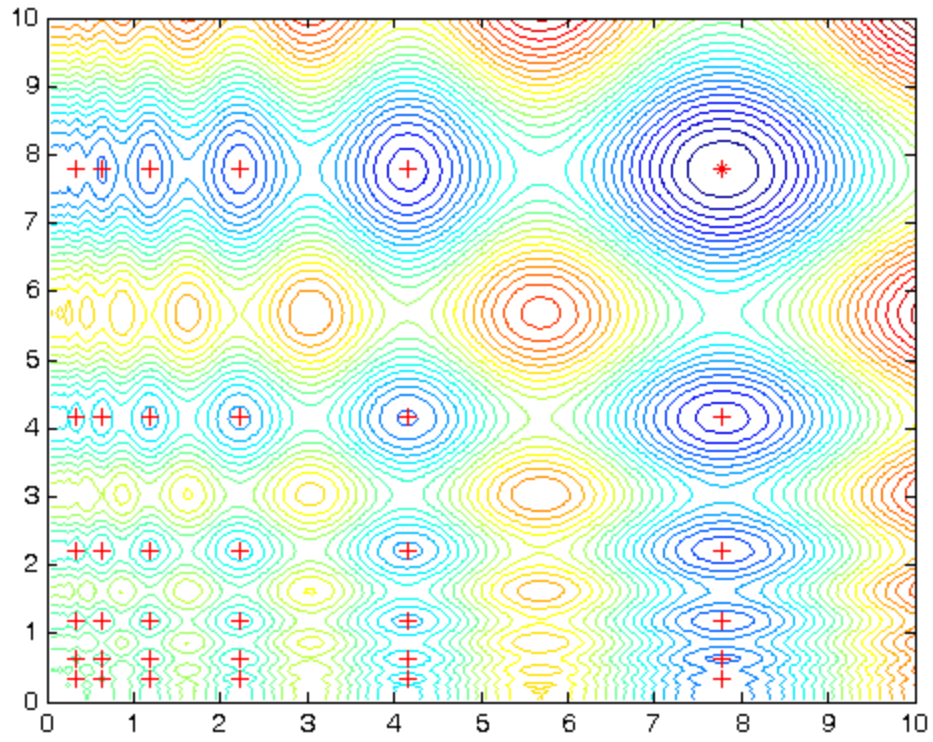
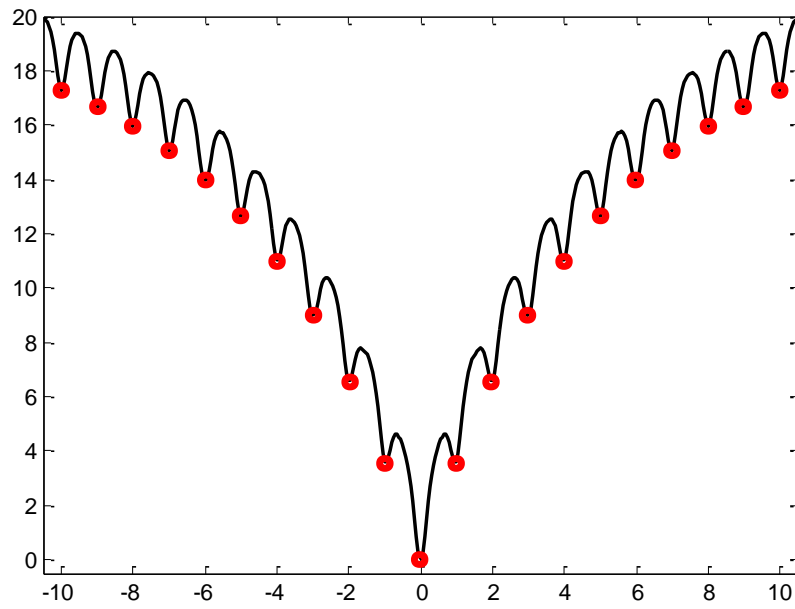
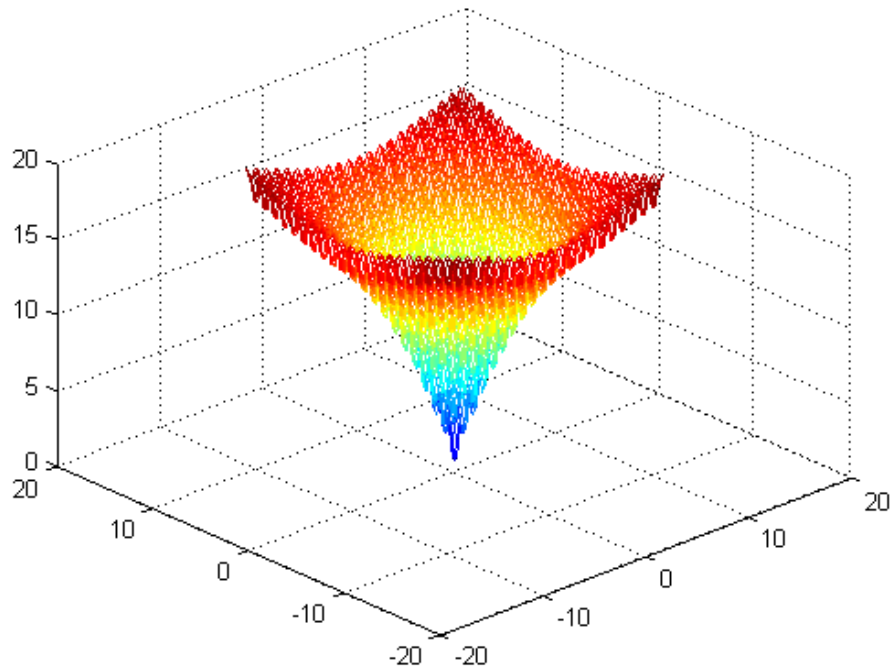


Figure 46: Two-dimensional modified Vincent's function (\mathbf{V}). It has one global minimum and 35 local minima in range of $[0.25, 10]^n$.



a)



b)

Figure 47: Ackley's function (**A**), a) $n=1$ b) $n=2$, This function has one global minimum and 21 local minima.

3.1 Results

In the simulation experiments, the range of the search space, the dimensionality of the problem, and the population size were varied. We solved each case study twenty different times; the values in

Table 18 reports the averages of twenty trials. We systematically evaluated the performance of each stochastic optimization algorithm in terms of speed (how quickly that algorithm completes for the same levels of accuracy), success rate (what proportion of runs finds all expected optima) and consistency (how much variations in the previous three criteria can be expected from one run to the next).

Performance-Speed. Our algorithm was compared to Beasley's sequential niche technique. Both algorithms were implemented in Object Pascal (Codegear). We tested the performance of both algorithms on problem H with four maxima shown in Figure 43 (Deb and Goldberg 1989). The prior sequential niche technique took 306 iterations for finding all four maxima with average calculation time of 531 msec. Our hybrid sequential niche algorithm took only 6.2 msec which is eighty eight times faster than the prior niche GA. We also tested the two-dimensional G with 17 equal and narrowly spaced minima. Beasley's niche technique required a population size of 800. The novel hybrid niche algorithm on the other hand only required a population size of 20 individuals, still leading to a performance increase of 1.6. In this case, the hybrid exhibits less dramatic performance increase, because the sequential search is through, but consumes time. Beasley's niche technique attacks all solutions indiscriminately leading to excessive failure rates. Hence, speed without reliability is insufficient for design optimization.

Success rates. Despite a population size of at least 800, Beasley's technique on problem H typically detected only three solutions, failing to detect at least one solution.

The averages success rate defined as finding all solutions was only 50%. When using smaller population sizes, its success rate fell as low as zero. In contrast, our algorithm required a population size of only 20 to find all four solutions in every trial. This result corresponds to a success rate of almost 100% with reasonable computational time. A summary of the comparative simulations is reported in Table 18.

We conducted more thorough tests according to the challenging test problems listed in Table 17 and Table 18. The prior method had a success rate of close to zero, so that the performance could not be compared in a meaningful way. In all tests, success rate of our novel hybrid method was almost 100%. We first chose hyper-spherical niche shape. If the success rate was not perfect in after several runs, we also experimented with the hyper-rectangular niche shape. Case 5 and case 6 best demonstrate the differences between spherical and rectangular niche adjustment strategy. With the spherical approach in the two-dimensional problem **B**, the success rate was only 60% to locate all solutions. The hyper-rectangular approach never failed to detect all extrema. We recommend hyper-rectangle niche adjustment for cases with insufficient information about the geometry of the search space. However, spherical niche radius adjustment strategy is generally faster than the hyper-rectangular approach.

Table 17: Results of Beasley's sequential niche technique

No	F/n	Population size	Iteration/maximum	Niche shape	Time (msec)	No of solution found/desired	Success rate
1	H/2	800	306/2,000	sphere	531	3.1/4	50%
2	G/2	800	977/2,000	sphere	977	17/17	100%
3	B/1	400	2,000/2,000	sphere	1,156	1/5	0%
4	B/2	800	2,000/2,000	sphere	3,687	1/25	0%
6	V/1	400	2,000/2,000	sphere	1,437	2.5/6	0%
7	V/2	800	3,000/3,000	sphere	18,543	6/36	0%
8	A/1	400	2,000/2,000	sphere	1,406	1/21	0%

Table 18: Results of hybrid sequential niche algorithm

No	F/n	Population size	Iteration /maximum	Niche shape	Time(msec)	No of solutions found/desired	Success rate
1	H/2	20	25/1,000	sphere	6.2	4/4	100%
2	G/2	20	151/1,000	sphere	616	17/17	100%
3	G/10	100	3,000/3,000	sphere	32,984	203/-	-
4	B/1	20	73/1,000	sphere	98	5/5	
5	B/2	40	1,097/2,000	sphere	1,770	24.1/25	60%
6	B/2	40	758/2,000	rectangle	3,500	25/25	100%
7	B/10	100	3,000/3,000	rectangle	17,015	196/-	-
8	V/1	20	84/1,000	sphere	343	6/6	100%
9	V/2	40	7,811/1,0000	sphere	73,344	36/36	100%
10	A/1	20	211/1,000	sphere	187	21/21	100%
11	A/2	40	21,533/30,000	sphere	117,453	441/441	100%

4 Realistic chemical engineering applications with solution multiplicity

Two large-scale applications should further demonstrate the suitability of the novel optimization method for chemical engineering problems. Non-isothermal catalyst pellets are known to exhibit multiple steady states. We wish to find all solutions of steady state non-isothermal concentration profiles in the catalytic pellet using hybrid sequential niche algorithm. The second example demonstrates a large distributed inversion problem to determine unknown transport and kinetic parameters with solution multiplicity.

4.1 The concentration and temperature profile of catalytic pellet.

For non-isothermal condition, the dimensionless concentration profile, y , in a porous catalytic pellet can be written as in eq (60). The independent variable x is the dimensionless pellet radius.

$$\frac{d^2y}{dx^2} + \frac{2}{x} \frac{dy}{dx} = \phi^2 y \exp\left(\gamma\beta \frac{1-y}{1+\beta(1-y)}\right) \quad (60)$$

The dimensionless variables, Thiele modulus, ϕ , activation energy, γ , heat of reaction, β , are defined in eq (61). Detailed parameter values and equations can be found elsewhere (Kulkarni et al., 2008).

$$\gamma = \frac{E_a}{RT}, \beta = \frac{C_s \Delta H D}{K T_s}, \phi = R \sqrt{k_o / D} \quad (61)$$

For the same parameter values γ , β and ϕ , multiple profiles may satisfy the transport and reaction equations in the catalyst pellet at steady state. Discretization via orthogonal collocation over finite elements converts the non-linear partial differential equation in (60) into the nonlinear algebraic system in (62)-(63):

$$Ay + G(y) = 0 \quad (62)$$

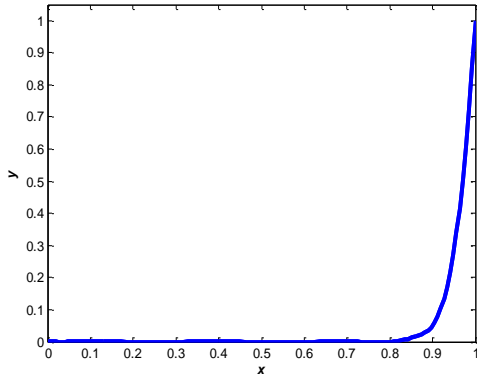
$$B y = c \quad (63)$$

In this non-linear algebraic system, y is the vector of dimensionless concentrations at the collocation nodes. Eq (62) is the transport and reaction equations suitably discretized at internal nodes of each finite element domain. Eq. (63) implements boundary conditions $x = 0, x = 1$, as well as C_0 and C_1 -continuity conditions at finite element boundaries. This case study uses four elements, j , with five nodes, i , in each segment. The nonlinear vector function $G(y)$ given in eq (64).

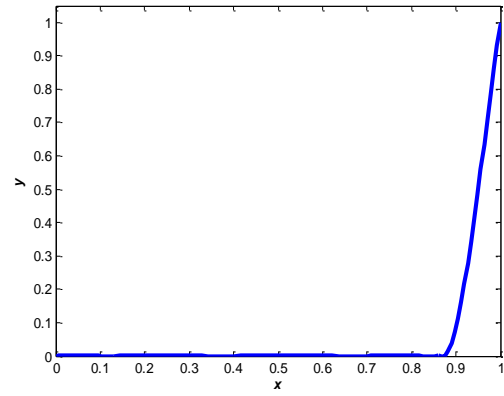
$$G(y_i^j) = -\phi^2 y_i^j \exp\left(\frac{\gamma\beta(1-y_i^j)}{1+\beta(1-y_i^j)}\right) \quad i = 1, \dots, 5 \quad j = 1, \dots, 4 \quad (64)$$

The objective of this problem is to find all solutions of the steady state pellet profiles. This task is accomplished by detecting all minima of the residual error function associated with eqs (62) and (63).

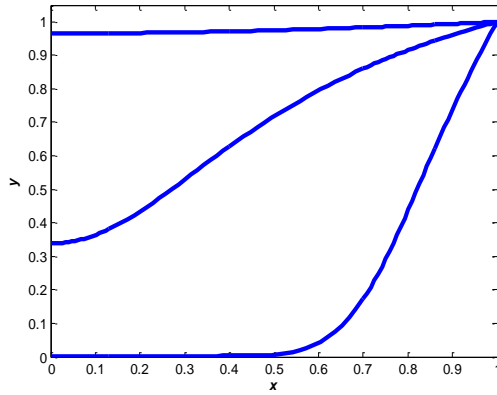
System in (62)-(63) is a multi-dimensional problem with solution multiplicity which we solved with a hybrid niche algorithm combined with Newton-type method to accelerate the local optimization. Inside certain parameter ranges for the dimensionless parameters- f, g, b , we obtained different concentration profiles as expected. Figure 48 depicts typical results with several sets of parameters f, g, b . The average calculation time was about 33.98sec; which is quite reasonable. The same problem was also tackled with a deterministic global terrain method developed by Lucia et al. (2008) with comparative CPU time. More details on tackling the challenge of solution multiplicity in pellet reactors can be found elsewhere (Kulkarni et al., 2008).



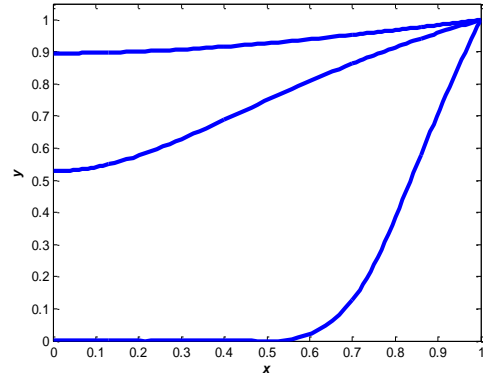
a) $\gamma = 10, \beta = 0.4, \phi = 10$.



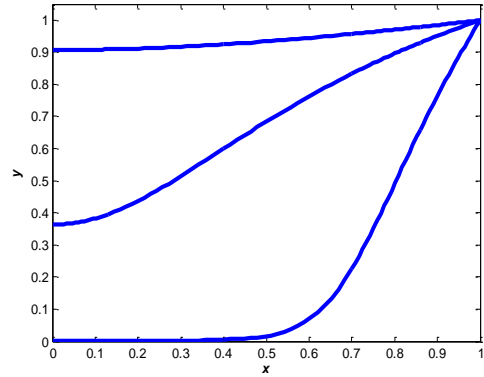
b) $\gamma = 30, \beta = 0.4, \phi = 1$



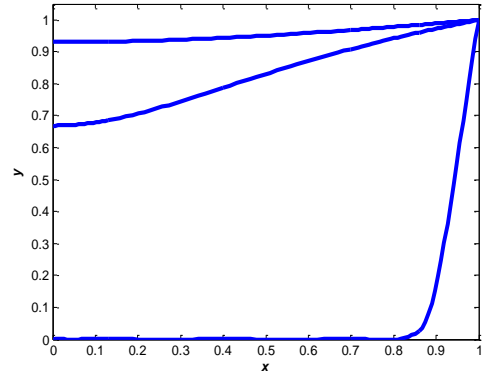
c) $\gamma = 20, \beta = 0.4, \phi = 0.6$.



d) $\gamma = 30, \beta = 0.4, \phi = 0.6$



e) $\gamma = 40, \beta = 0.2, \phi = 0.6$



f) $\gamma = 40, \beta = 0.3, \phi = 0.5$

Figure 48: Pellet concentration profiles evaluated by OCFE with hybrid sequential niche algorithms, where x is the dimensionless pellet radius and y is the dimensionless concentration in eq (60). For parameters sets in frame a) and b), our algorithm verified only one solution exists. The algorithm found three solutions for the parameters and conditions of the examples depicted in frame c-f. More details about these cases are described elsewhere (Kulkarni et al., 2008).

4.2 Determination of reaction kinetics in catalytic reactors

This inversion problem aims at quantifying unknown micro-scale chemical kinetic and transport phenomena inside catalytic pellets based on only macroscopic measurements in the reactor bulk. Specifically, we determine all possible solutions of the reaction rate, k_m and diffusivity, D of catalytic pellet reactor with the novel hybrid method. Solution multiplicity is due to the non-linearity in the pellet kinetics as discussed above as well as measurement errors at the macro-scale (Zhang et al., 2007). Multiple solutions are found frequently in kinetic inversion problem of distributed systems due to experimental data errors. These transport equations involve an objective function minimizing weighted difference between measurements as in eq (65) subject to the distributed transport model with partial differential equation constraints given in eqs (66)-(68).

$$\min_{D, k_{ref}, \mathbf{C}, \mathbf{T}} \psi(\theta) = \left(\begin{bmatrix} \mathbf{C} \\ \mathbf{T} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{C}} \\ \hat{\mathbf{T}} \end{bmatrix} \right)^T F^{-1} \left(\begin{bmatrix} \mathbf{C} \\ \mathbf{T} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{C}} \\ \hat{\mathbf{T}} \end{bmatrix} \right) \quad (65)$$

$$-\bar{\nabla} P = \frac{\mu}{k_m} \bar{u} \quad : \text{Darcy's law} \quad (66)$$

$$\nabla(\bar{u} C_A) = D \nabla \nabla C_A + r_A \left| \mathcal{J}^{-1} \right| \quad \text{Mass balance} \quad (67)$$

$$\rho_g C_p \nabla(\bar{u} T) = k_e \nabla \nabla T + r_A \Delta H \left| \mathcal{J}^{-1} \right| \quad \text{Energy balance} \quad (68)$$

where \mathbf{C} is concentration, \mathbf{T} is temperature, $\hat{\mathbf{c}}$ is measured concentration, and $\hat{\mathbf{t}}$ is measured temperature. These partial differential equation constraints are discretized by finite volume approach. A more detailed description of the discretization and parameters values can be found elsewhere (Zhang et al., 2007).

This problem was solved using the hybrid sequential niche algorithm with parameters: population size=10, selection ratio=0.5, and mutation ratio=0.05. To reduce

the problem dimensionality, the objective function (65) was evaluated as the difference between concentration and temperature measurement and the simulated concentration and temperature fields. The simulation used a candidate parameter set to solve the discretization equations, while the hybrid sequential niche search was conducted in the reduced unknown parameter space of k_m and diffusivity, D . For this inversion problem, two distinct solutions were found in 468 msec on Pentium IV with a 3.06GHz processor. Figure 49 shows the sequential progress of the variable niche method. First, coalescence occurred around solution \mathbf{S}_1 . Typically, 12 iteration steps located solution \mathbf{S}_1 precisely. The niche radius was determined based on the cluster dimensions. The penalty terms of the rating function prevent the search in the vicinity of the first solution. The algorithm coalesced close to a region around \mathbf{S}_2 . At this point, a steepest descent method was launched to exactly locate the second solution \mathbf{S}_2 . The optimal parameter sets identified were for \mathbf{S}_1 ($k_m = 0.731 \times 10^{-4} \text{1/s}$, $D = 0.400 \times 10^{-4} \text{m}^2/\text{s}$), and for \mathbf{S}_2 ($k_m = 0.732 \times 10^{-4} \text{1/s}$, $D = 0.799 \times 10^{-4} \text{m}^2/\text{s}$). The two solutions were embedded in niches with different niche radii, $r_1 = 0.05$, $r_2 = 0.08$. For comparison, we also solved this problem with Beasley's sequential method. With his method, a population size of at least 400 was required to locate both solutions reliably with computational time of 136 sec. When an only local optimizer is used for finding solutions, only one solution could be obtained, this solution depends strongly on an initial guess (Zhang et al., 2007).

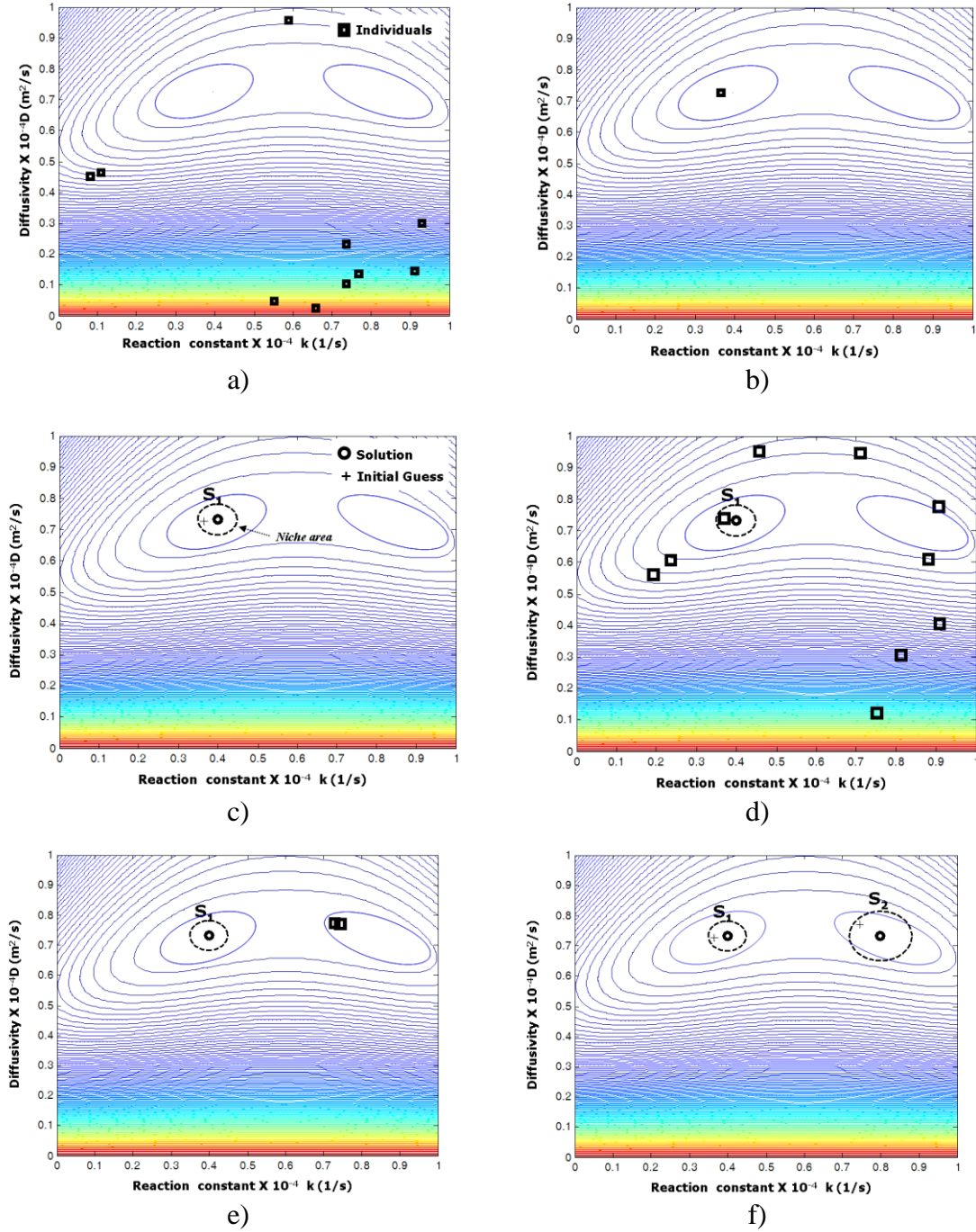


Figure 49: Snapshots of individual distributions in pellet reactor inversion problem. a) Iteration 1: ten individuals are generated randomly. b) Iteration 6: every individual coalesces to point (0.36, 0.72) c) from point (0.36, 0.72), the solution S_1 (0.40, 0.73) was obtained using a local optimizer. Then niche was set. d) Iteration 7; population is reinitialized to find the next solution, e) Iteration 15: every individual coalesces to point (0.76, 0.75). f) Finally, another solution S_2 (0.80, 0.73) was obtained.

5 Conclusions and future works

Limitations: In the basic implementation of algorithm, we produced new random samples of each population whenever a niche is identified. This strategy has been found to be robust, but may be improved by reusing the information obtained from prior candidate solutions. The idea implies a second pool of individuals in addition to the normal population. This pool would collect individuals outside solution clusters, rather than discarding them. Once a new niche has been identified, these outlier samples in special pool could be re-used to generate a new initial generation inexpensively. In some cases, we found this strategy to produce more uniform search at faster performance. To make this modification more robust, further work is needed in the future.

Since our algorithm relies on finite precision arithmetic and addresses non-polynomial hard problems, global convergence guarantee cannot be given rigorously but only in a statistical sense. In addition, the saddle points often pose a challenge in global optimization. However, in our algorithm, saddles are typically not recognized as extremum points, because the candidate population in the vicinity of saddle has both inferior and superior individual. Therefore, to the hybrid algorithms, such a region does not look like a solution cluster. The genetic search typically passes this region instead of inadvertently recognizing it as an extremum.

Like any numerical algorithms, the proposed hybrid sequential niche algorithm requires adjustable tuning parameters. These tuning parameters are population size, the number of maximum iterations, size of default niche, and niche shape.

Nomenclature (Chapter V)

B	best-in-cluster solution
C	Concentration of reactant
\hat{c}	measured concentration of reactant
d	Distance
D	Diffusivity
E_a	Activation energy
fit	Fitness
F	object function value
F_{\max}	maximum value of objective function
G	derating function
ΔH	heat of reaction
M	niche membership count
$M(\mathbf{x})$	modified fitness
N_{pop}	population size
n	number of traits in individual
p	number of solution
r	niche radius
r_A	reaction rate
R	catalytic pellet radius
S	extremum, solution
T	Temperature
\hat{T}	measured temperature
x	dimensionless radius of pellet
X	Individual
x_k	the k^{th} trait of an individual \mathbf{x}
y	dimensionless concentration in pellet

Greek symbols

α	default niche radius
β	dimensionless heat of reaction
ε	scalar for detecting optimal region
ϕ	thiele modulus
γ	dimensionless activation energy

APPENDICES

APPENDIX A- MATHEMATICAL MODELS OF ISOMERIZATION PROCESS

APPENDIX A provides detailed dynamic models of the isomerization flowsheet used to perform the case studies described in Chapter III. Sections 1-4 summarize the variables and constant definitions, as well as the parameter values used in the studies.

1 DIB column

The DIB column is used for separating iC_4 from mixed stream. For this unit, tray by tray balances with hold up and equilibrium were solved. A constant α model is used. This column is controlled by changing the reflux ratio(r) and vapor ratio(s). The column is controlled by changing reflux ratio (r) and vapor ratio (s). The number of differential equations is N_T and the number of state variables is $10+4N_T$. The mathematical model is illustrated in eqs (A1)-(A17).

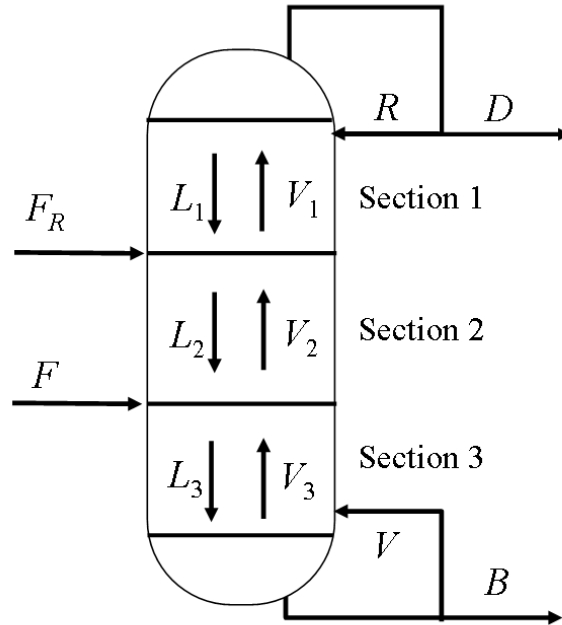


Figure A1: DIB unit

$$M_D \frac{dx_{k,1}}{dt} = V_1 y_{k,2} - (D + L_1) x_{k,1} \quad \text{Top stage} \quad (\text{A1})$$

$$M_T \frac{dx_{i,j}}{dt} = L_1 x_{i-1,j} + V_1 y_{i+1,j} - L_1 x_{i,j} - V_1 y_{i,j} \quad \text{Section 1} \quad (\text{A2})$$

$$M_T \frac{dx_{NF_R,j}}{dt} = L_1 x_{NF_R-1,j} + V_2 y_{NF_R+1,j} - L_2 x_{NF_R,j} - V_1 y_{NF_R,j} + F_R z_{F_R,j} \quad \text{Recycle feed stage} \quad (\text{A3})$$

$$M_T \frac{dx_{i,j}}{dt} = L_2 x_{i-1,j} + V_2 y_{i-1,j} - L_2 x_{i,j} - V_2 y_{i,j} \quad \text{Section 2} \quad (\text{A4})$$

$$M_T \frac{dx_{NF,j}}{dt} = L_2 x_{NF-1,j} + V_3 y_{NF+1,j} - L_3 x_{NF,j} - V_2 y_{NF,j} + F z_j \quad \text{Input feed stage} \quad (\text{A5})$$

$$M_T \frac{dx_{i,j}}{dt} = L_3 x_{i-1,j} + V_3 y_{i+1,j} - L_3 x_{i,j} - V_3 y_{i,j} \quad \text{Section 3} \quad (\text{A6})$$

$$M_B \frac{dx_{NT,j}}{dt} = L_3 x_{NT,j} - B x_{NT,j} - V_3 y_{NT,j} \quad \text{Bottom stage} \quad (\text{A7})$$

The liquid and vapor flowrates throughout DIB column are determined by several algebraic equations, (A8)-(A17).

Liquid phase:

$$B = \frac{[(q+r)F + (q_R+r)F_R]}{(1+r+s)} \quad (\text{A8})$$

$$R = rD \quad (\text{A9})$$

$$L_1 = R \quad (\text{A10})$$

$$L_2 = R + Fq \quad (\text{A11})$$

$$L_3 = R + Fq + F_R q_R \quad (\text{A12})$$

Vapor phase:

$$D = \frac{[(1 - q_R + s)F_R + (1 - q + s)F]}{(1 + r + s)} \quad (\text{A13})$$

$$V = sB \quad (\text{A14})$$

$$V_1 = V + F(1 - q) + F_R(1 - q_R) \quad (\text{A15})$$

$$V_2 = V + F(1 - q) \quad (\text{A16})$$

$$V_3 = V \quad (\text{A17})$$

2 Feed Effluent Heat exchanger

This unit is a typical shell-and tube exchanger. The heat of the output stream from the reactor and input stream to the reactor is exchanged in this unit, as shown in Figure A2. For the performance evaluation, we consider the steady-state model, neglecting the dynamics of the heat exchanger.

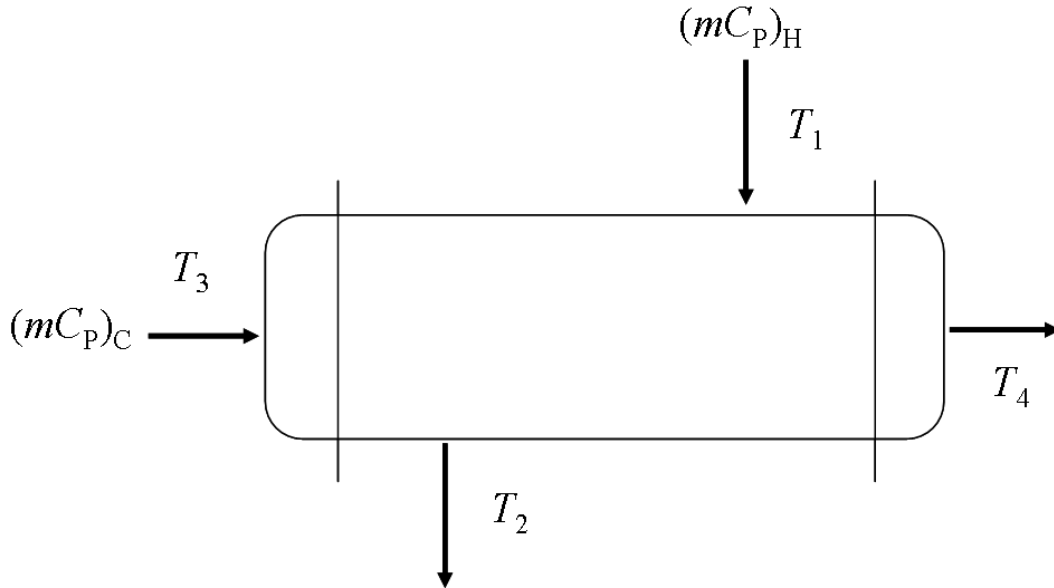


Figure A2: Heat exchanger

$$\begin{aligned}
Q &= (mC_p)_H (T_1 - T_2) = (mC_p)_C (T_4 - T_3) \\
&= UA \frac{(T_1 - T_4) - (T_2 - T_3)}{\ln \frac{T_1 - T_4}{T_2 - T_3}}
\end{aligned} \tag{A18}$$

3 Purge Distillation column

This unit is used to eliminate the heaviest compound (iC_5) from the main stream, which is vaporized and directed to the reactor. Five percent of the input feed is eliminated at the bottom of this unit. Like the DIB column, a tray-by-tray model and a constant-relative-volatility (α) model is used. Reflux ratio is used to control this unit.

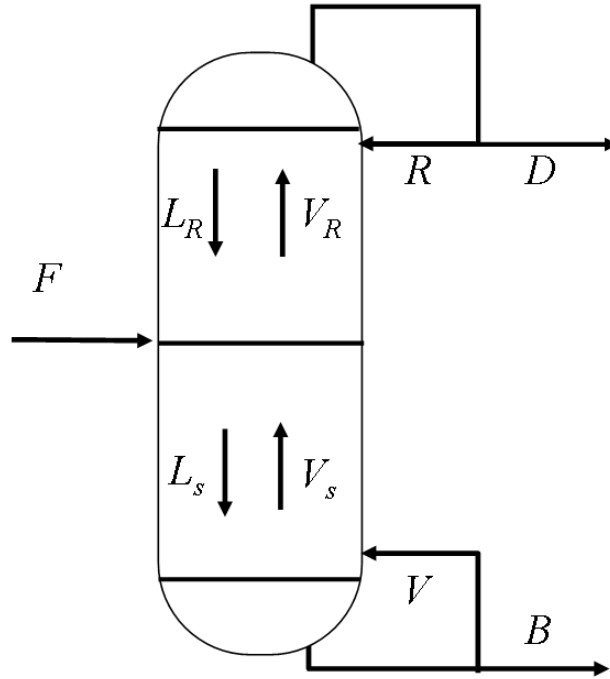


Figure A3: Purge column unit

$$M_D \frac{dx_{1,j}}{dt} = V_R y_{2,j} - (L_0 + D) x_{1,j} \quad \text{Top stage} \tag{A19}$$

$$M_T \frac{dx_{i,j}}{dt} = L_R (x_{i-1,j} - x_{i,j}) + V_R (y_{i+1,j} - y_{i,j}) \quad \text{Rectifying section} \quad (\text{A20})$$

$$M_T \frac{dx_{NF,j}}{dt} = L_R (x_{NF-1,j} - x_{i,j}) + V_S y_{NF+1,j} - L_S x_{NF+1,j} - V_R y_{NF,j} + Fz_{NF,j} \quad \text{Feed} \quad (\text{A21})$$

$$\text{stage} \quad (\text{A22})$$

$$M_T \frac{dx_{i,j}}{dt} = L_S (x_{i-1,j} - x_{i,j}) + V_S (y_{i+1,j} - y_{i,j}) \quad \text{Stripping section} \quad (\text{A22})$$

$$M_B \frac{dx_{NT,j}}{dt} = L_S x_{NT-1,j} - V_S y_{NT,j} - Bx_{NT,j} \quad \text{Last stage of the column} \quad (\text{A23})$$

Liquid phase:

$$B = \frac{(q+r)F}{(1+r+s)} \quad (\text{A24})$$

$$R = rD \quad (\text{A25})$$

$$L_R = R \quad (\text{A26})$$

$$L_S = R + Fq \quad (\text{A27})$$

Vapor phase:

$$D = \frac{(1-q+s)F}{(1+r+s)} \quad (\text{A28})$$

$$V = sB \quad (\text{A29})$$

$$V_R = V + F(1-q) \quad (\text{A30})$$

$$V_S = V \quad (\text{A31})$$

4 Discretized plug flow reactor model

In this unit, nC4 is converted to iC4. This is the first-order irreversible exothermic reaction. To simplify the partial differential model, we discretized the reactor. The reactor

temperature and pressure are controlled by the input feed temperature. Mathematical models are illustrated in eqs (A32)-(A34).

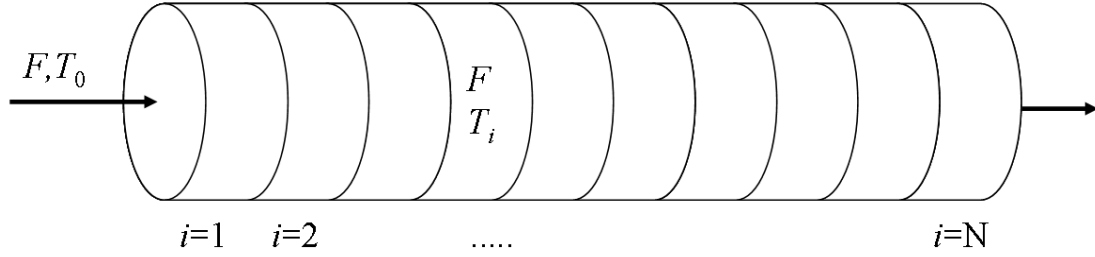


Figure A4: Discretized plug flow reactor

$$\frac{dC_{i,nC_4}}{dt} = \frac{FN}{V_R} (C_{i-1,nC_4} - C_{i,nC_4}) - k_i \quad (\text{A32})$$

$$\frac{dT_i}{dt} = \frac{FN}{V_R} (T_{i-1} - T_i) - \frac{\lambda k_i}{\rho C_P} \quad (\text{A33})$$

$$k_i = C_{i,nC_4} k_0 e^{-E/RT_i} \quad (\text{A34})$$

APPENDIX B- INTRODUCTION TO GLOBAL TERRAIN METHOD

Summary

The purpose of this report is to implement and test the global terrain method using object oriented programming code written in C++ and Matlab to test the effectiveness of the method in finding extrema. The methodology is based upon the notion of intelligent movement along the ridges and valleys of a particular function. The movement encompasses a trust region method for downhill movement, and a Newton-like method in addition to a sequential quadratic programming corrector step for uphill movement. This method allows for the determination of all solution points (i.e. minimum, maximum, and saddle points) of the objective function. Several case studies involving theoretical and practical chemical engineering applications were run in which all-possible solutions were obtained.

1 Global Terrain Method

Lucia and Yang have also developed a deterministic global optimization technique known as the global terrain method that is based off a method from several decades ago (Lucia and Yang, 2002). Lucia's method is a robust and novel technique of finding all solution points of a least-squares objective function. The method is focused on the intelligent movement from one solution to another along valleys and ridges of a least-squares environment.

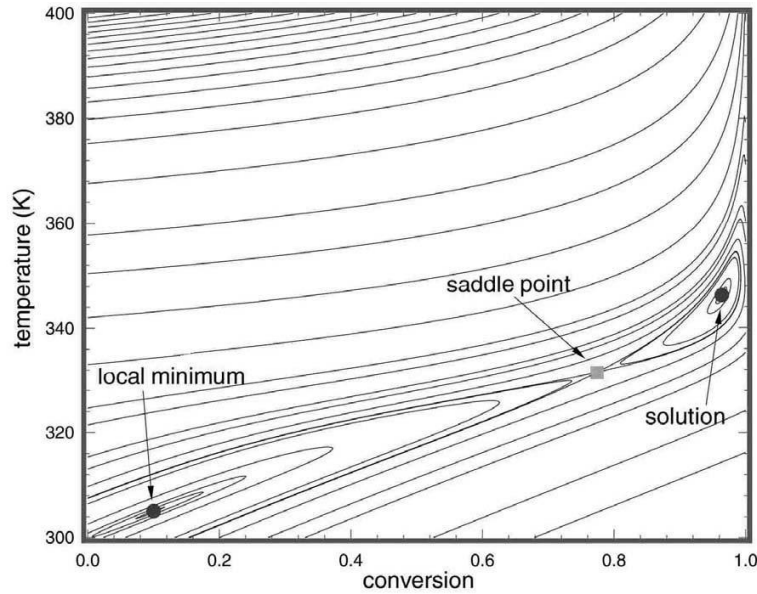


Figure B.1: Connectedness of solution points for a nonlinear continuous stirred tank reactor. (Lucia and Yang, 2002)

As shown in Figure B.1, solution points are “connected” to one another through valleys and ridges. This connectedness holds true regardless of the dimension of the function. The main focus of the global terrain method is to exploit this connectedness of the least-squares landscape in order to find all physically meaningful solutions to the objective function by following the terrain from point to point. It is easiest to imagine the valleys and ridges as “lines with special properties” and following a terrain path as tracing along these “lines”. The main assumptions that the global terrain method makes are:

- Solution and singular points are smoothly connected
- The terrain (i.e. valleys, ridges, ledges, etc.) provide this connectedness
- The terrain is composed of a collection of constrained extrema over a set of level curves
- The terrain can be characterized as solutions to eigenvector problems
- Newton-like vectors naturally flow along the terrain

In order to appreciate the global terrain method, the governing equations must be explained.

2 Methodology

2.1 Governing Equations

Suppose F is a vector function defined on the set of all real numbers by Eqn. (B.1), and the least-squares function is $F^T F$. Eqn. (B.1) shows that F is a set of n functions set equal to zero. The global terrain method is an equation solving technique because it solves for the solutions to Eqn. (A.1).

$$F = \begin{bmatrix} f_1 = 0 \\ \vdots \\ f_n = 0 \end{bmatrix} \quad \text{B.1}$$

The terrain (i.e. valleys, ridges, ledges, etc.) of the least-squares function can then be defined by V , to a set of general nonlinearly constrained optimization problems.

$$V = \left\{ \begin{array}{l} \text{Min}_x g^T g \\ \text{s.t. } F^T F - L = 0, \forall L \in \mathbf{L} \end{array} \right\} \quad \text{B.2}$$

where $g = 2J^T F$ is defined as the gradient of the least-squares objective function (derivation in Sect. 7.1.1), J is the Jacobian of F , and L is a level curve of the least-squares function while \mathbf{L} is the collection of such L . Eqn. (B.2) is the backbone of the global terrain method, and is vital to implement the technique. Two conditions must be satisfied for the technique to work. The first being that $F^T F = L$ is satisfied the second being that Eqn. (B.3) is satisfied.

$$Mg - \lambda g = 0 \quad \text{B.3}$$

where M is defined by $M = J^T J + \sum H_i F_i$, while H_i is the Hessian of F_i , and λ is a Lagrange multiplier involved with the level constraint L .

Eqn. (B.2) is proved by introducing Karush Kuhn Turcker conditions. The Lagrangian function of Eqn. (B.1) is then defined by Eqn. (B.4), and its gradient defined as Eqn. (B.5).

$$L(x, \lambda) = g^T g - 2\lambda F^T F \quad \text{B.4}$$

$$\begin{aligned} \frac{\partial L(x, \lambda)}{\partial x} &= \nabla g^T g - 2\lambda \nabla F^T F = 0 \\ 2Mg - 2\lambda g &= 0 \end{aligned} \quad \text{B.5}$$

$$Mg - \lambda g = 0$$

2.2 Movement Between Stationary Points

As Lucia showed in his paper, stationary points (i.e. minima, maxima, saddle points) are connected to one another through the valleys of the least-squares function. If these valleys are traced out, it is possible to find where all these solution points are located. The global terrain method uses this tracing feature by moving both up and down the valleys. To initiate movement, the eigendirection of the Hessian matrix is necessary in order to follow the direction of the largest curvature ridges. These movements are described in the following subsections.

2.2.1 Initial Movement

A crucial aspect of the global terrain method is the effective calculation of eigenvalues and eigenvectors of the objective function. These eigenvectors allow for the determination of the direction of steps to take between solution points. Analytically, eigenvalues are determined by solving for the determinant of Eqn. (B.6), where λ is the eigenvalue. Once λ is obtained, the eigenvectors are easily determined, and will direct the path from one solution point to another.

$$(A - \lambda I) = 0 \quad \text{B.6}$$

At any stationary point, the direction of the valley is discovered from the eigendirection of the Hessian matrix of the least squares function. Subsequent movement is then determined using this eigen-information of the objective functions. Figure B.2a shows the Hessian eigendirection at a stationary point (i.e. energy function). Locally, the eigendirection of the Hessian points towards the valley and the eigenvector is the answer to Eqn. (B.3). Figure B.2b also shows the eigendirections of the Hessian on the least-squares terrain.

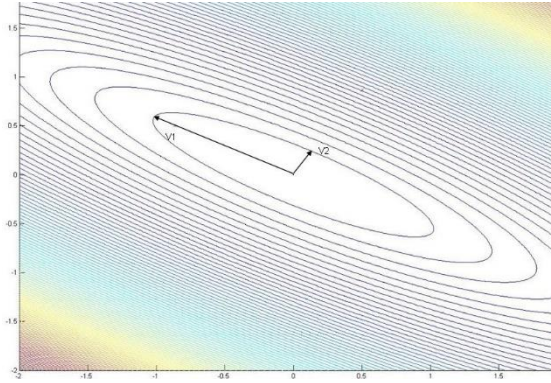


Figure B.2a: Energy function of Hessian matrix at the minimum point

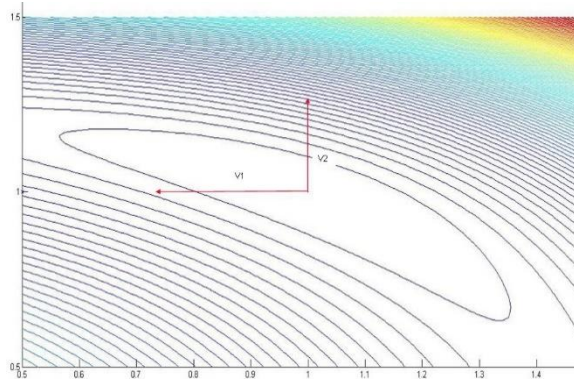


Figure B.2b: Eigendirection of Hessian matrix at the minimum point

From these graphs it is apparent that the eigendirection of the Hessian is directing the valley. It is for this reason that the eigendirection of the Hessian is used to initiate movement.

All eigenvalues are either positive or negative, and the sign of the eigenvalue is crucial. The eigendirection associated with a positive value is uphill (the value of $F^T F$ at x is less than the value of $F^T F$ at $x + \Delta x$), whereas negative values are downhill. Downhill movement from a maximum point or saddle point to another solution point occurs along the negative curvature. The magnitude of the step in this negative eigendirection is,

$$\alpha = (-2F^T F / \xi)^{\frac{1}{2}} \quad \text{B.7}$$

where ζ is the most negative eigenvalue of the Hessian matrix of the least-squares function. Uphill movement is initialized in the eigendirection corresponding with a positive eigenvalue.

2.2.2 Downhill Movement

Downhill movement occurs at a local maximum point or possibly a saddle point. The initial point (i.e. maximum or saddle point) is of greater value than its surrounding points, or level curves, and the downhill movement goes in a direction of decreasing value of the level curves. A trust-region method is used for the downhill movement. This method encompasses a Newton method along with a steepest descent direction method. The main advantage of this method is that it can be used when Jacobians and Hessians are singular. A scalar, γ , is the radius of the maximum allowable change in variables (i.e. $|p| \leq \gamma$). This change is the step size, denoted by p , and is shown in Eqn. (B.8),

$$p = -(J^T \cdot F + \mu I)^{-1} \cdot J^T \cdot f(x^k) \quad \text{B.8}$$

where f is the least-squares function, μ is the nonnegative Kuhn Tucker multiplier, and x^k is the starting location.

2.2.3 Uphill Movement

Uphill movement is initiated at either a minimum point or a saddle point. This is because the level curves surrounding the initial point are of greater value than the starting point. The uphill movement will move from a level curve of lesser value to one of greater value. A different method is needed for uphill travel because the trust region method will only find locations of lesser value than the current location. Uphill movement is composed of two parts, a predictor step, and a corrector step. The predictor step predicts the next ridge's location, but in a nonlinear function, the ridge is rarely exactly where it is predicted to be. Depending on the step size, the difference between the correct and actual location of

the ridge could be enough to initiate a corrector step. The corrector step will move the point back onto the ridge. A Newton-like method is used for the predictor step, and a SQP method is implemented for the corrector step. The initial movement is still produced using the eigendirection, but the predictor/corrector steps will follow initial movement until another solution is found or the boundary is found.

A detailed visualization of the predictor and corrector step is depicted in Figure B.3. The movement begins from the ridge point P_1 , which lies at the maximum curvature on the level set L_1 . At P_1 , the Newton direction is computed to the next ridge point, P_2 . However, P_2 does not lie exactly on L_2 , so a corrector step is necessary. The corrector step maximizes the curvature, and the solution of the corrector step is C_2 , which is an optimal ridge point on L_2 . Through this sequence the next solution point, S_2 , can be found. Details of the two steps aforementioned will be represented in the next subsections.

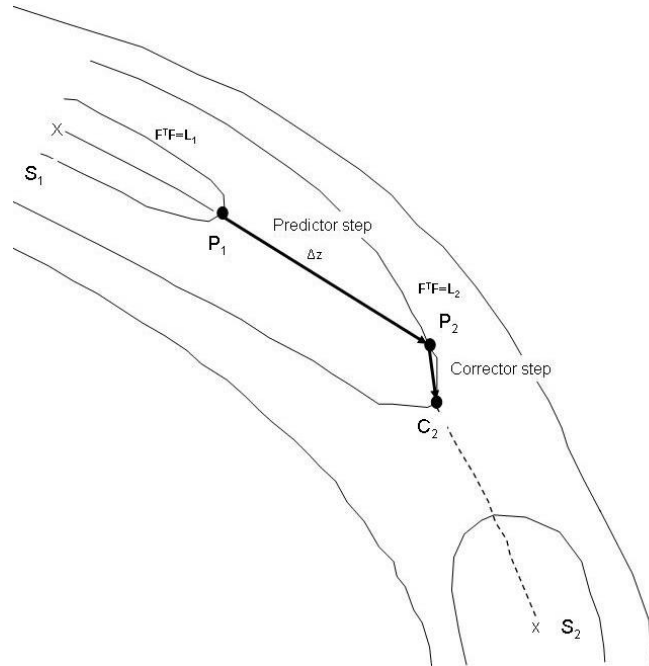


Figure B.3: Illustration of uphill predictor-corrector step

2.2.4 Predictor Step

The uphill Newton-like step vector field is,

$$\Delta z_N = J^{-1}F \quad \text{B.9}$$

where z is a vector of unknown variables.

A suitable increase in size of the uphill movement α_p is,

$$\alpha_p = \min \left\{ \|J^{-1}F\|, 2\varepsilon |(J^T F)^T \Delta_N| / \left| \Delta_N^T \left(\sum F_i H_i \right) \Delta_N \right| \right\} \quad \text{B.10}$$

where Δ_N is the normalized uphill Newton-like direction and given by and ε is some tolerance. Uphill predictor steps are computed from

$$\Delta z_p = \alpha_p \Delta z_N \quad \text{B.11}$$

2.2.5 Corrector Step

The uphill Newton-like step follows the terrain fairly well, but it can drift off course due to the nonlinearity of the objective function. In order to account for this drift, and return the step back to a level curve, Eqn. (B.12) is solved. In order to accomplish this, a sequential quadratic programming (SQP) method is used by iteratively solving,

$$\begin{aligned} \text{Min } & g^T H \Delta_c + \frac{1}{2} \Delta_c^T M \Delta_c \\ \text{Such that } & g^T \Delta_c = -(F^T F - L) \end{aligned} \quad \text{B.12}$$

where M is the Hessian matrix of the Lagrangian function, λ is a Lagrange multiplier, and the remaining variables depend on z . Boundaries can be applied easily to Eqn. (B.12). The size of the corrector steps, Δ_c , can be controlled by forcing improvement in the gradient norm through line searching or trust region methods, and thus,

$$\Delta z_c = \alpha_c \Delta z_N \quad \text{B.13}$$

Due to gradient directions being orthogonal to the tangent of level curves, and are only exactly collinear in valleys or on ridges, a criterion for implementing the corrector step is given by inequality condition, (B.14).

$$\cos \theta = \Delta_g^T \Delta_N < \theta \quad \text{B.14}$$

In (B.14), q is the angle between the normalized gradient D_g and the normalized Newton-like step. Θ is a number close to 1, and usually 0.95 will suffice.

2.3 Saddle Detection

Condition (B.15) is a general criterion for the termination of uphill movement. When the criterion is met the movement is terminated and the point is dubbed a ‘nearby’ singular point.

$$\Delta_N^T J^T J \ll \Delta_N^T \left(\sum F_i H_i \right) \Delta_N \quad \text{B.15}$$

When the uphill movement determines where this ‘nearby’ singular point is located the algorithm logic switches back to ‘equation-solving’ to look for the actual singular point using an acceleration technique. Either quadratic acceleration, Krylov subspace methods such as a conjugate gradient, or null space rotation techniques are used. The uphill exploration only provides a ‘good’ initial guess for the determination of solution points.

2.4 Termination

The termination of the global terrain method is the most controversial aspect of the technique because it excludes some of the eigendirections. Lucia suggested that a total of only four directions (i.e. \pm most negative eigendirection and \pm least positive eigendirection) would be investigated. The algorithm terminates in each of those directions when it either collides with a boundary or discovers another solution point. When a collision occurs, a previously discovered point is returned to and a new direction is explored. Once all the possible directions for all stationary points are explored, the program terminates.

3 Whole procedure of Global Terrain method

A flowchart of global terrain method's algorithm is displayed below as Figure B.4. This figure depicts the logic used in the algorithm and shows how the global terrain method works overall.

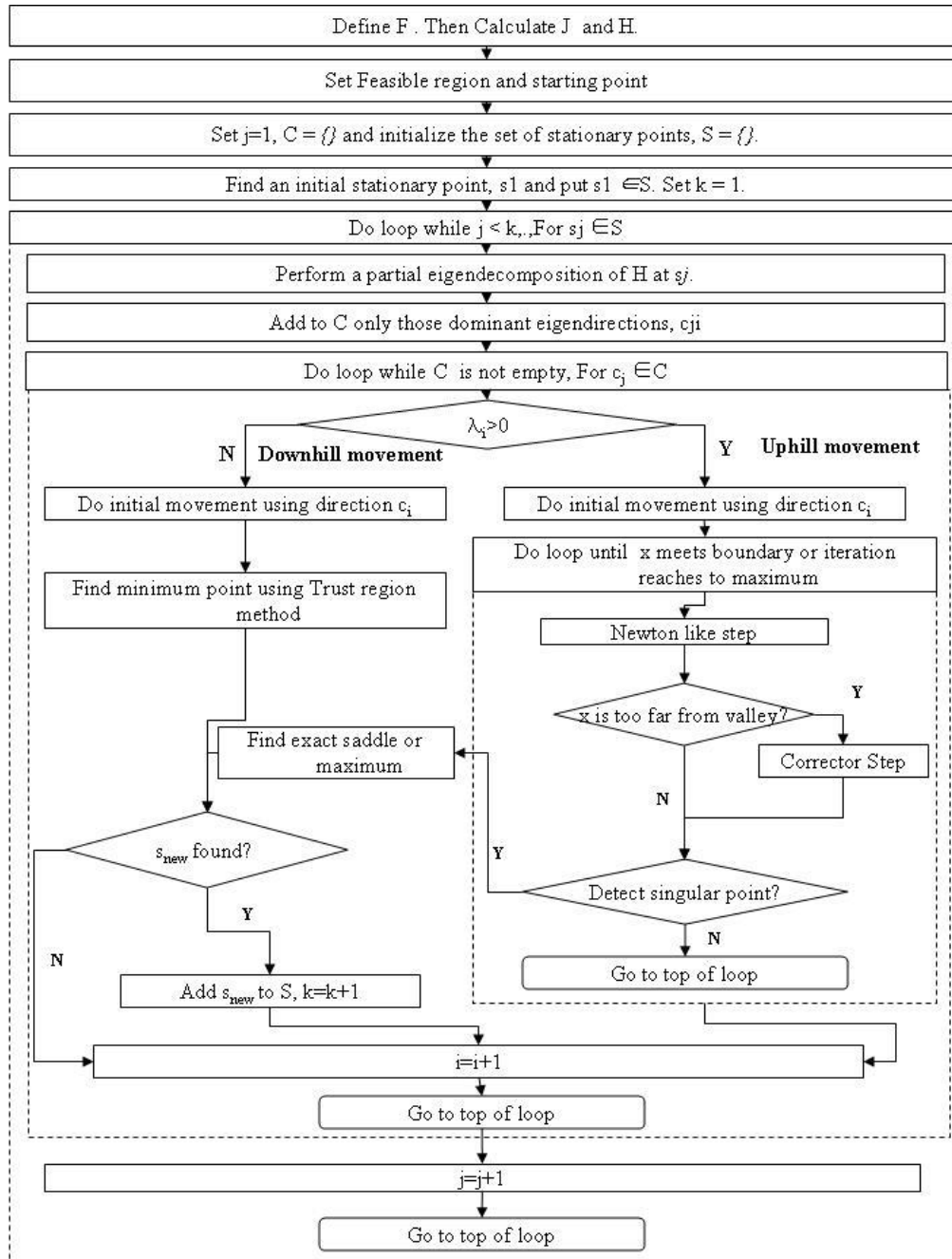


Figure B.4: Flowchart of the global terrain method's algorithm

4 Applications

In this section are several case studies that tested the global terrain method in theoretical and chemical engineering applications. Additional examples can be found in Section 6.4.

4.1 Theoretical Applications

4.1.1 Example 1 – Nonlinear Equations

This example shows how the global terrain method can be used to solve a system of nonlinear equations. The set of equations is,

$$F = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2^2 - 2 \\ e^{x_1-1} + x_2^3 - 2 \end{pmatrix} \quad \text{B.16}$$

And the feasible region is,

$$\begin{aligned} -1.5 < x_1 < 1.8 \\ -1.2 < x_2 < 1.35 \end{aligned} \quad \text{B.17}$$

As shown in Fig. 6, the system has 3 minimum points (M), 4 saddle points(S) and one maximum point(X). The objective of this case is to find all of these stationary points and saddle points using global terrain method.

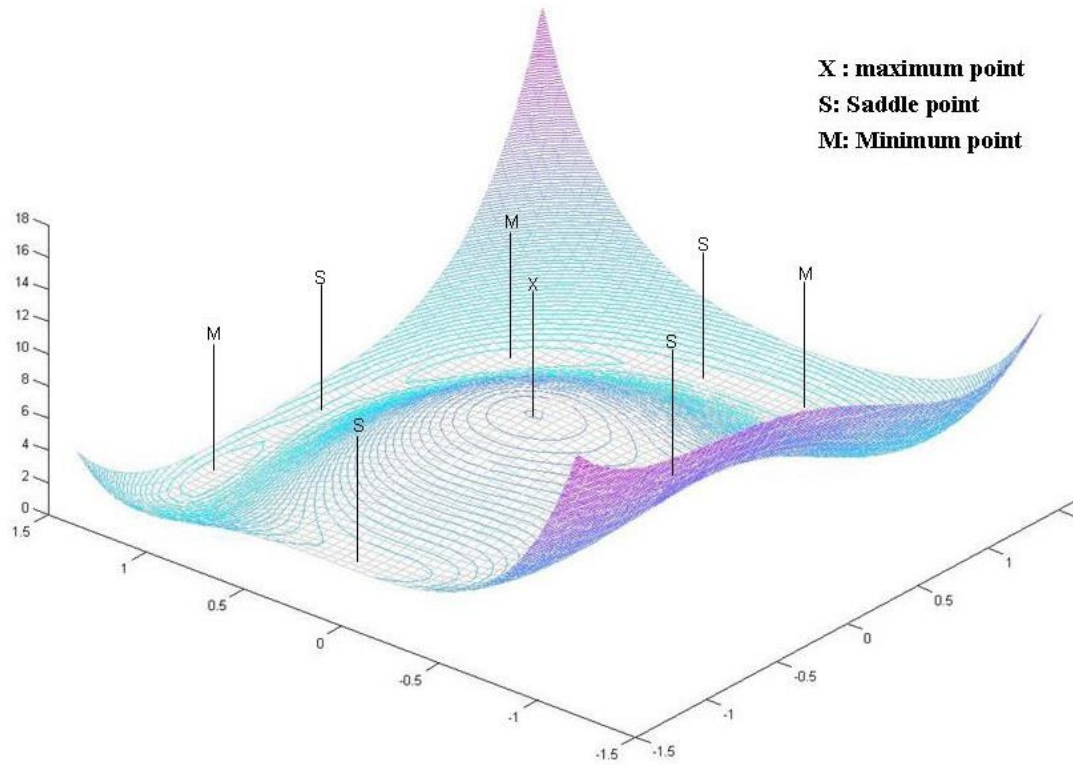


Figure B.5: 3D Contour plot of Eqn. (16)

Step 1: Calculating Jacobian and Hessian Matrix

First, the Jacobian and Hessian matrices of F must be calculated. By the definition the Jacobian matrix of this example is,

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 & 2x_2 \\ e^{x_1-1} & 3x_2^2 \end{bmatrix} \quad \text{B.18}$$

And the Hessian matrices are

$$H_1 = \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \frac{\partial^2 f_1}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f_1}{\partial x_2 \partial x_1} & \frac{\partial^2 f_1}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad \text{B.19}$$

$$H_2 = \begin{bmatrix} \frac{\partial^2 f_2}{\partial x_1^2} & \frac{\partial^2 f_2}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f_2}{\partial x_2 \partial x_1} & \frac{\partial^2 f_2}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} e^{x_1-1} & 0 \\ 0 & 6x_2 \end{bmatrix} \quad \text{B.20}$$

Step 2: Finding the first stationary point from the starting point

The starting point, S_0 of this example is (1,1.12). We have to find the closet minimum point from starting point using Newton method or some other root solving method. In Matlab, this is calculated using the function ‘fsolve’,

```
options=optimset('Display','off','LargeScale','off');  
[x,fvel,flag]=fsolve(@UserFunction,x_initial,options);
```

where $x_initial$ is the starting point and “UserFunction” is implementation code of F.

Thus the definition of “UserFunction” in this example is,

```
function f=UserFunction(x) % set of F  
r(1)=x(1)^2+x(2)^2-2; %f1  
r(2)=exp(x(1)-1)+x(2)^3-2; %f2  
f=r';
```

From the starting point (1,1.12) , the first stationary point, S_1 (1.0000, 1.0000) was found.

Step 3: Uphill movement from S_1

Eigenvalues of the second order approximation of the least square function at the point S_1 , are $\lambda_1=0.9377$, $\lambda_2=17.0623$ and each eigenvectors are $V_1(-0.8649,0.5019)$, $V_2(0.5019,0.8649)$. Four directions are to be explored: $+V_1, -V_1, +V_2, -V_2$. All eigenvalues are positive, so only uphill movement is done in this point S_1 (All positive eigenvalue at some point means that this point is minimum point).

The result of exploration of each direction is,

Table B.1: Direction of exploration from S1 and result

Eigenvalue	Eigendirection	Result
0.9377	$+V_1(-0.8649, 0.5019)$	Near Saddle Point (0.1277, 1.2261)
	$-V_1(0.8649, -0.5019)$	Near Saddle Point (1.4388, 0.3785)
17.0623	$+V_2(0.5019, 0.8649)$	Boundary Collision
	$-V_2(-0.5019, -0.8649)$	Boundary Collision

With the directions $\pm V_1, V_2$ we found near the saddle points which satisfy Eqn. (B.15). But these points are not the actual saddle points. At these points we found the real saddle points using this Matlab code,

```
[x_saddle, fvel, flag]=fsolve(@UserResidualGridentFunction,x,options);
```

where x is 'near the saddle point', 'UserResidualGridentFunction' is the gradient of least square function. The real saddle points are S_2 (0.1117, 1.2275) for $+V_1$, S_3 (1.4435, 0.3598) for $-V_1$.

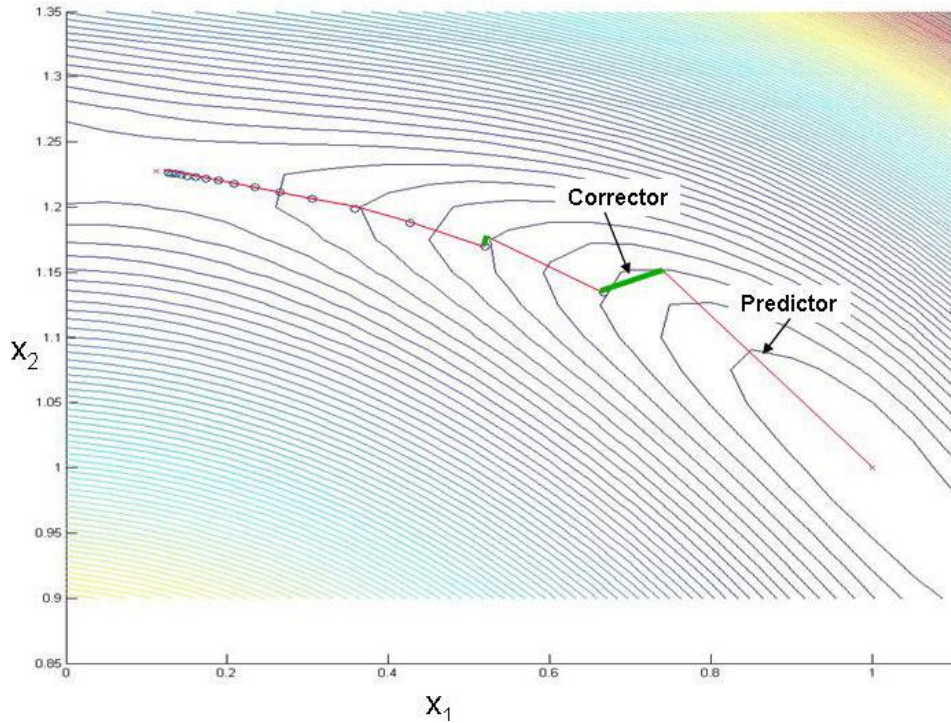


Figure B.6: Predictor and corrector step for Example 1

Figure B.6 shows the predictor step and corrector step. For the predictor step, a Newton-like method is used and for the corrector step, a SQP method is used. In Matlab code, the function 'fmincon' is used to implement the corrector step.

```
options=optimset('Display','off','LargeScale','off');
A=[];
b=[];
Aeq=[];
beq=[];
lb=[];
ub=[];
Residual_Vector=UserFunction(x);
L=Residual_Vector'*Residual_Vector;
xc=fmincon(@UserObjectFunction,x,A,b,Aeq,beq,lb,ub,@UserConstraintFunction,options,L);
```

Where UserObjectFunction is the function which returns the value of $g^T g$ at the point x .

Step 4. Finding a extrema point from saddle point S_2

$S_2(0.1117, 1.2275)$ found from S_1 is a saddle point. The eigenvalues at this point are $\lambda_1 = -0.8400$, $\lambda_2 = 27.6290$. Like Step 3, it was explored for 4 directions. But this point has all negative eigenvalues, so only downhill movement was done. The results are shown in Table B.2.

Table B.2: Direction of exploration from S_2 and result

eigenvalue	direction	result
-0.8400	$+V_1(-0.9964, 0.0849)$	Minimum (-0.7137, 1.2209)
	$-V_1(0.9964, -0.0849)$	Minimum (1.0000, 1.0000)
27.6290	$+V_2(0.0849, 0.9964)$	Boundary Collision
	$-V_2(-0.0849, -0.9964)$	Boundary Collision

With $+V_1$, a minimum point is found. The trust region method was used for downhill movement. To implement this in Matlab, we used the function 'fsolve'. Before using the function 'fsolve', initial movement must be done. Figure B.7 shows initial movement step and next secant method step.

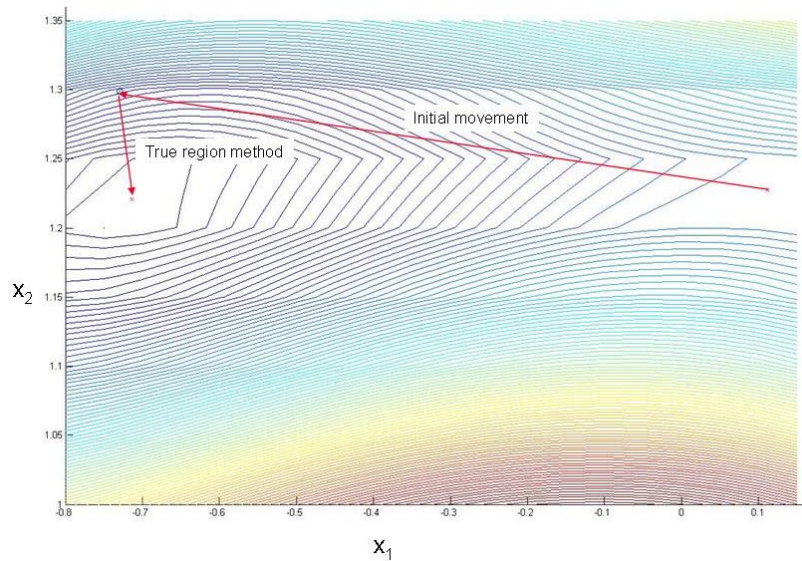


Figure B.7: Downhill movement of Example 1

Summary of results

In these ways as explained above, we got 3 minimum points, 4 saddle points and 1 maximum point. Figure B.8 shows the sequence of the algorithms progress and solution coordinate. From S_1 , S_2 and S_3 were found. S_2 found S_4 and S_3 found S_5 . From S_4 , S_6 was found and from S_5 , S_7 was found. S_8 maximum point was found using connectedness with S_7 . Table A.3 shows coordinates of each point and its corresponding eigenvalue. For a minimum point, the eigenvalues are positive, for saddle point one is negative and the other is positive, and a maximum point has only negative eigenvalues. Figure B.9 shows a computational tree of solutions and eigendirections.

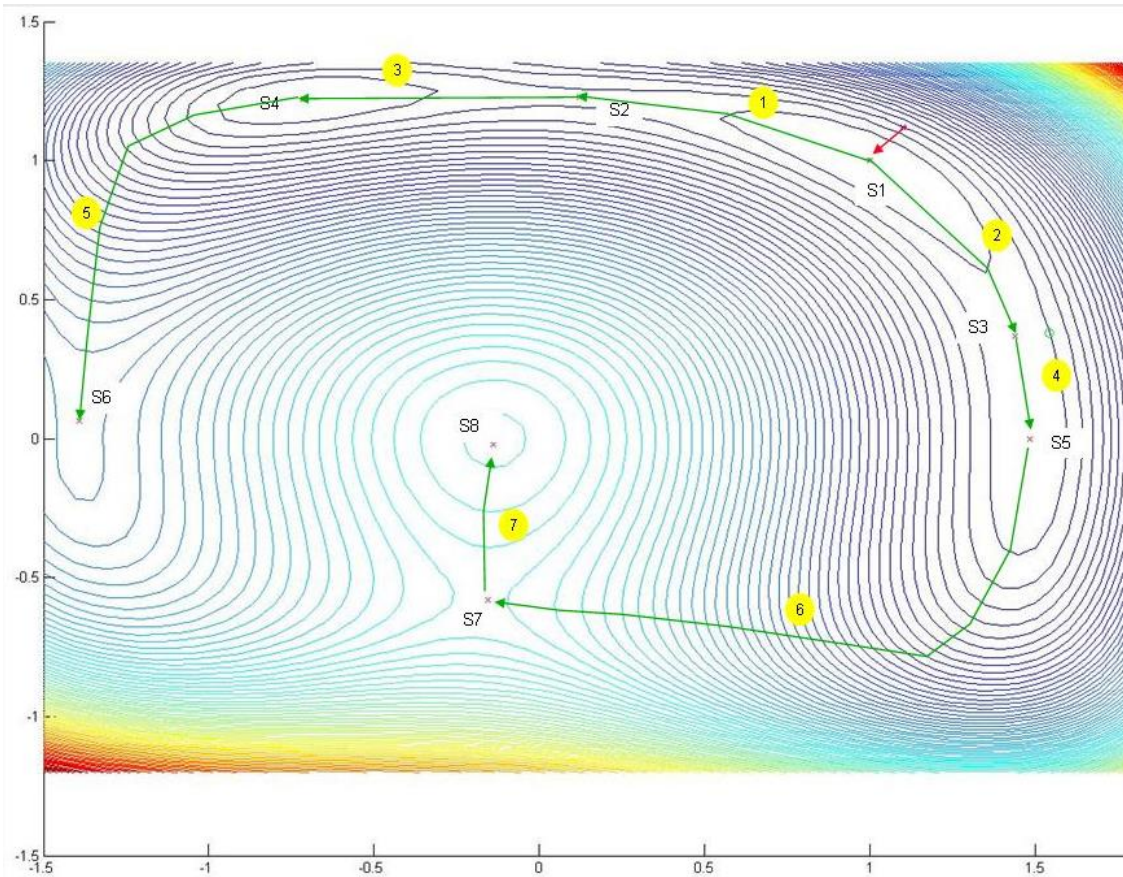


Figure B.8: Connectedness of stationary points for Example 1

Table B.3: The results of example 1

No	Position	Eigenvalue	Solution Type
S ₁	1.0000,1.0000	0.9377,17.0623	Minimum
S ₂	0.1117,1.2275	-0.8400,27.6290	Saddle
S ₃	1.4435,0.3598	-0.4130,11.2293	Saddle
S ₄	-0.7137,1.2209	1.7733,26.2553	Minimum
S ₅	1.4851,0.0000	0.4109,11.2609	Minimum
S ₆	-0.1776,-0.5780	-3.6840,5.6652	Saddle
S ₇	-1.3919,0.0000	-0.1254,7.4576	Saddle
S ₈	-0.1360,0.0000	-4.3250,-3.9630	Maximum

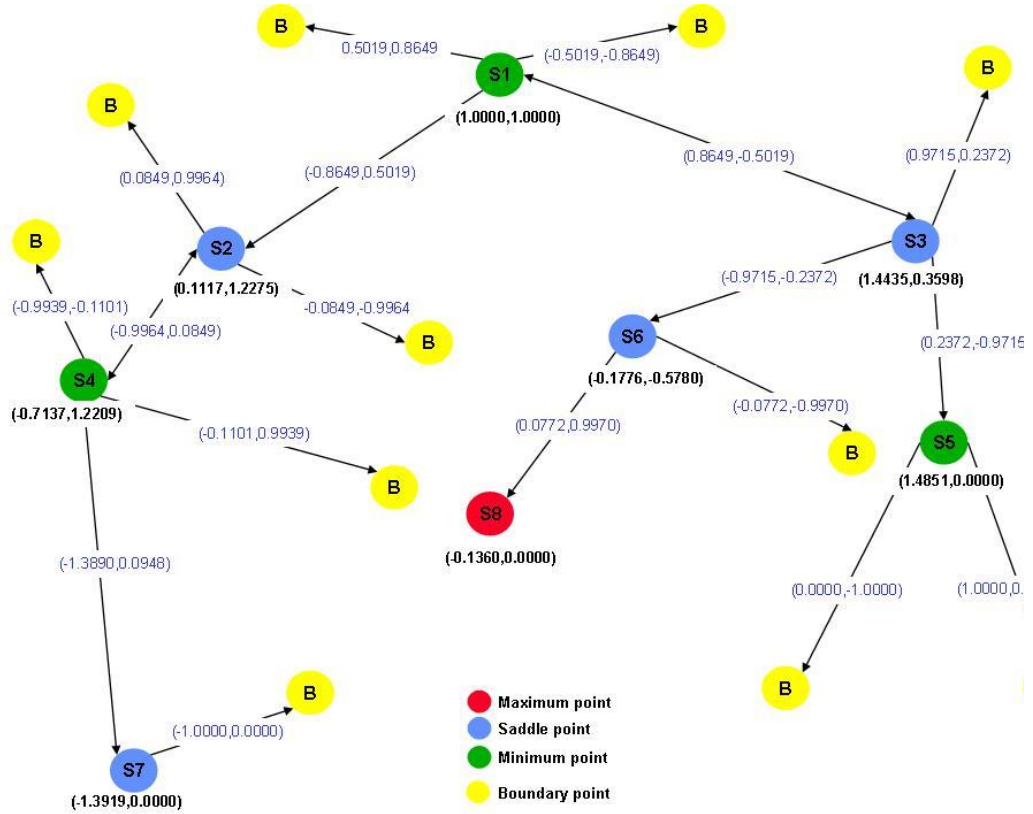


Figure B.9: Computational tree for Example 1

Numerical derivative method

We also used numerical derivative method to examine whether the global Terrain Method can function with less accurate Jacobian and Hessian information. The result of this was the same as the analytical method but calculation time (CPU time = 107.6548 s) was 2.2 times longer than the analytical method (CPU time = 10.2961 s).

The code for implementing the Jacobian matrix is,

```
function [J]=jacobian(func,x)
% computes the Jacobian of a function
n=length(x);
fx=feval(func,x);
eps=1.e-8; % could be made better
xperturb=x;
for i=1:n
xperturb(i)=xperturb(i)+eps;
J(:,i)=(feval(func,xperturb)-fx)/eps;
xperturb(i)=x(i);
end
```

And the code for the Hessian is,

```
function [H1,H2]=UserHessian(x)
alpha=0.0001;
J1=UserGridentFunction([x(1)+alpha,x(2)]);
J2=UserGridentFunction([x(1)-alpha,x(2)]);
J3=UserGridentFunction([x(1),x(2)+alpha]);
J4=UserGridentFunction([x(1),x(2)-alpha]);

H1(1,1)=(J1(1,1)-J2(1,1))/2/alpha;
H1(1,2)=(J1(1,2)-J2(1,2))/2/alpha;
H1(2,1)=H1(1,2);
H1(2,2)=(J3(1,2)-J4(1,2))/2/alpha;
```

```

H2 (1,1)=(J1 (2,1)-J2 (2,1)) /2/alpha;
H2 (1,2)=(J3 (2,1)-J4 (2,1)) /2/alpha;
H2 (2,1)=H2 (1,2) ;
H2 (2,2)=(J3 (2,2)-J4 (2,2)) /2/alpha;

```

Figure B.10 shows the difference of step points between the trajectory of the analytical method and the numerical method. This trajectory is from S_1 to S_3 . There is only a very small difference between both methods. Because the result was the same, we know that this difference does not critically affect finding solutions.

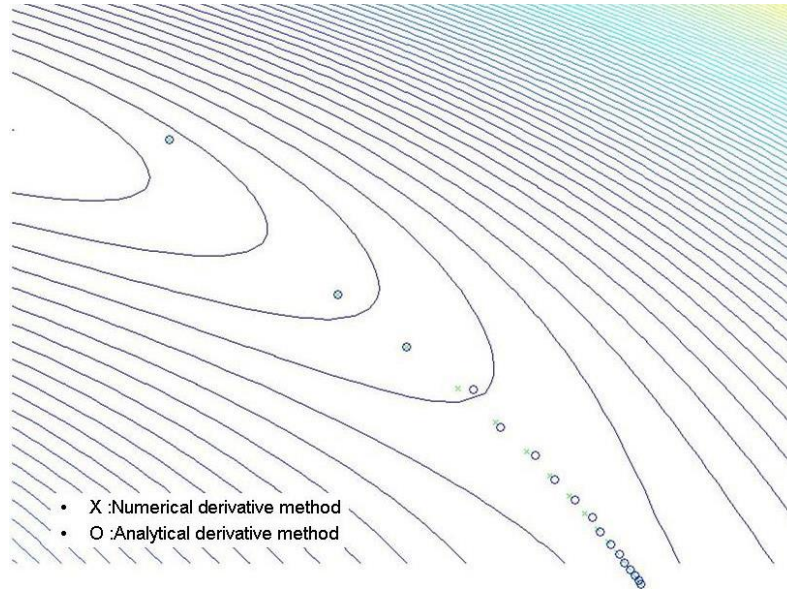


Figure B.10: Difference of pathways between analytical and numerical derivative method in example1

4.1.2 Example 2 – Polynomial Complex Roots

This case study shows how the Global Terrain Method can be used to find roots to a polynomial. Consider the Eqn. (B.21), which has the roots 1, $2+j$, and $2-j$.

$$x^3 - 5x^2 + 9x - 5 = 0 \quad \text{B.21}$$

If the roots are assumed to have the form $x + yj$, then Eqn. (B.22) must also be true.

$$(x + yj)^3 - 5(x + yj)^2 + 9(x + yj) - 5 = 0 \quad \text{B.22}$$

$$(x^3 - 3xy^2 - 5x^2 + 5y^2 + 9x - 5) + (3x^2y - y^3 - 10xy + 9y)j = 0$$

In order for Eqn. (B.22) to be zero, the real and imaginary parts of the root must be equal to zero. This allows for the definition of F as Eqn. set (B.23), and the corresponding Jacobian and Hessians as follows.

$$F = \begin{pmatrix} x^3 - 3xy^2 - 5x^2 + 5y^2 + 9x - 5 = 0 \\ 3x^2y - y^3 - 10xy + 9y = 0 \end{pmatrix} \quad \text{B.23}$$

$$J = \begin{bmatrix} 3x^2 - 10x - 3y^2 + 9 & 10y - 6xy \\ 10y - 6xy & 3x^2 - 10x - 3y^2 + 9 \end{bmatrix} \quad \text{B.24}$$

$$H_1 = \begin{bmatrix} 6x - 10 & -6y \\ -6y & 6x - 10 \end{bmatrix} \quad \text{B.25}$$

$$H_2 = \begin{bmatrix} -6y & 10 - 6x \\ 10 - 6x & -6y \end{bmatrix} \quad \text{B.26}$$

Using equations B.23-B.26 with the global terrain method starting at point (0.6, 0), the following contour plot in Figure B.11 shows the results.

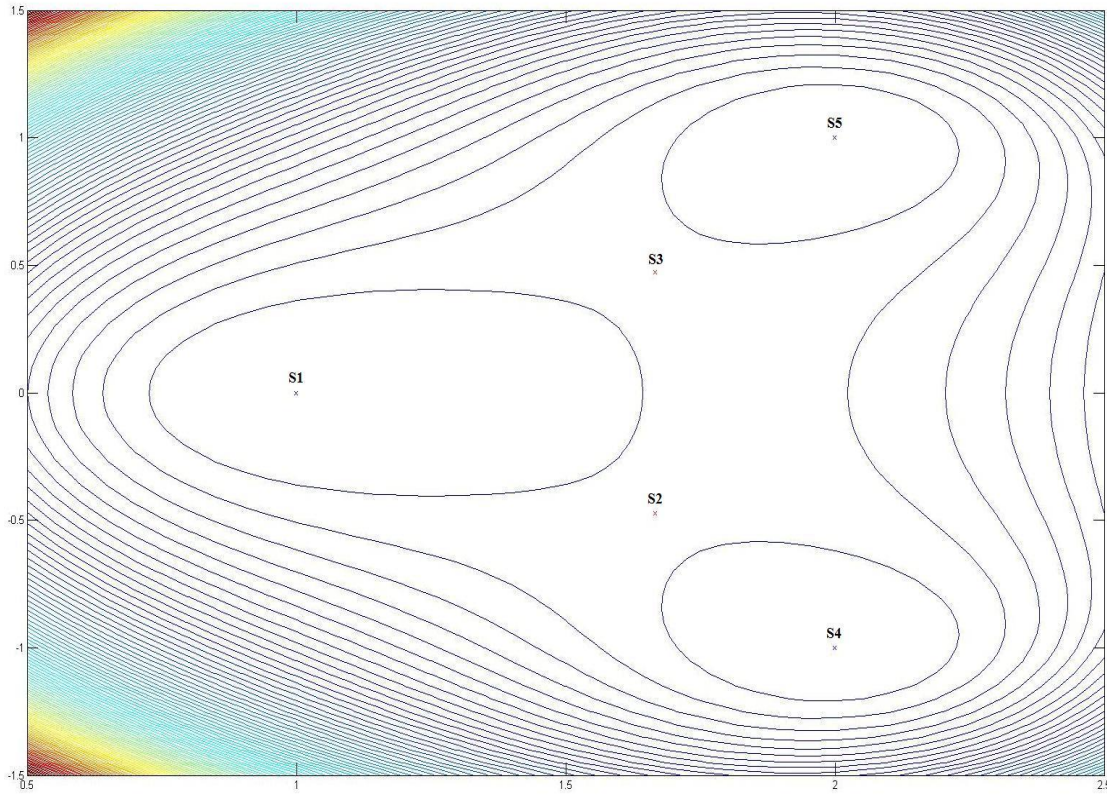


Figure B.11: Contour plot containing results for example 2

Table 4 shows the positions for all results, and the minimum values obtained coincided with the expected results. The total CPU computational time was 4.2744 seconds.

Table B.4: Results of example 2

No	Position	Eigenvalue	Solution Type
S ₁	1.0000,0.0000	4.0000,4.0000	Minimum
S ₂	1.6667,-0.4714	-2.1773,2.1773	Saddle
S ₃	1.6667,0.4714	-2.1773,2.1773	Saddle
S ₄	2.0000,-1.0000	8.0000,8.0000	Minimum
S ₅	2.0000,1.0000	8.0000,8.0000	Minimum

1.1 Chemical Engineering Applications

4.1.3 Example 3 – Nonlinear Continuous Stirred Tank Reactor

This example was obtained from Lucia's paper (Lucia and Yang, 2002). System (B.27) contains equations that describe a nonlinear continuous stirred tank reactor with lumped parameters.

$$\begin{aligned}
 F_1(x, T) &= 120x - 75k(1 - x) = 0 \\
 F_2(x, T) &= x(873 - T) - 11(T - 300) = 0 \\
 k &= 0.12 \exp[12581(T - 298) / (298T)]
 \end{aligned}
 \tag{B.27}$$

$k = 0.12 \exp[12581(T - 298) / (298T)]$ The Jacobian and Hessians are defined below.

$$J = \begin{bmatrix} 120 + 75k & -75k \left(\frac{12581(T - 298)}{298T^2} - \frac{12581}{298T} \right) (x - 1) \\ 873 - T & -x - 11 \end{bmatrix}
 \tag{B.28}$$

$$\begin{aligned}
H_{1,11} &= 0 \\
H_{1,12} &= -75k \left(\frac{12581(T-298)}{298T^2} - \frac{12581}{298T} \right) \\
H_{1,21} &= H_{1,12} \\
H_{1,22} &= 75k \left[\left(\frac{12581(T-298)}{149T^2} - \frac{12581(T-298)}{149T^2} \right) (x-1) \right. \\
&\quad \left. + \left(\frac{12581(T-298)}{298T^2} - \frac{12581(T-298)}{298T} \right)^2 (x-1) \right] \\
H_1 &= \begin{bmatrix} H_{1,11} & H_{1,12} \\ H_{1,21} & H_{1,22} \end{bmatrix}
\end{aligned} \tag{B.29}$$

$$H_2 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{B.30}$$

The conversion bounds are $0 < x < 1$, the temperature (in Kelvin) range is $298 < T < 450$, and the starting point is (0.5, 330). Figure B.12 displays the results of example 3 using a contour plot.

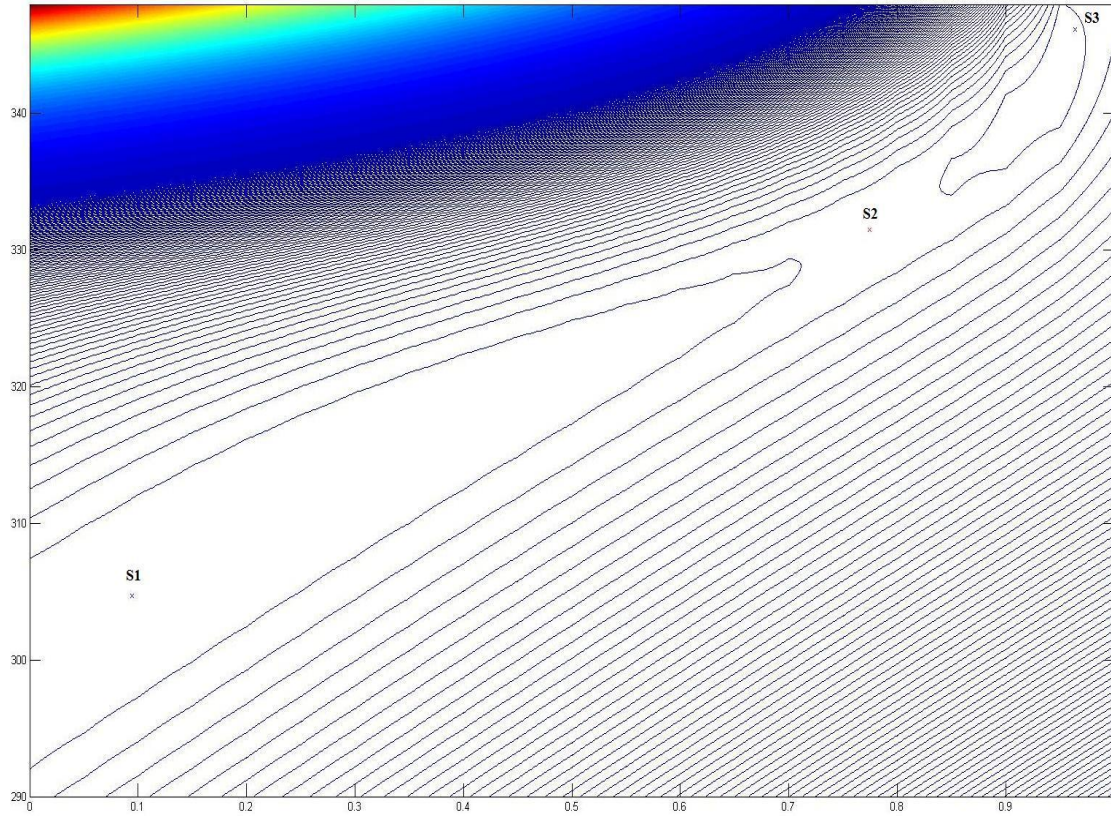


Figure B.12: Contour plot containing results of example 3

The results for this example agreed with Lucia's reported values, as shown in Table

B.5. The total CPU computational time was 9.4069 seconds.

Table B.5: Comparison of Lucia's results to our results

No.	Lucia's Results		Ex. 3 Results		
	Position	Eigenvalue	Position	Eigenvalue	
S ₁	0.094594,304.6772	2.10713,	0.0946,304.6772	2.1071,343486.7730	Local Minimum
S ₂	0.774548,331.5069	-92.7533,	0.7745,331.5069	-92.7333,874274.3390	Saddle
S ₃	0.963868,346.1637	Not specified!	0.9639,346.1637	98.2769,11307855.7208	Solution

4.1.4 Example 4 – Catalytic Pellet Reactor

Non-isothermal pellets are known to exhibit multiple steady state solutions we would like to obtain. In this case, we wish to find all solutions of steady state non-isothermal concentrations and temperature in the catalytic pellet. The pellets can be treated as a porous media, where reactant diffuses axially along the reactor, and also radially into the pellets. As we focus our attention on solving pellets shown in Figure B.13, catalytic pellets are porous to allow diffusion of reactant, consequently creating a concentration gradient with respect to the radius.

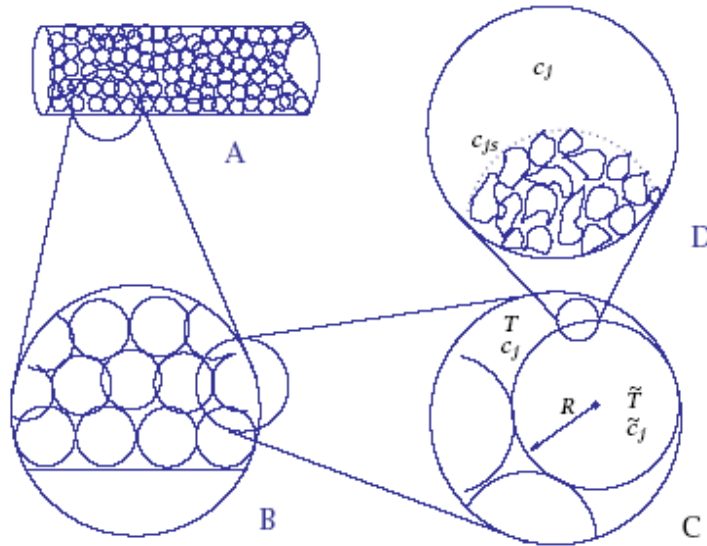


Figure B.13: Expanded view of a fixed-bed pellet reactor

Developing mathematical model

Isothermal pellets only present mass transfer within the solid and is modeled by Eqn.

(B.31),

$$\frac{dC_A^2}{dr^2} + \frac{2}{r} \frac{dC_A}{dr} - \frac{C_A^2}{De} = 0 \quad \text{B.31}$$

where D_e is diffusivity coefficient of the pellet.

The boundary conditions are

$$C_A|_{r=R} = C_S, \quad \frac{dC_A}{dr}|_{r=0} = 0 \quad \text{B.32}$$

For non-isothermal condition the temperature profile is also developed with concentration profile.

$$\begin{aligned} \frac{d^2\varphi}{d\theta^2} + \frac{2}{\theta} \frac{d\varphi}{d\theta} - \Phi^2 \exp\left[\gamma\left(1 - \frac{1}{\zeta}\right)\right] \varphi &= 0 \\ \frac{d^2\zeta}{d\theta^2} + \frac{2}{\theta} \frac{d\zeta}{d\theta} - \beta \exp\left[\gamma\left(1 - \frac{1}{\zeta}\right)\right] \varphi &= 0 \\ \varphi = \frac{C_A}{C_{ref}}, \theta = \frac{r}{R}, \zeta = \frac{T}{T_{ref}}, \gamma = \frac{E_a}{RT}, \beta = \frac{\Delta H k_s}{\lambda} \end{aligned} \quad \text{B.33}$$

a is a dimensionless variable that accounts for reference reaction rate coefficient.

From here on, we will directly indicate reference conditions to be surface conditions, which are indicated by subscript s. R is the radius of the pellet. The boundary conditions for the dimensionless equations are:

$$[\zeta]_{\theta=1} = \frac{T}{T_s} = 1, \left[\frac{d\zeta}{d\theta}\right]_{\theta=0} = 0 \quad \text{B.34}$$

$$[\varphi]_{\theta=1} = \frac{C}{C_s} = 1, \left[\frac{d\varphi}{d\theta}\right]_{\theta=0} = 0 \quad \text{B.35}$$

Damkohler has pointed out that there is a relationship between the reactant concentrations and the temperatures at any point within the pellet.

$$\Delta T = T - T_s = -\frac{HD}{K}(C_s - C) \quad \text{B. 36}$$

So, expressing the mass diffusion equation is with a single variable is shown below in Eqn. (B.37).

$$\frac{d^2y}{dx^2} + \frac{2dy}{xdx} = \phi^2 y \exp\left(\gamma\beta \frac{1-y}{1+\beta(1-y)}\right) \quad \text{B.37}$$

$$g = \frac{E_a}{RT}, b = \frac{C_s HD}{KT_s}, f = R\sqrt{(k_o/D)}, y = \frac{C}{C_s}, x = \frac{r}{R}$$

The parameters g and β each have a convenient physical significance. g expresses the sensitivity of the reaction rate to temperature and β is the maximum temperature variation which could exist within the particle relative to the boundary temperature.

To solve these equations numerically, a collocation method using a Lagrangian polynomial is implored. The definition of this method is,

$$C_{approximate}(x) = \sum_{j=1}^n l_j \phi_j(x) \quad \text{B.38}$$

To discretize the differential Eqn. (B.37) and boundary condition (B.41) a Lagrangian polynomial is used,

$$y(x) = \sum_{i=0}^m l_i(x) y_i \quad \text{with} \quad l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^N \frac{x - x_j}{x_i - x_j} \quad \text{B.39}$$

Its first order derivative and second order derivative are:

$$y'(r)|_{x=x_1} = \frac{dy(r)}{dr} = \sum_{i=0}^N \frac{d(l_i(r))}{dr} y_i = \sum_{i=0}^N l'_i(r) y_i \quad \text{B.40}$$

$$y''(r)|_{x=x_1} = \frac{d^2 y}{dr^2} = \sum_{i=0}^N \frac{d^2(l_i(r))}{dr^2} y_i = \sum_{i=0}^N l''_i(r) y_i = \sum_{i=0}^N l''_{i,x_1} y_i \quad \text{B.41}$$

We set five nodes $[0, 0.25, 0.5, 0.75, 1]$ and from Eqn. (B.31), Eqn. (B.36) and Eqn. (B.38), 1 linear equation and 3 nonlinear equations are obtained.

The concentration function approximated by

$$\begin{aligned} \sum_{i=0}^m l''_i(x_i) y_i + \frac{2}{x_i} \sum_{i=0}^m l'_i(x_i) y_i \\ = \phi^2 \left(\sum_{i=0}^m l_i(x_i) y_i \right) \exp \left[\gamma \beta \frac{1 - \sum_{i=0}^m l_i(x_i) y_i}{1 + \beta (1 - \sum_{i=0}^m l_i(x_i) y_i)} \right] \end{aligned} \quad \text{B.42}$$

Also with boundary condition defined by Eqn. (B.43).

$$y_m = 1$$

$$\left. \frac{dy}{dx} \right|_{x_1} = 0, \quad \text{B.43}$$

When $i=1 \dots N$, $N-1$ equations which also have $N-1$ variables (y_i) are obtained from Eqn. (B.42) and Eqn. (B.43).

Isothermal case

For the isothermal case, system (B.44) is obtained from Eqn. (B.42) and Eqn. (B.43).

To get this equations, the node $N=5$ is used and the node values are (0, 0.25, 0.5, 0.75, 1).

$$\begin{aligned} f_1 &= -8.3333x_1 + 16.0000x_2 - 12.0000x_3 + 5.3333x_4 - 1.0000 \\ f_2 &= 6.6667x_1 - 53.3333x_2 - 1.0000F^2x_2^2 + 56.0000x_3 - 10.6667x_4 + 1.3333 \\ f_3 &= 10.6667x_2 - 40.0000x_3 - 1.0000F^2x_3^2 + 32.0000x_4 - 2.6667 \\ f_4 &= -2.2222x_1 + 10.6667x_2 - 8.0000x_3 - 17.7778x_4 - 1.0000F^2x_2^2 + 17.3333 \end{aligned} \quad \text{B.44}$$

We solved Eqn. (B.44) using the global terrain method. It found one minimum solution at (0.9206, 0.9253, 0.9397, 0.9643). Figure B.14 shows concentration profile of the isothermal condition. One result is also obtained solving Eqn. (B.44) analytically. We know that there is some difference, but trend is same.

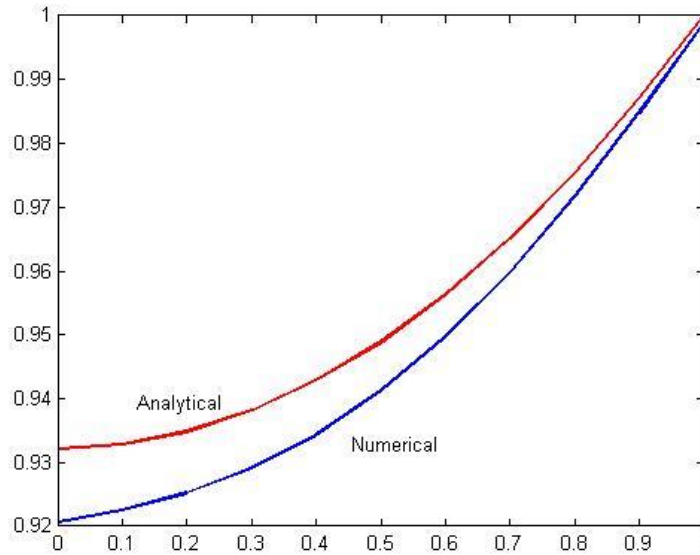


Figure B.14: Concentration and temperature profile in pellet

Non-isothermal case with two nodes

For non-isothermal case, temperature and concentration profiles must be solved simultaneously. In this case, two kinds of nodes are used: [0, 0.5, 1] and [0, 0.25, 0.5, 0.75, 1]. Each set of main equations is shown as Eqn. (B.45) and Eqn. (B.46).

$$\begin{aligned} f_1 &= -3.0000x_1 + 4.0000x_2 - 1.0000 \\ f_2 &= -8.0000x_2 + 8.0000 - \phi^2 x_2 \exp\left(\frac{\gamma\beta(1-x_2)}{1+\beta(1-x_2)}\right) \end{aligned} \quad \text{B.45}$$

$$\begin{aligned} f_1 &= -8.3333x_1 + 16.0000x_2 - 12.0000x_3 + 5.3333x_4 - 1.0000 \\ f_2 &= 6.6667x_1 - 53.3333x_2 + 56.0000x_3 - 10.6667x_4 + 1.3333 \\ &\quad - \phi^2 x_2 \exp\left(\frac{\gamma\beta(1-x_2)}{1+\beta(1-x_2)}\right) \\ f_3 &= 10.6667x_2 - 40.000x_3 + 32.0000x_4 - 2.6667 \\ &\quad - \phi^2 x_3 \exp\left(\frac{\gamma\beta(1-x_3)}{1+\beta(1-x_3)}\right) \\ f_4 &= -2.2222x_1 + 10.6667x_2 - 8.0000x_3 - 17.7778x_4 + 17.3333 \\ &\quad - \phi^2 x_4 \exp\left(\frac{\gamma\beta(1-x_4)}{1+\beta(1-x_4)}\right) \end{aligned} \quad \text{B.46}$$

The value of parameters are $\beta = 0.2$, $\gamma = 30$, and $\phi = 0.7314$. These values came from Weisz and Hicks (1962).

In case of three nodes, the result is in Table B.6. In this table, two minima points and two saddles were obtained. Another minimum point S_5 was obtained, But S_5 is not the answer because it is out of feasible region.

Table B.6: The results of non-isothermal case $N=3$, $\beta = 0.2$, $\gamma = 30$, and $\phi = 0.7314$

No	Position	Eigenvalue	Solution Type
S_1	0.8433,0.8825	3.4761,35.0685	Minimum
S_2	0.6001,0.7001	-4.3721,19.7687	Saddle
S_3	0.3636,0.5227	3.6113,35.7228	Local Minimum
S_4	0.0630,0.2972	-44.2738,11.7030	Saddle
S_5	-0.0963,0.1778	8.4466,269.2281	Out of Range

The contour plot of this case is shown in Figure B.15.

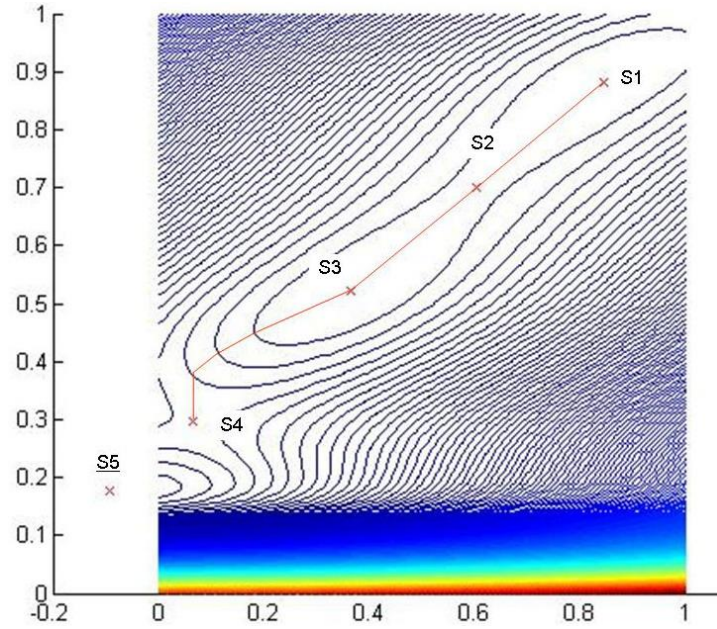


Figure B.15: Connectedness of stationary points for example 4($N=3$), $\beta = 0.2$, $\gamma = 30$, and $\phi = 0.7314$

Concentration and temperature profiles are shown in Figure B.16.

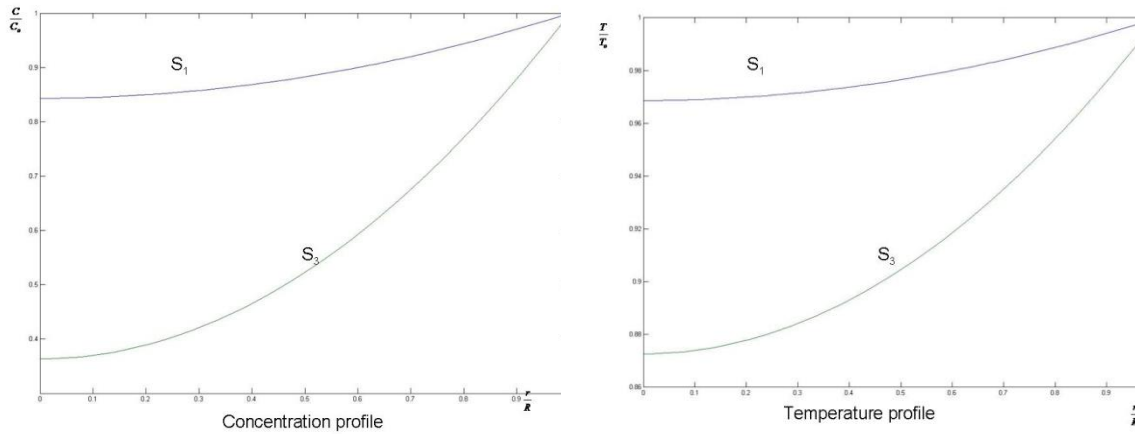


Figure B.16: Concentration and Temperature profile in pellet ($N=3$), $\beta = 0.2$, $\gamma = 30$, and $\phi = 0.7314$

Non-isothermal case with five nodes

When the amount of nodes is equal to five, the result is shown in Table B.7,

Table B.7: Results of non-isothermal case $N=5$, $\beta = 0.2$, $\gamma = 30$, and $\phi = 0.7314$

No	Position	Eigenvalue	Solution Type
S_1	0.8574,0.8684,0.8990,0.9449	10.0694,79.5217,1063.2216,7838.6649	Minimum

S_2	0.3305,0.4100,0.5765,0.7827	-0.4258,72.6108,883.7917,6802.8926	Saddle
S_3	0.3136,0.3949,0.5666,0.7786	0.4374,72.5834,883.6059,6808.1188	Minimum

And the Concentration and Temperature profiles are shown in Figure B.17.

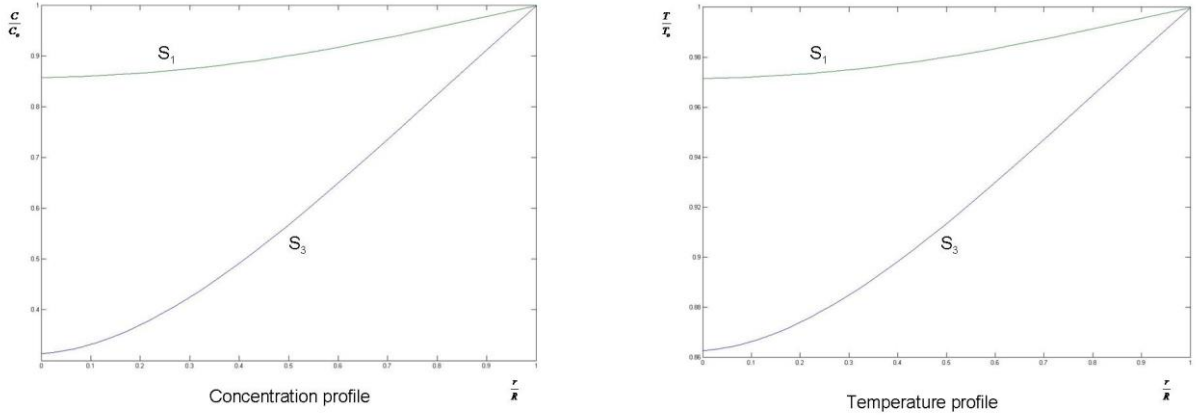


Figure B.17: Concentration and Temperature profile in pellet ($N= 5$) $\beta = 0.2$, $\gamma = 30$, and $\phi = 0.7314$

Another result for different values of $\beta = 0.4$, $\gamma = 30$, and $\phi = 0.430$ are in Table B.8.

Concentration and temperature profiles are illustrated in Figure B.18.

Table B.8: Results of non-isothermal case $N=5$, $\beta = 0.4$, $\gamma = 30$, and $\phi = 0.430$

No	Position	Eigenvalue	Solution Type
S_1	0.5338,0.6271,0.7836,0.9032	19.1965,266.4771,768.5611,5006.1343	Minimum
S_2	0.7465,0.7804,0.8384,0.9116	-17.3244,68.8776,893.4418,6728.4457	Saddle
S_3	0.9578,0.9608,0.9695,0.9830	14.4848,85.1176,1115.9223,8055.2649	Minimum

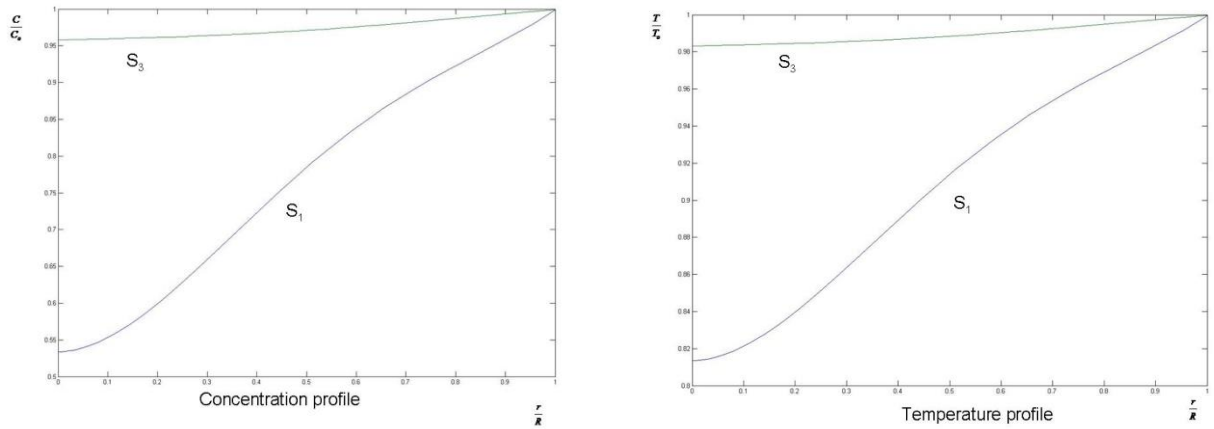


Figure B.18: Concentration and Temperature profile in pellet (N=5) $\beta = 0.2$, $\gamma = 30$, and $\phi = 0.7314$

Weisz and Hicks (1962) said that multiple solutions can be found when $\beta \geq 5$. Using the global terrain method, we found these multiple solutions.

Numerical derivative method

With a numerical derivative like with Example 1, we got the same results with the analytical derivative method. Table B.9 shows the results of the numerical derivative method. And in this case, calculation times of two methods are not quite different. This proves that we can use numerical derivative method in global terrain method.

Table B.9 Results of non-isothermal case using numerical derivative, N=5, $\beta = 0.4$, $\gamma = 30$, and $\phi = 0.430$

No	Position	Eigenvalue	Solution Type
S ₁	0.5338,0.6271,0.7836,0.9032	19.1965,266.4771,768.5611,5006.1343	Minimum
S ₂	0.7465,0.7804,0.8384,0.9116	-17.3244,68.8776,893.4418,6728.4457	Saddle
S ₃	0.9578,0.9608,0.9695,0.9830	14.4848,85.1176,1115.9223,8055.2649	Minimum

5 Conclusion

The global terrain method proposed by Lucia and Yang was studied in this paper. This method was devised as a way to find the stationary points of a function through the use of a least-squares landscape. The method uses the ledges, ridges, and valleys of the least-squares landscape in order to travel from one stationary point to another. An uphill step from a minimum point uses a Newton-like step in conjunction with a sequential quadratic programming corrector step to find a saddle or maximum point. From these saddle and maximum points, a trust region method is used to find other minima.

This method was able to solve systems of nonlinear equations and solve for the real and imaginary roots of a polynomial function in a short amount of CPU computational time. A chemical engineering case involving a nonlinear continuous stirred tank reactor was also studied. All three of these case studies resulted in the determination of all stationary points on the least squares function. The results show that the global terrain method is able to solve small dimensional problems reliably and quickly.

REFERENCES

- Adjiman, C. S., Androulakis, I. P., Maranas, C. D., & Floudas, C. A. (1996). A global optimization method, α BB, for process design. *Computers & Chemical Engineering*, 20, S419-S424.
- Alkhayyal, F. A. (1990). Jointly constrained bilinear programs and related problems - an overview. *Computers & Mathematics with Applications*, 19, 53-62.
- Alkhayyal, F. A., & Falk, J. E. (1983). Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8, 273-286.
- Androulakis, I. P., Maranas, C. D., & Floudas, C. A. (1995). α BB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7, 337-363.
- Bahri, P. A., Bandoni, J. A., & Romagnoli, J. A. (1997). Integrated flexibility and controllability analysis in design of chemical processes. *AIChE Journal*, 43, 997.
- Bansal, V., Perkins, J. D., & Pistikopoulos, E. N. (2002). A case study in simultaneous design and control using rigorous, mixed-integer dynamic optimization models. *Industrial & Engineering Chemistry Research*, 41, 760-778.
- Beasley, D., Bull, D. R., & Martin, R. R. (1993). A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1, 101-125.
- Brengel, D. D., & Seider, W. D. (1992). Coordinated design and control optimization of nonlinear processes. *Computers & Chemical Engineering*, 16, 861-886.
- Chelouah, R., & Siarry, P. (2003). Genetic and nelder-mead algorithms hybridized for a more accurate global optimization of continuous multim minima functions. *European Journal of Operational Research*, 148, 335-348.

- Cowell, W. R. (1984). Sources and development of mathematical software. Englewood Cliffs, N.J.: Prentice-Hall.
- Davidor, Y., Männer, R., & Schwefel, H.-P. (1994). Parallel problem solving from nature-
- evolutionary computation : International Conference on Evolutionary Computation held jointly with the Third Conference on Parallel Problem Solving from Nature, PPSN III, Jerusalem, Israel, October 9-14, 1994 : proceedings. Berlin ; New York: Springer-Verlag.
- De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems: University of Michigan.
- Deb, K. (2000). An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, 186, 311-338.
- Deb, K., & Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In proceedings of the Third International Conference on Genetic Algorithms Morgan Kaufmann, 42-55.
- Dimitriadis, V. D., & Pistikopoulos, E. N. (1995). Flexibility analysis of dynamic systems. Ind. Eng. Chem. Res., 34, 4451.
- Egea, J. A., Rodriguez-Fernandez, M., Banga, J. R., & Marti, R. (2007). Scatter search for chemical and bio-process optimization. Journal of Global Optimization, 37, 481-503.
- Floudas, C. A. (2000). Deterministic global optimization: theory, methods, and applications. Dordrecht ; Boston: Kluwer Academic Publishers.

- Floudas, C. A., Gumus, Z. H., & Ierapetritou, M. G. (2001). Global optimization in design under uncertainty: Feasibility test and flexibility index problems. *Industrial & Engineering Chemistry Research*, 40, 4267-4282.
- Floudas, C. A., & Visweswaran, V. (1990). A global optimization algorithm (Gop) for certain classes of nonconvex Nlps .1. Theory. *Computers & Chemical Engineering*, 14, 1397-1417.
- Floudas, C. A., & Visweswaran, V. (1993). Primal-relaxed dual global optimization approach. *Journal of Optimization Theory and Applications*, 78, 187-225.
- Gan, J., & Warwick, K. (2001). Dynamic niche clustering: A fuzzy variable radius niching technique for multimodal optimisation in GAs. In: *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, IEEE Press.*
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability : a guide to the theory of NP-completeness*. San Francisco: W. H. Freeman.
- Gatzke, E. P., & Doyle III, F. J. (2002). Use of multiple models and qualitative knowledge for on-line moving horizon disturbance estimation and fault diagnosis. *Journal of Process Control*, 12, 339-352.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass.: Addison-Wesley Pub. Co.
- Goldberg, D.E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*,. 41-49.

- Goldberg, D.E. (1989). Sizing populations for serial and parallel genetic algorithms. In Proceedings of the Third International Conference on Genetic Algorithms., Morgan Kaufman, 70-79.
- Grossmann, I. E., & Floudas, C. A. (1987). Active constraint strategy for flexibility analysis in chemical processes. *Computers & Chemical Engineering*, 11, 675-693.
- Halemane, K. P., & Grossmann, I. E. (1983). Optimal process design under uncertainty. *AIChE Journal*, 29, 425-433.
- Hansen, E. R. (1979). Global optimization using interval analysis - One-dimensional case. *Journal of Optimization Theory and Applications*, 29, 331-344.
- Hansen, P., Jaumard, B., & Lu, S. H. (1992). Global optimization of univariate Lipschitz Functions .1. Survey and Properties. *Mathematical Programming*, 55, 251-272.
- Hansen, P., Jaumard, B., & Lu, S. H. (1992). Global Optimization of Univariate Lipschitz functions. 2. New Algorithms and Computational Comparison. *Mathematical Programming*, 55, 273-292.
- Haseltine, E. L., & Rawlings, J. B. (2005). Critical evaluation of extended Kalman filtering and moving-horizon estimation. *Industrial & Engineering Chemistry Research*, 44, 2451-2460.
- Holland, J. H. (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. Ann Arbor: University of Michigan Press.
- Horst, R., & Hoang, T. (1996). Global optimization: deterministic approaches (3rd rev. and enl. ed.). Berlin; New York: Springer.

- Hsia, T. C. (1977). System identification: least-squares methods. Lexington, Mass.: Lexington Books.
- Ierapetritou, M. G. (2001). New approach for quantifying process feasibility: Convex and 1-D quasi-convex regions. *AIChE Journal*, 47, 1407-1417.
- James, B. A. P. (1985). Variance reduction techniques. *Journal of the Operational Research Society*, 36, 525-530.
- Jelasy, M. (1998). An abstract niching technique for global optimization. In: *Parallel problem solving from Nature-PPSN V*, 1498. Amsterdam : Springer.
- Kan, A. H. G. R., & Timmer, G. T. (1987). Stochastic global optimization methods .1. clustering methods. *Mathematical Programming*, 39, 27-56.
- Kim, J.-H., & Myung, H. (1997). Evolutionary programming techniques for constrained optimization problems. *Evolutionary Computation, IEEE Transactions on*, 1, 129-140.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.
- Kookos, I. K., & Perkins, J. D. (2001). An Algorithm for Simultaneous Process Design and Control. *Industrial & Engineering Chemistry Research*, 40, 4079-4088.
- Kulkarni, K., Moon, J., Zhang, L. B., Lucia, A., & Linninger, A. A. (2008). Multiscale Modeling and Solution Multiplicity in Catalytic Pellet Reactors. *Industrial & Engineering Chemistry Research*, 47, 8572-8581.
- Lawler, E. L. (1985). The Traveling salesman problem : a guided tour of combinatorial optimization. Chichester West Sussex ; New York: Wiley.

- Lenhoff, A. M., & Morari, M. (1982). Design of Resilient Processing Plants .1. Process Design under Consideration of Dynamic Aspects. *Chemical Engineering Science*, 37, 245-258.
- Levy, A. V., & Montalvo, A. (1985). The Tunneling Algorithm for the Global Minimization of Functions. *Siam Journal on Scientific and Statistical Computing*, 6, 15-29.
- Li, F., & Aggarwal, R. K. (2000). Fast and accurate power dispatch using a relaxed genetic algorithm and a local gradient technique. *Expert Systems with Applications*, 19, 159-165.
- Liebman, M. J., Edgar, T. F., & Lasdon, L. S. (1992). Efficient Data Reconciliation and Estimation for Dynamic Processes Using Nonlinear-Programming Techniques. *Computers & Chemical Engineering*, 16, 963-986.
- Lucia, A., DiMaggio, P. A., Bellows, M. L., & Octavio, L. M. (2005). The phase behavior of n-alkane systems. *Computers & Chemical Engineering*, 29, 2363-2379.
- Lucia, A., & Feng, Y. (2002). Global terrain methods. *Computers & Chemical Engineering*, 26, 529-546.
- Lucia, A., Gattupalli, R. R., Kulkarni, K., & Linninger, A. (2008). A barrier-terrain methodology for global optimization. *Industrial & Engineering Chemistry Research*, 47, 2666-2680.
- Luyben, M. L., & Floudas, C. A. (1994). Analyzing the interaction of design and control: I. A multiobjective framework and application to binary distillation synthesis. *Computers & Chemical Engineering*, 18, 933.

- Luyben, M. L., Tyreus, B. D., & Luyben, W. L. (1996). Analysis of control structures for reaction/separation/recycle processes with second-order reactions. *Industrial & Engineering Chemistry Research*, 35, 758-771.
- Luyben, W. L. (2007). *Chemical reactor design and control*. New York Hoboken, N.J.: AIChE ;Wiley-Interscience.
- Luyben, W. L., Tyréus, B. D., & Luyben, M. L. (1998). *Plantwide process control*. New York: McGraw-Hill.
- Mahfoud, S.W. (1992). Crowding and preselection revisited. In Manner, R., Manderick , B.(Eds.), *Parallel Problem Solving from Nature*, 2(pp. 27-36). Amsterdam, The Netherlands. Elsevier Science.
- Malcolm, A., Polan, J., Zhang, L., Ogunnaike, B. A., & Linninger, A. A. (2007). Integrating systems design and control using dynamic flexibility analysis. *AIChE Journal*, 53, 2048-2061.
- Maner, B. R., Doyle, F. J., Ogunnaike, B. A., & Pearson, R. K. (1996). Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. *Automatica*, 32, 1285-1301.
- Maranas, C. D., & Floudas, C. A. (1994). Global minimum potential-energy conformations of small molecules. *Journal of Global Optimization*, 4, 135-170.
- Maranas, C. D., & Floudas, C. A. (1997). Global optimization in generalized geometric programming. *Computers & Chemical Engineering*, 21, 351-369.
- Mcdonald, C. M., & Floudas, C. A. (1995). Global optimization for the phase-stability problem. *AIChE Journal*, 41, 1798-1814.

- Mckay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42, 55-61.
- Michalewicz, Z. (1996). Genetic algorithms + data structures = evolution programs (3rd rev. and extended ed.). Berlin ; New York: Springer-Verlag.
- Miller, B.L., & Shaw, M. J. (1996). Genetic algorithms with dynamic niche sharing for multimodal function optimization, *IEEE international conf. on Evolutionary Computations* 786-791
- Mohideen, M. J., Perkins, J. D., & Pistikopoulos, E. N. (1996). Optimal design of dynamic systems under uncertainty. *AIChE Journal*, 42, 2251.
- Mohideen, M. J., Perkins, J. D., & Pistikopoulos, E. N. (1996). Optimal synthesis and design of dynamic systems under uncertainty. *Computers & Chemical Engineering*, 20, S895-S900.
- Moon, J., Kulkarni, K., Zhang, L., & Linninger, A. A. (2008). Parallel hybrid algorithm for process flexibility analysis. *Industrial & Engineering Chemistry Research*, 47, 8324-8336.
- Narraway, L., & Perkins, J. (1994). Selection of process-control structure-based on eEconomics. *Computers & Chemical Engineering*, 18, S511-S515.
- Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7, 308-313.
- Neumaier, A., Shcherbina, O., Huyer, W., & Vinko, T. (2005). A comparison of complete global optimization solvers. *Mathematical Programming*, 103, 335-356.

- Ostrovsky, G. M., Volin, Y. M., Barit, E. I., & Senyavin, M. M. (1994). Flexibility analysis and optimization of Chemical-Plants with uncertain parameters. *Computers & Chemical Engineering*, 18, 755-767.
- Palazoglu, A., & Arkun, Y. (1986). A multiobjective approach to design chemical-plants with robust dynamic operability characteristics. *Computers & Chemical Engineering*, 10, 567-575.
- Papalexandri, K. P., & Pistikopoulos, E. N. (1994). A multiperiod MINLP model for the synthesis of flexible heat and mass exchange networks. *Computers & Chemical Engineering*, 18, 1125-1139.
- Perkins, J. D., & Walsh, S. P. K. (1996). Optimization as a tool for design/control integration. *Computers & Chemical Engineering*, 20, 315-323.
- Perkins, J. D., & Wong, M. P. F. (1985). Assessing controllability of chemical-plants. *Chemical Engineering Research & Design*, 63, 358-362.
- Press, W. H. (2007). *Numerical recipes: the art of scientific computing* (3rd ed.). Cambridge, UK ; New York: Cambridge University Press.
- Psarris, P., & Floudas, C. A. (1991). Dynamic operability of MIMO systems with time delays and transmission zeros .1. Assessment. *Chemical Engineering Science*, 46, 2691-2707.
- Rawlings, J. B., & Bakshi, B. R. (2006). Particle filtering and moving horizon estimation. *Computers & Chemical Engineering*, 30, 1529-1541.
- Sabatini, A. M. (2000). A hybrid genetic algorithm for estimating the optimal time scale of linear systems approximations using Laguerre models. *Ieee Transactions on Automatic Control*, 45, 1007-1011.

- Sakizlis, V., Perkins, J. D., & Pistikopoulos, E. N. (2004). Recent advances in optimization-based simultaneous process and control design. *Computers & Chemical Engineering*, 28, 2069-2086.
- Schmid, C., & Biegler, L. T. (1994). Quadratic programming methods for reduced hessian SQP. *Computers & Chemical Engineering*, 18, 817-832.
- Seferlis, P., & Georgiadis, M. C. (2004). *The integration of process design and control*. Amsterdam ; Boston: Elsevier.
- Shir, O. F., & Back, T. (2006). Niche radius adaptation in the CMA-ES niching algorithm. *Parallel Problem Solving from Nature - Ppsn Ix, Proceedings*, 4193, 142-151.
- Stadtherr, M. A., Schnepper, C. A., & Brennecke, J. F. (1995). Robust phase stability analysis using interval methods, *AIChE Symp. Ser.*, 91, 356.
- Sun, A. C., & Seider, W. D. (1995). Homotopy-continuation method for stability analysis in the global minimization of the Gibbs free-energy. *Fluid Phase Equilibria*, 103, 213-249.
- Swaney, R. E., & Grossmann, I. E. (1985). An index for operational flexibility in chemical process design .1. Formulation and theory. *AIChE Journal*, 31, 621-630.
- Tang, W. Y., Zhang, L. B., Linninger, A. A., Tranter, R. S., & Brezinsky, K. (2005). Solving kinetic inversion problems via a physically bounded Gauss-Newton (PGN) method. *Industrial & Engineering Chemistry Research*, 44, 3626-3637.
- Tawarmalani, M., & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99, 563-591.


- Tuy, H., Thieu, T. V., & Thai, N. Q. (1985). A conical algorithm for globally Minimizing a concave function over a closed convex set. *Mathematics of Operations Research*, 10, 498-514.
- Walsh, S. P. K., & Perkins, J. D. (1994). Application of integrated process and control system design to waste water neutralisation. *Computers & Chemical Engineering*, 18, S183.
- Weisfeld, M. A. (2009). The object-oriented thought process. In *Developer's library* (3rd ed.). Upper Saddle River, N.J.: Addison-Wesley.
- Weisz, P. B. (1979). Citation classic - behavior of porous catalyst particles in view of internal mass and heat diffusion effects. *Current Contents/Engineering Technology & Applied Sciences*, 26-26.
- Zhang, L. B., Kulkarni, K., Somayaji, M. R., Xenos, M., & Linninger, A. A. (2007). Discovery of transport and reaction properties in distributed systems. *AIChE Journal*, 53, 381-396.

APPENDIX C- PERMISSION TO REUSE PUBLISHED MATERIALS

American Chemical Society Publication

http://pubs.acs.org/page/copyright/learning_module/module.html

[Log In](#) | [Register](#) | [Cart](#) | ACS | **ACS Publications** | [C&EN](#) | [CAS](#)

 ACS Publications
Most Trusted. Most Cited. Most Read.

ACS Journals | ACS eBooks | C&EN Global Enterprise

[Search](#) | [Citation](#) | [Subject](#) | [Advanced Search](#)

Enter search text / DOI | Anywhere | [Search](#)

[Home](#) | [Authors & Reviewers](#) | [Librarians & Account Managers](#) | [ACS Members](#) | [Alerts](#) | [About Us](#) | [ACS & Open Access](#)

Learning Module: What Chemists Need to Know about Copyright [Back to Learning Module Contents](#)

Frequently Asked Questions about Copyright

Presented by the American Chemical Society Joint Board/Council Committee on Publications Subcommittee on Copyright

General Questions

1. What is copyright?

...

30. May I compile a retrospective publication with a non-American Chemical Society (ACS) publisher containing the papers I have published in ACS journals? What restrictions are there on such use?

31. I am a student writing my thesis. May I use papers I have authored in ACS journals, or material from them, in my thesis without obtaining explicit permission?

You may reuse all or part of the Submitted, Accepted, or Published versions of your ACS papers in your thesis or dissertation. Such reuse is permitted subject to the ACS' [Ethical Guidelines to Publication of Chemical Research](#) and you should secure written confirmation from the respective ACS journal editor(s) to avoid potential conflicts with journal [prior publication policies](#). The ACS copyright credit line should be noted on the appropriate pages and appropriate citation of any Published versions. If the thesis or dissertation to be published is in electronic format, a direct link to the Published Work must be included using the [ACS Articles on Request](#) link.

Although ACS grants students automatic permission to use their ACS articles in theses, it is highly likely that the graduate school requires a statement of written permission. Students should use [Rightslink](#) to obtain permission, and provide their graduate school with the written document provided by Rightslink.

You may also wish to view the [ACS Thesis Policy](#) [PDF].

ADVERTISEMENT

A Complete Workflow for Comprehensive Characterisation of Monoclonal Antibodies

October 24, 2017
US Broadcast: 11:00am EDT
10:00am CDT / 8:00am PDT



For further guidelines about obtaining permission, please review our Frequently Asked Questions below:

[When is permission required? +](#)

[When is permission not required? +](#)

[From whom do I need permission? +](#)

[How do I obtain permission to use photographs or illustrations? +](#)

[Do I need to obtain permission to use material posted on a website? +](#)

[What rights does Elsevier require when requesting permission? +](#)

[How do I obtain permission from another publisher? +](#)

[What is Rightslink? +](#)

[What should I do if I am not able to locate the copyright owner? +](#)

[What is Elsevier's policy on using patient photographs? +](#)

[Can I obtain permission from a Reproduction Rights Organization \(RRO\)? +](#)

[Is Elsevier an STM signatory publisher? +](#)

[Do I need to request permission to re-use work from another STM publisher? +](#)

[Do I need to request permission to text mine Elsevier content? +](#)

[Can I post my article on ResearchGate without violating copyright? +](#)

[Can I post on ArXiv? +](#)

[Can I include/use my article in my thesis/dissertation? +](#)

Yes. Authors can include their articles in full or in part in a thesis or dissertation for non-commercial purposes.

[Which uses of a work does Elsevier view as a form of 'prior publication'? +](#)

JEONGHWA MOON

Ph.D Candidate and Graduate Research Assistant

University	Laboratory for Product and Process Design Department of Chemical Engineering (MC 110), 810 S. Clinton St, University of Illinois at Chicago Chicago, IL, 60607, USA. <i>Tel:</i> 312-355-2520 <i>E-Mail:</i> jmoon8(at)uic.edu	Residence	1300 Cambia Dr, #5315 Schaumburg, IL, 60193, USA <i>Tel:</i> 847-704-0490 <i>E-Mail:</i> jmoon8(at)gmail.com
-------------------	--	------------------	---

RESEARCH INTERESTS

- Integration of Process Design and Control
- Dynamic simulation
- Global Optimization
- Object Oriented Programming of Numerical Solver

EDUCATION

UNIVERSITY OF ILLINOIS AT CHICAGO, Chicago, USA (2005 - present)

College of engineering
Doctor of Philosophy (Ph.D) in chemical engineering
Class of 2005. Expected Graduation: Fall, 2009
Current GPA: 3.64/4.0

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY (POSTECH), Pohang, Korea (2000- 2002)

Master of Science (M. S.) in chemical engineering, GPA: 3.50/4.3
Bachelor of Science (B. S.) in chemical engineering, GPA: 3.16/4.3

RESEARCH EXPERIENCE

- Ph.D Dissertation – “Process Design under Uncertainty, Models and Algorithms for Global Optimization” – Ongoing (*Principal Ph.D Advisor:* Dr. Andreas Linninger, Assoc. Prof, chemical and bio engineering, UIC)
- Graduate research assistant, Laboratory for Product and Process Design, department of chemical engineering, UIC.
- M.S. Dissertation – “Object Oriented Analysis and Design for an Intelligent Operation Support System using UML”, POSTECH 2002.
- Graduate research assistant, Intelligent Process System Engineering, department of chemical engineering, POSTECH.

INDUSTRIAL EXPERIENCE

- SK Computers & Communications, Ltd., Seoul, Korea: software engineer, assistant manager, 2002- 2005.
SK C&C is the 3rd largest IT company in Korea.
Participated in several system integration projects.
Managed Terminal Automation System (TAS) in SK energy corp.
- ISYSTECH Co. Ltd., Pohang, Korea: programmer, 2000-2002.
Developed software for the multivariate statistical control system (ISYS-MSPC).
Participated in several projects about statistical process controls.

TEACHING EXPERIENCE

- Student Advisor in National Science Foundation – Research Experience for Undergraduates (NSF-REU):
Advised a NSF-REU Student, Summer, 2007-09, UIC.
- Teaching Assistant :
Process control, UIC, Spring, 2006.
Process analysis and design, POSTECH, Spring, 2001.

GRADUATE LEVEL COURSEWORKS (GPA = 3.64/4.0)

Selected coursework: transport phenomena, advanced mass transfer, advanced chemical reaction engineering, computer control theory, fluid phase equilibria, biochemical engineering, numerical mathematics, database systems, and process data analysis and modeling.

COMPUTER SKILLS/EXPERIENCES

- Programming languages: C/C++ , Pascal, Python, Java, MATLAB, etc
- Simulation & optimization tools - HYSYS, GAMS
- Database (Basic user): Oracle 8.0, SQL
- Object oriented design languages/tools: UML, Borland Together (Basic)
- Technical supporting: Computer management in depart. of chem. Eng. as Graduate Assistant, UIC, 2005-2006

PROGRAMMING CODE DEVELOPMENT

- LPPD Solver ver 5 - Newton methods, graphic user interface
- Genetic algorithm optimizers-Basic GA, constrained GAs, niche algorithms, hybrid GAs
- Modeling for integrated design and control case studies : MATLAB m-file, mex interface
- Intelligent system for multivariate statistical control- user interface, algorithms (PLS,PCA)
- User interface for fire dynamics simulator- pre interpreter, graphic user interface

AWARDS/HONORS

<i>Year</i>	<i>Awards/Honors</i>	<i>Details</i>
1) 2008	Fellowship for attending the PASI Conference	Recipient of Fellowship for Pan American advanced studied institute program on emerging trends in process systems engineering, 2008.
2) 2007	FOCAPO Student Grant Award	Recipient of Grand Award for paper presentation at FOCAPO Meeting, Cambridge, MA, 2008
3) 2007, 2009	UIC Graduate College Travel Award	Recipient of UIC Graduate College Travel Award for technical paper presentation at 2007 AIChE Annual Meeting, Salt Lake city, UT, 2007.

RESEARCH PUBLICATIONS

Journal Papers

1. Moon, J., and Linninger, A.A., "A Hybrid Sequential Niche Algorithm for Optimal Engineering Design with Solution Multiplicity", Computers and Chemical Engineering, 33, (7), 1261-1271, 2009.
2. Moon, J., Kulkarni, K., Zhang, L., and Linninger, A.A., "Parallel Hybrid Algorithm for Process Flexibility Analysis", Industrial and Engineering Chemistry Research, 47 (21), 8324-8436, 2008.
3. Kulkarni, K., Moon, J., Zhang, L., Lucia, A. and Linninger, A.A., "Multi-scale Modeling and Solution Multiplicity in Catalytic Pellet Reactors", Industrial and Engineering Chemistry Research, 47 (22), 8571-8582, 2008.

Proceeding Papers

4. Moon, J., Zhang, L., and Linninger, A.A., "Algorithmic Approaches to Integrated Design and Control under Uncertainty", 5th International Conference on Foundations of Computer-Aided Process Operations, June 29-July 2, Cambridge, MA, USA, pp. 217-220, 2008.
5. Moon, J., Kim, S., and Linninger, A.A., "Integrated Design and Control under Uncertainty-Algorithms and Applications", 7th International Conference on Foundations of Computer-Aided Process Design, June 7-12, Breckenridge, Colorado, USA, 2009.

Conferences and Posters

6. Moon, J., Kim, S., Ruiz, G. And Linninger, A.A., "Advanced Hybrid Sequential Niche Algorithm for Finding Multiple Solutions in Global Optimization", AIChE Annual Meeting, November 8-13, Nashville, TN, 2009.
7. Moon, J., and Linninger A.A., "A Hybrid Sequential Algorithm for multimodal Optimization Problems", 10th International Symposium on Process Systems Engineering, August 23-28, Chicago, IL, 2009.
8. Moon, J., and Linninger A., "Embedded Control for Optimizing Flexible Dynamic Process Performance", 20th International Symposium on Mathematical Programming, August 16-20, Salvador, Brazil, 2009.
9. Moon, J., Kim, S., Ruiz, G., and Linninger A.A., "Embedded Control for Optimizing Flexible Dynamic Process Performance", 10th International Symposium on Process Systems Engineering, August 16-20, Salvador, Brazil, 2009.
10. Ruiz, G., Kim, S., Moon, J., and Linninger A.A., "Design and Optimization of Energy Efficient Complex Separation Networks", 10th International Symposium on Process Systems Engineering, August 16-20, Salvador, Brazil, 2009.
11. Moon, J., Kim, S., and Linninger A., "Integrated Design and Control under Uncertainty-Algorithms and Applications", FOCAPO 2009 Meeting, June 7-12, Breckenridge, CO, USA, 2009.
12. Ruiz, G., Kim, S., Moon, J., and Linninger A., "Design and Optimization of Energy Efficient Complex Separation Networks", FOCAPO 2009 Meeting, June 7-12, Breckenridge, CO, USA, 2009.

-
13. Moon, J., Zhang, L., and Linninger A., "Hybrid Niche Algorithm for Problem Inversion of Distributed Systems with Solution Multiplicity", 196e, AIChE Annual Meeting, November 16-21, Philadelphia, PA, 2008.
 14. Zhang, L., Moon, J., Grosman, B., and Linninger A., "Energy-Efficient Complex Column Synthesis", 408d, AIChE Annual Meeting, November 16-21, Philadelphia, PA, 2008.
 15. Zhang, L., Moon, J., Grosman, B., and Linninger A., "Rigorous Mathematic Approach for Chemotherapy Design In the Brain", 382c, November 16-21, Philadelphia, PA, 2008.
 16. Grosman, B., Moon, J., Zhang, L. and Linninger A., "Integrated Design and Control Using Adaptive Full State Identification", 407b, AIChE Annual Meeting, November 16-21, Philadelphia, PA, 2008.
 17. Zhang, L., Grosman, B., Moon, J., and Linninger A., "Structural and Parametric Optimization of Separation Networks", INFORMS Annual Meeting, October 12-15, Washington, DC, 2008.
 18. Moon, J., Zhang, L., and Linninger A., "Algorithmic Approaches to Integrated Design and Control Under Uncertainty", FOCAPO 2008 Meeting, June 29-July 2, Cambridge, MA, 2008.
 19. Zhang, L., Moon, J., and Linninger A.A., "The Synthesis of Separation Networks with Complex Column", AIChE Spring Meeting, April 6-10, New Orleans, LA, 2008.
 20. Moon, J., Zhang, L., and Linninger A.A., "Solving Flexibility Index Problem Using Combined Stochastic Method And Reduced Space Search Algorithms", AIChE Annual Meeting, November 4-9, Salt Lake City, UT, 2007.
 21. Moon, J., Zhang, L., and Linninger A.A., "Algorithmic Approaches to Integrated Design and Control under Uncertainty", AIChE Annual Meeting, November 4-9, Salt Lake City, UT, 2007.
 22. Kulkarni, K., Moon, J., Zhang, L., and Linninger A.A., "Multi-Scale Modeling and Solution Multiplicity in the Catalytic Pellet Reactor", AIChE Annual Meeting, November 4-9, Salt Lake City, UT, 2007.
 23. Moon, J., Zhang, L., and Linninger A.A., "A Hybrid Sequential Niche Genetic Algorithm for Multimodal Objective Function", INFORMS Midwest Regional Conference, August 24-25, Evanston, IL, 2007.
 24. Kulkarni, K., Moon, J., Zhang, L., and Linninger A.A., "Solution Multiplicity of Inversion Problems in Distributed Systems", AIChE Annual Meeting, November 12-17, San Francisco, CA, 2006.
 25. Moon, J., Lee, Y., Lee, D., and Han, C., "ISYS-MSPC: Development of an Intelligent Operation Support System", KICChE Annual Meeting, December 4-9, Seoul, Korea, 2001.

PROFESSIONAL AFFILIATIONS

- Member, Korean American Scientists and Engineering Association (KSEA)
- Student Member, American Institute of Chemical Engineers (AIChE)
- Student Member, Institute for Operations Research and the Management Sciences. (INFORMS)
- Student Member, Korean Institute of Chemical Engineers (KICChE)

MY ADVISORS

- Ph.D - Dr. Andreas Linninger, Associate Professor of Chemical and Bioengineering, University of Illinois at Chicago
- M.S, B.S - Dr. Chonghun Han, Professor of Chemical Engineering, Seoul National University, Korea